URL: http://www.elsevier.nl/locate/entcs/volume2.html 14 pages

Linear Time Computable Problems and Logical Descriptions

Detlef Seese

University Karlsruhe (TH),
AIFB Institute for Applied Computer Science and Formal Description Methods,
D - 76128 Karlsruhe, Germany

e-mail: seese@aifb.uni-karlsruhe.de

Abstract

It is a general problem to investigate the trade off between the complexity of algorithmic problems, the structure of the input objects and the expressive power of problem description languages. The article concentrates on linear time algorithms and on first order logic (**FO**) as problem description language. One of the main results is a proof that each **FO**-problem can be solved in linear time for arbitrary relational structures of universally bounded degree.

Keywords: Linear Time Computability, Relational Structures, Descriptive Complexity Theory, First-Order Logic, Finite Model Theory.

1 Introduction

It could be one of the ultimate goals of computer science to find a general problem solver, i.e. a machinery which transforms each algorithmic problem P into an algorithm to solve P for arbitrary inputs. But unfortunately, it is well-known that there is no such algorithm whose input is a formal description φ of an arbitrary algorithmic problem P and whose output is an algorithm to solve P. Hence it is an interesting question to find restrictions on φ and P under which the above question could have an affirmative answer. A possible formalization is to find a language $\mathcal{L}$ (or more precise a logic, i.e. a language and its semantics) and a class K of structures, such that each problem P which can be described by a formula φ in $\mathcal{L}$ can be solved efficiently for each structure $G \in K$. Now it is the problem to find $\mathcal{L}$ and K in such a way that the expressive power of $\mathcal{L}$ is strong, i.e. many algorithmic problems can be expressed in it, the class K is large, i.e. it contains many structures interesting for applications, and all these problems can be solved with low complexity, i.e. in polynomial or better in linear time.

Many approaches developed in connection with this problem concentrated on strong languages with high expressive power, but had to pay the price

of very restricted classes of structures. The most successful ones investigated extensions of the strong monadic second order language (MSO) or related algebraic-logical approaches in connection with structures of universally bounded tree width, i.e. structures which are up to a certain parameter close related to trees ([2,6,8,9,11,13,14,16,31,36,40,42]). In many of these cases one gets even linear time algorithms. However in case that one is interested in larger classes of structures it seems to be necessary to lower the expressive power of the corresponding language $\mathcal{L}$. With respect to such classes of structures almost all existing general results belong to the theory of descriptive complexity.

Historically, the investigation of connections between complexity classes on one hand and descriptions in (logical) languages on the other started with Ronald Fagin's seminal paper [22], where he proved that **NP** coincides with the class of problems expressible in existential second order logic. Immerman [33] proved corresponding results for **P**, **NL** and several other complexity classes using extensions of the classical first-order calculus by various operators and prompted thus the development of descriptive complexity as an own branch of complexity theory. Several of the articles in this area concentrated on the famos **P**-**NP**– or the **NP**-**Co**-**NP**–Problem and investigated languages whose expressive power is higher than that of first-order logic, e.g. extensions of first-oder logic by certain operators, as e.g. a fixpoint operator (LFP), for which it was proved that **P**= (**FO** + LFP) if structures with an ordered universe are regarded (see [33]). Pure first order logic did not get so much attention, since its expressive power is relatively weak and it was one of the early results of this area that **FO** is strictly contained in **L** and hence in **P**, where **L** denotes the class of deterministic logspace computable problems (see [4,32]).

The main result of this article is a proof that each first-order problem can be solved in linear time if only relational structures of bounded degree are regarded. The basic idea of the proof is a localization technique which based on a method which was originally developed by Hanf [30] to show that two infinite structures agree on all first-order sentences. Fagin, Stockmeyer and Vardi [23] developed a variant of this technique which is applicable in descriptive complexity theory to finite relational structures of universally bounded degree. Variants of this result can be found also in [25] (see also [46]). The essential content of this result, which is denoted also as the Hanf-Sphere Lemma, is that two relational structures of bounded degree fulfill the same first-order sentences of a certain quantifier-rank, if both contain up to a certain number m the same number of isomorphism types of substructures of a bounded radius r.

The paper is organized as follows: section 2 introduces the basic terminology and the notion BIORAM which serves as basis of the linear time computability used in this paper. Section 3 introduces local r-types and handles the case of structures of bounded degree by reducing the general problem to a local investigation of r-types. Some open problems and remarks conclude the paper in section 4.

Martin Kreidler prepared the LaTeX-version of this article. I wish to thank him for his efficient and diligent work. Furthermore, I thank him for his helpful suggestions and careful reading of several versions of the "final" text. Finally, I must express my gratitude to Wolfgang Thomas and his colleagues in Kiel, where I presented and discussed the first version of the main result.

2 Definitions and conventions

This section is devoted to a brief introduction of the basic terminology, Notions from logic or complexity theory not introduced in this or the following sections are standard and the reader is referred e.g. to [10,12,20,37].

A *finite signature* S for relational structures is a finite set of relation symbols $R_1, \ldots, R_s$, each with a fixed arity $r_i > 0$, and constant symbols $c_1, \ldots, c_t$, but without function symbols. An S-structure $\mathcal{G} = (A^{\mathcal{G}}, R_1^{\mathcal{G}}, \ldots, R_s^{\mathcal{G}}, c_1^{\mathcal{G}}, \ldots, c_t^{\mathcal{G}})$ consists of a finite set $A^{\mathcal{G}}$, the domain or universe, from which we assume that it is the set $\{1, \ldots, n\}$ for a natural number n, relations $R_i^{\mathcal{G}}$ over $A^{\mathcal{G}}$ of arity r_i (for each $i : 1 \leq i \leq s$) and elements $c_j^{\mathcal{G}}$ of $A^{\mathcal{G}}$ (for each $j : 1 \leq j \leq t$). The individuals of the domain of a structure are sometimes denoted as *points* or in analogy to graphs as *vertices*. For a structure $\mathcal{G}$ we will denote its domain by $\mid \mathcal{G} \mid$. The number of the elements of an arbitrary set B will be denoted also as $\mid B \mid$, but this will cause no difficulties, since sets and structures can be typographically distinguished. An S-structure $\mathcal{G}$ is called finite in case that its domain is. Unless otherwise stated, throughout the rest of this article we make the assumption that all structures that will be considered are finite. If S is a signature, let $STRUCT(S) = \{\mathcal{G} \ : \ \mathcal{G} \text{ is a finite } S\text{-structure}\}$.

For a subset $B \subseteq A^{\mathcal{G}}$, that contains all $c_j^{\mathcal{G}}$ (with $1 \leq j \leq t$), the *induced substructure* $\mathcal{G} \downarrow B$ is the structure $(B, R_1^{\mathcal{G}} \cap B_1^r, ..., R_s^{\mathcal{G}} \cap B_s^r, c_1^{\mathcal{G}}, \ldots, c_t^{\mathcal{G}})$.

Individuals a and b from $A^{\mathcal{G}}$ are said to be *adjacent* by $R_i^{\mathcal{G}}$, if there are $x_1, \ldots, x_{r_i}$, such that $a = b$ or $((x_1, \ldots, x_{r_i}) \in R_i^{\mathcal{G}}$ and $a = x_j, b = x_k$ for some $j, k \leq r_i)$. We will sometimes denote $(x_1, \ldots, x_{r_i})$ as $R_i^{\mathcal{G}}$*-edge* connecting a and b. a and b are said to be *adjacent in* $\mathcal{G}$ if there is a relation $R_i^{\mathcal{G}}$ such that a and b are adjacent by $R_i^{\mathcal{G}}$. In this case a is said to be *incident* with the corresponding edge $(x_1, ..., x_{r_i})$. The *degree* of an individual a is the cardinality of the set of individuals adjacent to a but not equal to a. The *degree of a structure* is the maximum of the degrees of its individuals. A sequence $x_0, \ldots, x_m$ is called a $\mathcal{G}$*-path*, (or simply a path if it is clear which structure is used), if for every $j < m$, x_j and x_{j+1} are adjacent. m is denoted as the *length* of this path. The *distance* $d^{\mathcal{G}}(a, b)$ between a and b in $\mathcal{G}$ is the length of a shortest path from a to b in $\mathcal{G}$. For $r \geq 0$ the r*-neighbourhood* of x in $\mathcal{G}$, $N_r^{\mathcal{G}}(a)$, consists of all $b \in A^{\mathcal{G}}$ with $d^{\mathcal{G}}(a, b) \leq r$. The superscript $\mathcal{G}$ will be omitted, whenever possible.

The language $\mathrm{L}(S)$ of first-order logic for the signature S contains the relations and constant symbols from S, "=" to denote the equality relation, $\wedge$, $\vee$, $\neg$ as logical connectives "and", "or" and "not", x, y, z, x_1, x_2, x_3,... as individual variables, running over the elements of the domain and $\forall$, $\exists$ as symbols for the quantifiers "for all" and "there exists". The first-order

formulas for this language L(S) are build as usual.

The quantifier-rank of a formula φ, denoted as $qr(\varphi)$ is defined by induction on the structure of formulas:

$$qr(\varphi) := 0 \quad \text{if } \varphi \text{ is atomic ,}$$

$$qr(\varphi \wedge \psi) := max(qr(\varphi), qr(psi)) \ , \quad qr(\varphi \vee \psi) := max(qr(\varphi), qr(psi)) \ ,$$

$$qr(\neg\varphi) := qr(\varphi) \ ,$$

$$qr(\forall x\, \varphi) := qr(\varphi) + 1 \ , \qquad qr(\exists x\, \varphi) := qr(\varphi) + 1 \ .$$

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two S-structures and assume $n \in \mathbf{N}$. We define the relation $\equiv_n$ between structures of the same signature by: $\mathcal{G}_1 \equiv_n \mathcal{G}_2$ if and only if for each L(S)-formula φ with $qr(\varphi) \leq n$ it holds:

$\mathcal{G}_1 \models \varphi$ if and only if $\mathcal{G}_2 \models \varphi$.

We will say that $\mathcal{G}_1$ and $\mathcal{G}_2$ are *n-equivalent* in case that $\mathcal{G}_1 \equiv_n \mathcal{G}_2$ holds. It was one of the trendsetting results of Ehrenfeucht [21] and Fraïssé [24] to give a game-theoretic and an algebraic characterization of n-equivalence. It is introduced in form of a game between two players, which have the possibility to choose alternatively one of the structures and a point in the chosen structure. Player I, the spoiler[1] , starts the game by choosing arbitrarily one of the two structures and a point in its domain. Player II, the duplicator, continues by choosing a point in the other structure. In the second round the spoiler chooses again arbitralily one of the two structures and a point in its domain and the duplicator chooses a point in the other structure, and so on. The game is played in this way over n rounds. Assume that the points selected in $\mathcal{G}_1$ are $a_1, \ldots, a_n$ and the points selected in $\mathcal{G}_2$ are $b_1, \ldots, b_n$, where the index i of a_i, b_i $(1 \leq i \leq n)$ indicates that the point was chosen in round i. Let A be the set $\{a_1, \ldots, a_n\}$ and B $\{b_1, \ldots, b_n\}$. The duplicator wins the game if $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ is an isomorphism between $\mathcal{G}_1 \downarrow A$ and $\mathcal{G}_2 \downarrow B$, otherwise the spoiler wins. In case that the duplicator has a winning strategy, we write $\mathcal{G}_1 \cong_n \mathcal{G}_2$ and call $\mathcal{G}_1$ and $\mathcal{G}_2$ Ehrenfeucht-Fraïssé n-equivalent.

Some of the known results on these relations are collected in the following

Theorem 2.1 *[21,24] (see also [20]): Let S be a finite signature for relational structures and let $\mathcal{G}_1$ and $\mathcal{G}_2$ be S-structures.*

(i) For each $n \in \mathbf{N}$ it holds: $\mathcal{G}_1 \equiv_n \mathcal{G}_2$ if and only if $\mathcal{G}_1 \cong_n \mathcal{G}_2$

(ii) $\equiv_n$ and $\cong_n$ are equivalence relations with finitely many equivalence classes and for each equivalence class , there is an L(S)-formula η_Γ of quantifier-rank n such that

$$, = \{\mathcal{G} : \mathcal{G} \text{ is an } S\text{-structure and } \mathcal{G} \models \eta_\Gamma\}$$

We will think of a problem or a property P as a set of S-structures for a signature S. P is *expressible* or *definable* by a formula φ from L(S) (L(S)-definable for short) if $P = \{\mathcal{G} : \mathcal{G} \text{ is a finite } S\text{-structure and } \mathcal{G} \models \varphi\}$. The latter set is also denoted as $MOD(\varphi)$. Let **FO** be $\{P :$ there is a

[1] According to J. Spencer (1991).

finite signature S and a first-order formula $\varphi \in L(S)$ with $P = MOD(\varphi)\}$. Similarly, we define $\mathbf{FO}(S) := \{P$: there is a first-order formula $\varphi \in L(S)$ with $P = MOD(\varphi)\}$. Aho, Ullman and Immerman proved

Theorem 2.2 *[4,34]:* $\mathbf{FO} \subset \mathbf{L}$.

Here **L** is the set of problems decidable by a deterministic Turing machine in logarithmic space and hence each **FO**-problem can be computed in polynomial time.

It is the objective of this article to show that each **FO**-problem can be decided in linear time for structures of bounded degree. But linear time is not such a robust notion as polynomial time and there is no canonical definition of linear time in literature, since it depends heavily on the used model of computation (see [27], [26], [29]). For this reason, it is necessary to define what is meant by linear time computable problems in this article.

First, we have to define the size of a relational structure $\mathcal{G}$. For that reason, it is not difficult to generalize the adjacency-list representation known as one of the standard representations for graphs. Here we have an entry for each individual of the domain with s pointers to the lists of the r_j-tuples with which the corresponding individual is incident. The r_j-tuples are represented as lists with pointers to the corresponding vertices, while the constants are marked by t additional entries with pointers to and from the corresponding individuals. We will denote this generalized adjacency-list representation also as *gal representation*.

The number of registers needed to store this gal representation of a structure $\mathcal{G}$ will be denoted as $size(\mathcal{G})$, the size of $\mathcal{G}$. Here we assume that the name of each of the individuals fits int o one register, i.e. we follow the idea of the uniform cost measure (see [3]).

It is possible to prove the same results also for the logarithmic cost measure if one assumes that the size of $\mathcal{G}$ is by a multiplicative factor $\log n$ larger, where n is the number of individuals of $\mathcal{G}$.

We will use a very special kind of RAM, which we will denote as BIO-RAM, bounded input output random access machine, which has besides the usual memory and operating registers also a sequence of input registers and a sequence of output registers and which allows moreover the usual operations also with these registers (especially, it is allowed to read and write into the output registers), with the restriction that as arithmetic operations only addition "+" and subtraction "−" are allowed and one of the operands of these operations has to be universally bounded by an arbitrary but fixed constant B.

With this BIORAM we can solve decision problems as well as calculate functions. Let **BIOLIN** be the set of all problems on relational structures which can be decided by a BIORAM in time $O(size(\mathcal{G}))$, starting with the above gal representation of $\mathcal{G}$ in the first cells of the input register. Moreover, let $\mathbf{BIOLINFU}(S)$ be the set of all functions from $STRUCT(S)$ to $STRUCT(S)$ which are computable by a BIORAM in time linear in the size of the input structure $\mathcal{G}$ in such a way that the gal representation of $\mathcal{G}$ is

written in the first $size(\mathcal{G})$ cells of the input register and the output structure $\mathcal{H}$ is written also in gal representation in the first cells of the output register.

It is not the goal of this article to investigate the relation of this model of linear time computation to other models. Here it serves only as an aid to deduce a linear time algorithm. But at first it is obvious that the ordinary **LINTIME**, i.e. the languages decidable by a deterministic Turing machine in linear time, are contained (up to a certain coding) in **BIOLIN**. It is not difficult to see that the model is not too powerful. For that reason we interpret the content of the registers as natural numbers. Let k be the largest number of an input register and assume that the input length, the number of nonempty imput cells, is n. Then the largest number which is represented by an arbitrary register cell after the performance of $c*n$ steps is at most $c*B*n+k$. On the other hand, it is suitably designed to handle problems for relational structures and especially graphs, since some of the well-known linear time computable graph functions can be computed on BIORAMS in linear time.

Lemma 2.3 *Let S be the signature for graphs. Assume that DFS or BFS are the functions computing to a graph one of its depth or breath first search trees. Then DFS and BFS are in* **BIOLINFU***(S). The problem to decide for a graph its connectedness is in* **BIOLIN**.

To prove this lemma one follows simply the standard linear time algorithms (see e.g. [12]) and observes that they can be performed on BIORAMs in linear time. Similarly, one can calculate strongly connected components, decide planarity and perform topological sort.

It is useful to observe that functions computable in linear time by BIORAMs are closed under composition.

Lemma 2.4 *Let S be an arbitrary signature for relational structures. Then $f, g \in$* **BIOLINFU***(S) implies $f \circ g \in$***BIOLINFU***(S).*

This is obvious, since one can easily combine the output of one BIORAM with the input of the other, because it is allowed to rewrite the output it can serve as intermediate memory.

3 Structures of bounded degree and local properties

For an arbitrary natural number $d \geq 1$ let BD_d be the class of finite relational structures of degree $\leq d$ and assume that for an arbitrary class K of relational structures **FO**$\downarrow K$ is defined to be $\{P \cap K \ : \ P \in FO\}$. Now the main result can be formulated:

Theorem 3.1 *For each natural number $d \geq 1$ it holds:*
FO$\downarrow BD_d \subseteq$ **BIOLIN**.

For $d = 1$ the result is almost trivial and for $d = 2$ it is easy for the simplicity of the structures in BD_1 and BD_2. But it is not necessary to handle these cases extra. One of the ingredients in the proof is a technique of Hanf [30], developed to prove the elementary equivalence of infinite structures, which was adapted to finite structures and brought to the attention of the

computer science audience by Fagin, Stockmeyer and Vardi [23] (see also [46]).

Let $\mathcal{G}$ be a relational structure and let a be an element in the domain of $\mathcal{G}$. Moreover, let r be an arbitrary natural number. Hanf defined the r-type of a in $\mathcal{G}$ to be the isomorphism type of $(\mathcal{G} \downarrow N_r^{\mathcal{G}}(a), a)$, i.e. where $(\mathcal{G} \downarrow N_r^{\mathcal{G}}(a), a)$ is assumed to be the restriction of $\mathcal{G}$ to the r-neighbourhood of a in $\mathcal{G}$, where a is designated as value of a new constant, which is denoted as *root* or *origin*. More precisely, individuals a_1 and a_2 in the domain of two structures $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, have the same r-type if $\mathcal{G}_1 \downarrow \mathcal{N}_r^{\mathcal{G}_1}(a_1) \cong \mathcal{G}_2 \downarrow \mathcal{N}_r^{\mathcal{G}}(a_2)$ under an isomorphism mapping a_1 to a_2. r will be also denoted as *radius* of the r-type. A structure representing such a type is denoted as *rooted* S-structure of radius r. Let $r \in \mathbf{N}$ and $m \geq 1$ be given and assume that S is a signature for relational structures. Following Fagin, Stockmeyer and Vardi [23] we define: S-structures $\mathcal{G}_1$ and $\mathcal{G}_2$ are said to be (r, m)-equivalent if and only if for every r-type τ, either $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same number of individuals with r-type τ, or else both have at least m individuals with r-type τ. The following result is due to Fagin, Stockmeyer and Vardi:

Theorem 3.2 *[23]: Let n and d be positive integers. There are positive integers r, m, where r depends only on n, such that whenever $\mathcal{G}_1$ and $\mathcal{G}_2$ are (r,m)-equivalent structures of degree at most d, then $\mathcal{G}_1 \equiv_n \mathcal{G}_2$.*

Normally, an arbitrary **FO**-problem has to be regarded as a global problem, i.e. a problem which can only be decided by examining simultaneously different locations of the regarded structure, which are usually far away from each other. Theorem 3.2 is one of the essential ingredients of the proof of our Theorem 3.1, since it enables us to reduce an arbitrary **FO**-problem to a *local* problem (see also [42] for related aspects of more expressive languages) , i.e. a problem which can be decided by *visiting once* each vertex of the structure and looking only to its neighbourhood of a certain *fixed radius*.

Proof of Theorem 3.2: Assume that φ is an L(S)-formula for a signature S of relational structures. Let n be the quantifier-rank of φ and let d be a natural number, the upper bound of the degree of our structures. Let , be an equivalence class of $\equiv_n$. Hence, by Theorem 2.1 it is also an equivalence class for $\cong_n$. If φ holds in one of the structures $\mathcal{G}$ from , , i.e. $\mathcal{G} \models \varphi$, then it holds in all, since $qr(\varphi) \leq n$. In this case, we will say that φ holds in , . Let $, _1, \ldots, , _{k_1}$ be an enumeration of all equivalence classes for $\equiv_n$ in which φ holds and let $\Lambda_1, ..., \Lambda_{k_2}$ be an enumeration of all equivalence classes for $\equiv_n$ in which φ does not hold. Then $, _1, \ldots, , _{k_1}, \Lambda_1, \ldots, \Lambda_{k_2}$ is a complete enumeration of all equivalence classes of $\equiv_n$.

Now, choose r and m as in Theorem 3.2 for the above n and d. There is only a finite number of r-types for S-structures of degree $\leq d$, since there is a universal bound for the number of vertices. Let $\tau_1, \ldots, \tau_p$ be a complete enumeration of all possible r-types for S-structures. Each of these r-types τ_i can be represented by a structure $\mathcal{H}_i$. One should notice that these structures are almost S-structures, since they have one constant additional to the signature, the origin, or root. Hence, the enumeration $\tau_1, \ldots, \tau_p$ can be effectively determined by investigating all possible S-structures of radius $\leq r$.

Now, two S-structures $\mathcal{G}_1$ and $\mathcal{G}_2$ are (r,m)-equivalent if and only if for all i, with $1 \leq i \leq p$ the following holds:

$$\Big(\quad | \{a \ : \ a \in | \mathcal{G}_1 | \ \text{ and } (N_r^{\mathcal{G}_1}(a), a) \text{ is of } r\text{-type } \tau_i | = | \{b \ : \ b \in | \mathcal{G}_2 | \ \text{ and } (N_r^{\mathcal{G}_2}(b), b) \text{ is of } r\text{-type } \tau_i | \quad \Big)$$

or

$$\Big(| \{a \ : \ a \in | \mathcal{G}_1 | \ \text{ and } (N_r^{\mathcal{G}_1}(a), a) \text{ is of } r\text{-type } \tau_i | \geq m \text{ and } | \{b \ : \ b \in | \mathcal{G}_2 | \ \text{ and } (N_r^{\mathcal{G}_2}(b), b) \text{ is of } r\text{-type } \tau_i | \geq m \Big)$$

For an arbitrary S-structure $\mathcal{G}$ define $\tau(\mathcal{G})$ to be the p-tuple $(\varrho_1, \ldots, \varrho_p)$ by:

$$\rho_i := \begin{cases} | \{a \ : \ a \in | \mathcal{G} | \ \text{ and } (N_r^{\mathcal{G}}(a), a) \text{ is of } r\text{-type } \tau_i\} | & \text{if it is } < m \\ m & \text{otherwise} \end{cases}$$

Using this notion, we get that two S-structures $\mathcal{G}_1$ and $\mathcal{G}_2$ are (r,m)-equivalent if and only if $\varrho(\mathcal{G}_1) = \varrho(\mathcal{G}_2)$. Obviously, (r,m)-equivalence is an equivalence relation on the set of S-structures. But the number of such p-tuples $\tau(\mathcal{G})$ is bounded by $(m+1)^p$, since only $0, 1, \ldots, m$ appear as components. Hence, there are at most $(m+1)^p$ equivalence classes of S-structures for the (r,m)-equivalence relation. Each such (r,m)-equivalence class can be represented by a p-tuple $\tau(\mathcal{G})$ for a suitable S-structure $\mathcal{G}$. Note that not each such p-tuple can represent a real S-structure, since some are eventually not realizable. By Theorem 3.2, the equivalence relation for (r,m)-equivalence is a subdivision of the equivalence relation for $\equiv_n$. Let $\mathcal{U}_1, \ldots, \mathcal{U}_q$ with $q \leq (m+1)^p$ be a maximal system of pairwise not (r,m)-equivalent S-structures, i.e. a complete system of representatives for each (r,m)-equivalence class. Now select those of these representatives $\mathcal{U}_i$ in, which φ holds. Assume that $\mathcal{V}_1, \ldots, \mathcal{V}_{q_1}$ is a complete system of representatives for these. The structures $\mathcal{V}_1, \ldots, \mathcal{V}_{q_1}$ can be chosen as representatives for the $(\equiv_n)-$equivalence classes $, _1, \ldots, , _{k_1}$, where some of these classes have many representatives among the structures $\mathcal{V}_1, \ldots, \mathcal{V}_{q_1}$, since (r,m)-equivalence refines $(\equiv_n)$. Such a system can be effectively determined if $\mathcal{U}_1, ..., \mathcal{U}_q$ is given. The result will follow then from the next two statements:

$(*)$ For an arbitrary S-structure $\mathcal{G}$ it holds: $\mathcal{G} \models \varphi$ if and only if there is an i with $1 \leq i \leq q_1$ such that $\mathcal{G}$ and $\mathcal{V}_i$ are (r,m)-equivalent.

To prove $(*)$ assume first $\mathcal{G} \models \varphi$. There exists a j with $1 \leq j \leq q$ such that $\mathcal{G}$ is (r,m)-equivalent to $\mathcal{U}_j$, since $\mathcal{U}_1, \ldots, \mathcal{U}_q$ is a complete system of representatives for (r,m)-equivalence. By Theorem 3.2 it holds that $\mathcal{U}_j \models \varphi$. But then $\mathcal{U}_j$ is one of the $\mathcal{V}_i$ for a certain i with $1 \leq i \leq q_1$. For the other

direction assume that i with $1 \leq i \leq q_1$ is chosen such that $\mathcal{G}$ and $\mathcal{V}_i$ are (r,m)-equivalent. By Theorem 2.1 it holds that $\mathcal{V}_i \equiv_n \mathcal{G}$ and hence $\mathcal{G} \models \varphi$ since $qr(\varphi) \leq n$ and $\mathcal{V}_i \models \varphi$, what proves $(*)$.

$(\diamond)$ For an arbitrary S-structure G the following holds: $\mathcal{G} \models \varphi$ if and only if there is an i with $1 \leq i \leq q_1$ such that $\varrho(\mathcal{G}) = \varrho(\mathcal{V}_i)$.

This is a simple combination of the above statement on ϱ and $(*)$. The algorithm wanted is now the following:

The input is an S-structure $\mathcal{G}$ of degree $\leq d$.

FOR $i := 1$ **TO** p **DO**
  $\mu_i := 0$
  $\mu := (\mu_i)_{1 \leq i \leq p}$
END $(*\text{FOR}*)$
FOR each vertex $a \in |\mathcal{G}|$ **DO**
  determine $N_r^{\mathcal{G}}(a)$
  FOR $i := 1$ **TO** p **DO**
    IF $[N_r^{\mathcal{G}}(a)$ has r-type τ_i and $\mu_i \leq m]$
      THEN $\mu_i := \mu_i + 1$
    END $(*\text{ IF }*)$
  END $(*\text{ FOR }*)$
  IF $\mu \in \{\varrho(\mathcal{V}_1), \ldots, \varrho(\mathcal{V}_{q_1})\}$
    THEN print "$\mathcal{G} \models \varphi$"
    ELSE print "$\mathcal{G} \models \neg\varphi$"
  END $(*\text{ IF }*)$
END $(*\text{ FOR }*)$.

The correctness of this algorithm follows immediately from $(\diamond)$. It is not difficult to see that it can be performed on a BIORAM in time linear in $size(\mathcal{G})$ in case that $\mathcal{G}$ is given in gal-representation. At first, note that the size of $N_r^{\mathcal{G}}(a)$ is universally bounded (i.e. independent of the input $\mathcal{G}$) for S-structures $\mathcal{G}$ of degree $\leq d$. Hence, also the size of each of the structures $\mathcal{V}_1, \ldots, \mathcal{V}_{p_1}$ is universally bounded.
Moreover, the checking whether $N_r^{\mathcal{G}}(a)$ has r-type τ_i can also be performed in a universally bounded time, independent of $\mathcal{G}$, a and i. But p and q_1 are also fixed and each component of the vector $\varrho(\mathcal{V}_i)$ is $\leq m$, hence also the test $\mu \in \{\varrho(\mathcal{V}_1), \ldots, \varrho(\mathcal{V}_{q_1})\}$ can be performed in constant time. All the μ_i are $\leq m$. Hence, also the addition $\mu_i := \mu_i + 1$ and the test $\mu_i \leq m$ can be performed on our BIORAM in constant time. Therefore, the time of each step (each new vertex a) of the main **FOR**-loop can be universally bounded by a constant. But each vertex a (i.e. together with its neighbourhood) is regarded exactly once, hence the algorithm works in linear time on a BIORAM if $\mathcal{G}$ is

given in gal-representation. This proves Theorem 3.2. □

Remark 3.3 *It is not difficult to see that a similar result can be proved if instead of the uniform cost measure the logarithmic cost measure is regarded. But then one has to measure the time complexity of the algorithm in the logarithmic size of the input structure $\mathcal{G}$.*

The result is surprising, since on the first view one would expect a polynomial time algorithm to decide for a given (arbitrary) structure the truth of a given (but fixed) formula with a large block of nested quantifiers. The result shows the usefulness of Theorem 3.2 which can be regarded as a general principle to reduce global (first-order) problems to local problems (on neighbourhoods).

4 Concluding remarks

Obviously, there are linear time computable problems which are not expressible in first-order logic. One of those problems is for instance EVEN, the problem to determine, whether the domain of a given structure has even cardinality or the problem of the connectedness of a graph.

The algorithm presented in this article works in linear time, but it is not practical since the hidden constants grow exponentially. But as long as the **P**≠**NP**–problem is undecided, there is no hope to avoid this disadvantage.

Theorem 4.1 *If* **P**≠**NP**, *then there is no polynomial time algorithm to solve the above problem whose hidden constants are polynomially bounded.*

This can easily be seen by coding the satisfiability problem of an arbitrarily given propositional formula with n propositional variables into a first-oder formula with n existential quantifiers in the signature with only one unary relation, denoted e.g. as Q. Each propositional variable p then corresponds to $Q(x_p)$ and each negated propositional variable $\neg q$ corresponds to $\neg Q(x_q)$, where x_p and x_q are new individual variables. The propositional connectives $\neg$, $\vee$ and $\wedge$ are translated by themself. The new introduced individual variables x_p (for each propositional variable of the original proposition) are quantified at the beginning of the transformed formula by existential quantifiers $(\exists x_q)$. The original propositional formula is satisfiable if the transformed first-oder formula is true in a very simple special model, which has $2n$ isolated vertices, n are in the unary relation and n are not. The truth of the transformed formula in the special model could be decided in polynomial time in case that the hidden constants in the algorithm would be polynomially bounded in the size of the input structure, hence polynomially bounded in n in this case. But then we have **P**=**NP**, which contradicts our assumption.

Even if one accepts these large constants as inevitable the presented method is disadvantageous, since the algorithm depends effectively on the key structures $\mathcal{V}_1, \ldots, \mathcal{V}_{p_1}$, which are very difficult to find. Here is an analogy to the results of Robertson and Seymour's polynomial time algorithms for minor closed classes of graphs [38]. But in contrast to these results, here one can show that the key structures cannot be found algorithmicly, since then the

Erfüllbarkeitsproblem der Prädikatenlogik (satisfiability problem of predicate logic) could be solved. Hence, the above method cannot be automatized. More precisely, we have the following

Theorem 4.2 *There is no algorithm which takes as input an arbitrary first-order formula φ and calculates the key structures $\mathcal{V}_1, \ldots, \mathcal{V}_{p_1}$ in the above algorithm.*

To see this, one has simply to observe that if one could compute these structures for an arbitrary first-order formula φ, then one could solve the satisfiability problem for finite structures, which is impossible by Trahtenbrot's Theorem (see [20]). It is not difficult to observe that this result holds also for relational structures of bounded degree. But to decide finite satisfiability it is only necessary to know whether $p_1 = 0$ or not. Hence, there is no algorithm which computes the key structures for an arbitrary first-order formula.

It seems to be open, whether there is a different method which is able to construct to an arbitrarily given first-order formula a linear time algorithm to solve the algorithmic problem defined by the formula for structures of bounded degree, and which could at least theoretically be automatized, of course with exponentially growing hidden constants.

Moreover, the question **FO** $\subset$ **BIOLIN** ? seems to be open. In this unbounded case one is not able to investigate isomorphism types of local neighbourhoods, since they can become very large, in the worst case they can contain the whole structure.

References

[1] S. Abitoul, M. Y. Vardi, V. Vianu, *Fixpoint logics, relational machines, and computational complexity*, IEEE FOCS 1992, pp. 156-192.

[2] S. Arnborg, B. Courcelle, A. Proskurowski and D. Seese, *An algebraic theory of graph reduction*, Technical Report LaBRI-90-02, Universite de Bordeaux. Jan. 1990, JACM, vol. 40 No. 5, 1993 pp. 1134-1164. (Extended abstract in Lec. Notes Comp. Sci. 532, 1991, pp. 70-83.)

[3] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The design and analysis of computer algorithms*, Addison Wesley, 1974.

[4] A. V. Aho, J. D. Ullman, *Universality of Data Retrieval Languages*, 6th Symposium on Principles of Programming Languages, 1979, pp. 110-117.

[5] M. Ajtai, R. Fagin, *Reachability is harder for directed than for undirected finite graphs*, Journal of Symbolic Logic, 55 (1), 1990, pp. 113-150.

[6] S. Arnborg, J. Lagergren, and D. Seese, *Easy problems for tree-decomposable graphs*, Journal of Algorithms 12, 1991, pp. 308-340.

[7] A. Baudisch, D. Seese, P. Tuschik, M. Weese, *Decidability and Quantifier-Elimination*, in Model-Theoretic Logics, ed. by J. Barwise and S. Feferman, Springer-Verlag, New York 1982, pp. 235-268.

[8] M.W. Bern, E.L. Lawler, and A. L. Wong, *Linear time computation of optimal subgraphs of decomposable graphs*, J. Algorithms 8, 1987, pp. 216-235.

[9] H.L. Bodlaender, *Dynamic programming on graphs with bounded tree-width*, MIT/LCS/TR-394, MIT, 1987.

[10] E. Börger, *Berechenbarkeit Komplexität Logik*, Vieweg 1992.

[11] R.B. Borie, R. G. Parker, and C.A. Tovey, *Automatic generation of linear algorithms from predicate calculus descriptions of problems on recursively constructed graph families*, manuscript, July 21, 1988.

[12] T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, *Introduction to Algorithms*, The MIT Press, Cambridge, 1990.

[13] B. Courcelle, *The monadic second order logic of graphs. III. Tree-width, Forbidden Minors, and Complexity Issues*, Report I-8852, Bordeaux-1 University, 1988.

[14] B. Courcelle, *Recognizability and second-order definability for sets of finite graphs*, Information and Comput. 85, 1990, pp. 12-75.

[15] B. Courcelle, *Monadic second-order logic and hypergraph orientation*, In Proceedings Eighth annual IEEE Symposium on Logic in Computer Science, June 19-23, 1993, Montreal, Canada, pp. 179-190.

[16] B. Courcelle, M. Mosbah, *Monadic second-order evaluations on tree-decomposable graphs*, Theoretical Computer Science 109, 1993, pp. 49-82.

[17] S.S. Cosmadakis, *Logical reducibility and monadic NP*,1993 IEEE, pp. 52-61.

[18] N. Creignou, *The class of problems that are linearly equivalent to Satisfiability or a uniform method for proving* **NP***–completeness*, Lecture Notes in Computer Science CSL'92, San Miniato, 1993, pp. 115-133.

[19] R. G. Downey, M. R. Fellows, *Fixed-Parameter Intractability (Extended Abstract)*, 1992 , pp. 36-49.

[20] H.-D. Ebbinghaus, J. Flum, and W. Thomas, *Mathematical Logic*, Second Edition, Springer-Verlag, New York, 1993

[21] A. Ehrenfeucht, *An application of games to the completeness problem for formalized theories*, Fund. Math., 49, 1961, pp. 129-141.

[22] R. Fagin, *Generalized First-Order spectra and polynomial-time recognizable sets*, In Complexity of Computation, (ed. R. Karp) SIAM-AMS Proc. 7, 1974, pp. 27-41

[23] R. Fagin, L. Stockmeyer, and M. Vardi, *On monadic NP vs. monadic co-NP*, In The Proceedings of the 8th Annual IEEE Conference on Structure in Complexity Theory, 1993, pp. 19-30.

[24] R. Fraïssé, *Sur quelques classifications des systèmes de relations*, Publ. Sci. Univ. Alger. Sr. A, 1, 1954, pp. 35-182.

[25] H. Gaifman, *On local and nonlocal properties*, In J. Stern, editor, Logic Colloquium'81, North-Holland, 1982, pp. 105-135.

[26] E. Grädel, *On the notion of linear time computability*, Proc. 3rd Italian Conf. Theoret. Comput. Sci. 1989, World Scientific Publ. Co. pp. 323-334, also appear in Internat. J. Foundations Comput. Sci. 1, 1990 pp. 295-307.

[27] E. Grandjaen, *Linear time algorithms and* **NP***–complete problems*, Lecture Notes in Computer Science CSL'92, San Miniato, 1993, pp. 248-273.

[28] Y. Gurevich, *Logic and the challenge of computer science*, Univ. Michigan Tech. Report CRL-TR-10-85, Sept. 1985.

[29] Y. Gurevich, and S. Shelah, *Nearly linear time*, Springer Lecture Notes in Computer Science 363, pp. 108-118. 86.

[30] W. Hanf, *Model-theoretic methods in the study of elementary logic*, In J. Addison, L. Henkin, and A. Tarski, editors, The Theory of Models, North Holland, 1965, pp. 132-145.

[31] W. Hohberg, R. Reischuk, *A framework to design algorithms for optimization problems on graphs*, Preprint, April 1990.

[32] N. Immerman, *Upper and Lower Bounds for First Order Expressibility*, JCSS, Vol. 25, No. 1, August 1982.

[33] N. Immerman, *Relational queries computable in polynomial time*, Information and Control 68, 1986, pp. 86-104.

[34] N. Immerman, *Languages that capture complexity classes*, SIAM J. Comput., Vol. 16, No. 4, August 1987, pp. 760-778. A preliminary version of this paper appeared in the 15th ACM STOC Symp., 1983, 347-354.

[35] N. Immerman, *Descriptive and computational complexity*, Computational Complexity Theory, Proc. Symp. Applied Math. Vol. 38 (J. Hartmanis, ed.), 1989, pp. 75-91.

[36] C. Lautemann, *Logical definability of NP-optimisation problems with monadic auxiliary predicates*, preprint 1990, pp. 327-339.

[37] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.

[38] N. Robertson, and P.D. Seymour, *Graph minors XIII; The disjoint path problem*, Journal of Combinatorial Theory (to appear).

[39] M. de Rougemont, *Second-order and inductive definability on finite structures*, Zeitschrift f. math. Logik und Grundlagen d. Math., Bd. 33, 1987, pp. 47-63.

[40] P. Scheffler, *Die Baumweite von Graphen als ein Mass fuer die Kompliziertheit algorithmischer Probleme*, Dissertation (A), AdW d. DDR, Berlin 1989.

[41] T. Schwentick, *Graph connectivity and monadic NP*, Informatik-Bericht Nr. 2/94, Johannes Gutenberg - Universität Mainz, Institut für Informatik, FB 17, 24. August 1994, p. 1-33.

[42] D. Seese, *Tree-partite graphs and the complexity of algorithms*, Preprint P-MATH-08/86, Karl-Weierstrass-Institute for Mathematics, Academie of Sciences, Berlin 1986, an extended abstract appeared also in FCT'85, edited by L. Budach, Lecture Notes in Computer Science 199, 1985, pp. 412-421.

[43] D. Seese, *The structure of the models of decidable monadic theories of graphs*, Annals of Pure and Applied Logic 53, 1991, pp. 169-195.

[44] D. Seese, *Interpretability and tree automata: a simple way to solve algorithmic problems on graphs closely related to trees*, in Tree Automata and Languages, edited by M. Nivat and A. Podelski, Elsevier Science Publishers, 1992, pp. 83-114.

[45] D. Seese, *FO-Problems and Linear Time Computability*, Bericht 315, AIFB Institut fuer Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe (TH), April 1995, pp. 1-20.

[46] W. Thomas, *On logics, tilings, and automata*, In Proc. 18th ICALP, Lecture Notes in Computer Science 510, Springer Verlag, 1991, pp. 441-454.