S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 11, стр. 557–566 (2014) УДК 510+519.6
MSC 03C40, 03C57

ON THE EXISTENTIAL INTERPRETABILITY OF STRUCTURES

A.S. MOROZOV, A.ZH. SATEKBAEVA, J.A. TUSSUPOV

Abstract. We introduce and study the notion of ∃–interpretability of constructive algebraic structures. It is shown that any finite partially ordered set is embeddable into the semilattice this interpretability generates; we also prove the existence of universal computable structures. As an application of this concept, we consider the transformations of abstract databases and their queries in case when one data structure is ∃–interpretable in another one.

Keywords: existential interpretability, definability, computable structure, constructive structure, semilattice.

1. Introduction

In this paper we consider a partial case of the concept of elementary definability (or interpretability) of a structure in another structure (see, for instance, [3]) which will be called ∃–interpretability (or existential interpretability). It should be noted that this interpretability is a partial case of Σ–interpretability (definability) of a structure in a hereditarily finite superstructure over another structure (see [4]), which enables us to effectively construct the structure we define provided that we have some effective realization of the initial structure.

In what follows, we will consider finite predicate signatures only; here we assume that operations are viewed as their graphs. We denote the length of a tuple x by $|\bar{x}|$.

Morozov A.S., Satekbaeva A.Zh., Tussupov J.A., On existential interpretability of structures.

The second author was supported by the grant «Computability and algebraic strustures» and the third author was supported by the grant «Formal specification methods and their applications in the development of programs» (#144, 04.02.2014) of the Ministry of Education and Science of the Republic of Kazakhstan.

Received May, 12, 2014, published July, 27, 2014.

Recall that a structure $\mathfrak{M}$ of a finite predicate signature together with a numbering $\nu : \omega \xrightarrow{onto} \mathfrak{M}$, i.e., the pair $\langle \mathfrak{M}, \nu \rangle$ is called a *constructive structure* if the relation $\{\langle x, y\rangle \mid \nu x = \nu y\}$ is computable and for each its predicate symbol P, the relation $\{\langle x_1, \dots, x_n\rangle \mid P(\nu x_1, \dots, \nu x_n)\}$ is computable as well. In this case, ν is called a *constructivization of* $\mathfrak{M}$. If a structure possesses a constructivization then we call it *constructivizable.* This definition could be naturally extended to many–sorted structures.

The following basic definition of this paper is a simple modification of the well-known definition of elementary definability of a structure in another structure:

Definition 1. *Assume that $\mathfrak{A}_0$ and $\mathfrak{A}_1$ are two structures of finite predicate signatures and let $\langle P_1, \dots, P_k\rangle$ be the signature of $\mathfrak{A}_0$. We say that $\mathfrak{A}_0$ is ∃–interpretable (or is existentially interpretable) in $\mathfrak{A}_1$, if there exist*

- *$n \in \omega$ and a finite tuple of parameters $\bar{p} \in \mathfrak{A}_1$*
- *∃–formula $U(\bar{x}, \bar{y})$ such that $|\bar{x}| = n$ and $|\bar{y}| = |\bar{p}|$*
- *∃–formulas $E^+(\bar{x}_0, \bar{x}_1, \bar{y})$ and $E^-(\bar{x}_0, \bar{x}_1, \bar{y})$ such that $|\bar{x}_0| = |\bar{x}_1| = n$*
- *∃–formulas $P^+(\bar{x}_1, \dots, \bar{x}_m, \bar{y})$ and $P^-(\bar{x}_1, \dots, \bar{x}_m, \bar{y})$, $|\bar{x}_1| = \dots = |\bar{x}_m| = n$, for each predicate symbol P of signature $\mathfrak{A}_0$, where m is the number of arguments of P*

such that if $A = \{\bar{x} \mid \mathfrak{A}_1 \models U(\bar{x}, \bar{p})\}$ then

(1) *The set A^2 is a disjunctive union of the sets $\{\langle \bar{x}_0, \bar{x}_1\rangle \mid \mathfrak{A}_1 \models E^\varepsilon(\bar{x}_0, \bar{x}_1, \bar{p})\}$, $\varepsilon \in \{+, -\}$.*

(2) *For each m–placed predicate symbol P of $\mathfrak{A}_0$, the set A^m is a disjunctive union of the sets $\{\langle \bar{x}_0, \dots, \bar{x}_m\rangle \mid \mathfrak{A}_1 \models P^\varepsilon(\bar{x}_0, \dots, \bar{x}_m, \bar{p})\}$, $\varepsilon \in \{+, -\}$.*

(3) *Let $\widehat{P}_i = \{\langle \bar{x}_1, \dots, \bar{x}_m\rangle \mid \mathfrak{A}_1 \models P^+(x_1, \dots, \bar{x}_m, \bar{p})\}$, for $i = 1, \dots, k$. Then $E = \{\langle \bar{x}_0, \bar{x}_1\rangle \mid \mathfrak{A}_1 \models E^+(\bar{x}_0, \bar{x}_1, \bar{p})\}$ is a congruence on the structure $\mathfrak{B} = \left\langle A, \widehat{P}_1, \dots, \widehat{P}_k \right\rangle$ and there exists an isomorphism α from the quotient $\mathfrak{B}/E$ onto $\mathfrak{A}_0$.*

The notation $\mathfrak{A} \prec_\exists \mathfrak{B}$ will mean that $\mathfrak{A}$ is ∃–interpretable in $\mathfrak{B}$.

An important property of the ∃–interpretability (which is also true for Σ–defnability) is that if $\mathfrak{A} \prec_\exists \mathfrak{B}$ and ν is a constructivization of $\mathfrak{B}$ then we could naturally transform the constructivization ν into some constructivization ν^* of $\mathfrak{A}$. We leave the details to the reader.

We leave proofs of the following two theorems to the reader.

Theorem 1. *∃–Interpretability is transitive:*

$$\mathfrak{A}_0 \prec_\exists \mathfrak{A}_1 \prec_\exists \mathfrak{A}_2 \text{ implies } \mathfrak{A}_0 \prec_\exists \mathfrak{A}_2.$$

Theorem 2. *For any structures $\mathfrak{A}_0$ and $\mathfrak{A}_1$, there exists a structure $\mathfrak{A}^*$ with the following properties:*

(1) *$\mathfrak{A}_0, \mathfrak{A}_1 \prec_\exists \mathfrak{A}^*$;*

(2) *If a structure $\mathfrak{A}$ of a finite signature has the property $\mathfrak{A}_0, \mathfrak{A}_1 \prec_\exists \mathfrak{A}$ then $\mathfrak{A}^* \prec_\exists \mathfrak{A}$.*

Taking into account that $\prec_\exists$ is obviously reflexive, we obtain that the ∃–interpretability on the set of at most countable (all constructivizable) structures of finite predicate signatures whose basic sets are subsets of a fixed countable set defines a preordering whose factor modulo the equivalence $\prec_\exists \cap (\prec_\exists)^{-1}$ forms an upper semilattice which will be denoted by $\mathcal{L}_\exists$ (respectively $\mathcal{L}^0_\exists$). Note that $\mathcal{L}^0_\exists$ is an ideal of the semilattice $\mathcal{L}_\exists$.

2. ∃–Interpretability of data structures and databases

As an argument in favor of the study of ∃–interpretability, here we give an example of application of ∃–interpretability, namely, we demonstrate a good behavior of databases and database queries in case one data structure is ∃–interpreted in another one. The definitions of databases and data structures we use here are variations of the definitions definitions used in literature; the definitions we give here neither claim to be original, nor general or even fully adequate. Nevertheless, we tried to follow the spirit of model theory as much as possible.

We consider databases in the framework of a relational approach, i.e., we assume that we have in our disposal an a priori given *basic algebraic data structure*, which could be many–sorted; its elements are used to form *records*, i.e., elementary facts that could be stored in a database. Some records of similar types will form *tables* which reflect the current state of a database, etc. For instance, the basic structure could have the following sorts (basic sets): symbols of the alphabet of some concrete language, words on these symbols, and natural numbers. It could also have the following operations: the concatenation operation on words, the length function on these words, whose values are natural numbers, addition and multiplication operations on the natural numbers. We can also consider the following relations: the usual ordering on the natural numbers, the relation 'a symbol x occurs in a word y', the relation 'x is the first symbol of a word y', etc.

Here it will be convenient to replace the basic structure (which in general case is a many–sorted structure) with its presentation by a predicate one–sorted structure, which is the usual practice in model theory. The universe of this new structure is the union of all universes of the initial structure; each sort is distinguished by some additional predicate, and all the operations are replaced by their graphs.

Inasmuch as all the basic data structures could be represented on a computer, their elements could be naturally associated with natural numbers so that all the basic relations will be computable on these numbers. By this we may assume all our structures to be constructive.

An ordered pair $\sigma = \langle \sigma_R, \sigma_N \rangle$ in which σ_R is a finite predicate signature and σ_N is a finite set of names for fields will be called a *database signature* or *a database signature over* σ_R. Any subset of σ_N will be called a *table type* or *type of a table* of the signature σ.

Assume that $\sigma = \langle \sigma_R, \sigma_N \rangle$ is a database signature, $\mathfrak{M}$ is a structure of signature σ_R, and τ is a table type of σ. Any mapping r from τ to the basic set of the structure $\mathfrak{M}$ will be called a *line (or record) of type* τ. Any (finite) set of records of type τ is called a (finite) *table of type* τ.

A *(constructive) database of signature* $\sigma = \langle \sigma_R, \sigma_N \rangle$ is any (constructive) structure of signature σ_R together with some family of tables over it.

To formalize the concept of query to a database, first we have to define a language $\mathtt{DBL}_\sigma$ of a database signature σ, which is a simple natural extension of the first order language of signature σ_R.

Let σ be a database signature. Fix the following countable families of variables:

V: for elements of the data structure,

S_τ**:** for records of type τ , for each table type τ of signature σ,

T_τ**:** for tables of type τ, for each table type τ of signature σ.

We assume the sets V, S_τ, T_τ to be mutually disjoint.

First we define the notion of *simple expression.* There are exactly two kinds of simple expressions:

(1) $x.N$, where $x \in S_\tau$ and $N \in \tau$, (it will be viewed as $x(N)$, the value of a field named N in the record x)
(2) any variable $x \in V$.

Atomic formulas of $\mathtt{DBL}_\sigma$ could be of one of the following kinds:

(1) $P(A_1, \dots, A_s)$, where P is an arbitrary s–ary relational symbol of σ_R and all A_i, $i = 1, \dots, s$ are simple expressions;
(2) $A = B$, where A and B are simple expressions;
(3) $A = B$, where A and B are record variables of the same type;
(4) $v \in t$, where $v \in S_\tau$ and $t \in T_\tau$.

Formulas of $\mathtt{DBL}_\sigma$ are built from atomic formulas by means of propositional connectives and universal and existental quantification over variables from V and S_τ (we do not consider quantifications over tables!). The definitions of free and bounded variables and the concept of truth for formulas of $\mathtt{DBL}_\sigma$ on valuations of its free variables by objects of appropriate types are defined in a standard way. Each time we use the notation $\varphi(x_1, \dots, x_n)$, where $x_1, \dots, x_n$ are variables, we mean that all the free variables of φ are contained in the list $x_1, \dots, x_n$. Note also that in this definition, item (3) could be omitted because if A and B are variables of type τ then the formula $A = B$ is equivalent to $\bigwedge_{N\in\tau}(A.N = B.N)$.

Now we are going to define a notion of *query.* Intuitively, we assume that a typical query uses some information from a finite tuple of tables $\overline{H} = H_1, \dots, H_n$ of a database, some values $\overline{c} = c_0, \dots, c_m$ from the basic data structure, the so–called *parameters of the query,* and that the result of execution of a query is a table, i.e., the set of records satisfying some conditions referring to $\overline{H}$ and $\overline{c}$. These conditions together with tables $\overline{H}$ and parameters $\overline{c}$ completely define this query. Thus, it is our understanding that a query is completely defined by the following elements:

- a variable $v \in S_\tau$ that denotes records belonging to the results of execution of a query;
- a tuple of parameters $\overline{c}$
- a formula $\varphi(v, \overline{H}, \overline{y})$ of the language $\mathtt{DBL}_\sigma$ that describes the conditions records v should satisfy (with parameters $\overline{y} = \overline{c}$).

Thus, we naturally arrive at the following formal definition:

Definition 2. *Let $\mathfrak{M}$ be a structure of signature σ_R and $\mathfrak{B}$ a database of signature $\sigma = \langle \sigma_R, \sigma_N \rangle$.*

A query to the database $\mathfrak{B}$ is a formula $\varphi(v, \overline{H}, \overline{c}) \in \mathtt{DBL}_\sigma$ with free variable v of record type, table parameters $\overline{H}$ of this database, and parameters $\overline{c}$ from the basic data structure.

The result of execution of the query $\varphi(v, \overline{H}, \overline{c})$ is a table

$$\{v \mid \mathfrak{M} \models \varphi(v, \overline{H}, \overline{c})\}.$$

A query $\varphi(v, \overline{H}, \bar{c})$ is called an $\exists$–*query* if $\varphi(v, \overline{H}, \bar{c})$ is equivalent to an $\exists$–formula. For any constructive basic data structure and any tuple $\overline{H}$ of finite tables, one can suggest an algorithm that enumerates records satisfying this query (more exactly, enumerates the tuples of ν–numbers of their elements). This algorithm actually simultaneously enumerates all possible values of variables bounded by existential quantifiers and all possible values of v and waits for satisfaction of the rest quantifier–free part of $\varphi(v, \overline{H}, \bar{c})$.

Here we show, how could one translate databases and $\exists$–queries of one database into queries of another database provided that there exists an $\exists$–interpretation of its data structure in the data structure of another database. Actually, theorem 3 below will be also used to obtain some more results.

Assume that $\mathfrak{A}_0$ and $\mathfrak{A}_1$ are structures of signatures σ_0 and σ_1 respectively, $\mathfrak{A}_0 \prec_\exists \mathfrak{A}_1$, and n, $\bar{p}$, U, E^+, E^-, P^+, P^- and α are taken as in the definition of $\exists$–interpretability. Let $\overline{\sigma}_0 = \langle \sigma_0, \sigma_0' \rangle$ be some database signature over σ_0. Define a database signature $\overline{\sigma}_1$ as

$$\overline{\sigma}_1 = \langle \sigma_1, \sigma_0' \times \{1, \ldots, n\} \rangle .$$

For short, denote the elements $\langle N, i \rangle$ from $\sigma_0' \times \{1, \ldots, n\}$ by $N^{(i)}$. Actually, for each name in σ_0', $\overline{\sigma}_1$ contains n new names related to it. For each table type $\tau = \{N_1, \ldots, N_k\}$ from $\overline{\sigma}_0$, define a table type

$$\tau * n = \{N_1^{(1)}, \ldots, N_1^{(n)}, \ldots, N_k^{(1)}, \ldots, N_k^{(n)}\}$$

of signature $\overline{\sigma}_1$, which is obtained by taking n copies of the field names of type τ.

Define a mapping $\overline{\alpha} : M \subseteq \mathfrak{A}_1^n \to \mathfrak{A}_0$, where $M = \{\bar{x} \mid \mathfrak{A}_1 \models U(\bar{x}, \bar{p})\}$, as follows:

$$\overline{\alpha}(x_1, \ldots, x_n) = \alpha \left(\langle x_1, \ldots, x_n \rangle /_E \right) .$$

This mapping could be naturally extended to records. Assume that x is a record of type $\tau * n$ over $\mathfrak{A}_1$ and

(1) *for all* $N \in \tau$ *holds* $\mathfrak{A}_1 \models U\left(x.N^{(1)}, \ldots, x.N^{(n)}, \bar{p}\right)$.

Then we can define a record $\overline{\alpha}(x)$ of type τ as follows:

$$\overline{\alpha}(x).N = \overline{\alpha}\left(x.N^{(1)}, \ldots, x.N^{(n)}\right), \ \textit{for all } N \in \tau.$$

If (1) fails to be true, we assume the value $\overline{\alpha}(x)$ to be undefinied.

The following theorem, which is rather standard, describes the relationship between the validity of formulas in the databases over $\mathfrak{A}_0$ and $\mathfrak{A}_1$.

Theorem 3. *There exists an effective transformation of formulas* $\varphi \mapsto \varphi^*$ *of the language* $\mathtt{DBL}_{\overline{\sigma}_0}$ *into formulas of the language* $\mathtt{DBL}_{\overline{\sigma}_1}$ *such that*

(1) *If* $\varphi(x_1, \ldots, x_k,\ s_1, \ldots, s_l,\ T_1, \ldots, T_m) \in \mathtt{DBL}_{\overline{\sigma}_0}$ *and* $x_1, \ldots, x_k$ *are usual variables for elements,* $s_1, \ldots, s_l$ *record variables of table types* $\pi_1, \ldots, \pi_l$ *respectively, and* $T_1, \ldots, T_m$ *are table variables of table types* $\tau_1, \ldots, \tau_m$ *respectively, then its image is a formula*

$$\varphi^*(x_1^1, \ldots, x_1^n, \ldots, x_k^1, \ldots, x_k^n,\ s_1', \ldots, s_l',\ T_1', \ldots, T_m', \bar{p}),$$

where $x_1^1, \ldots, x_1^n, \ldots, x_k^1, \ldots, x_k^n$ *are pairwise distinct variables for elements,* $s_1', \ldots, s_l'$ *are record variables of types* $\pi_1 * n, \ldots, \pi_l * n$ *respectively, and* $T_1', \ldots, T_m'$ *are table variables of types* $\tau_1 * n, \ldots, \tau_m * n$ *respectively.*

(2) *For any* $x_1^1, \ldots, x_1^n, \ \ldots, \ x_k^1, \ldots, x_k^n \in \mathfrak{A}_1$*, for arbitrary records* $s_1', \ldots, s_l'$ *over* $\mathfrak{A}_1$ *of types* $\pi_1 * n, \ldots, \pi_l * n$*, arbitrary tables* $T_1, \ldots, T_m$ *over* $\mathfrak{A}_0$ *of types* $\tau_1, \ldots, \tau_m$ *respectively, if all the values* $\overline{\alpha}\left(x_1^1, \ldots, x_1^n\right), \ \ldots, \ \overline{\alpha}\left(x_k^1, \ldots, x_k^n\right)$, $\overline{\alpha}(s_1'), \ldots, \overline{\alpha}(s_l')$ *are defined then the condition*

$$\mathfrak{A}_0 \models \varphi\Big(\overline{\alpha}\left(x_1^1, \ldots x_1^n\right), \ldots \overline{\alpha}\left(x_k^1, \ldots x_k^n\right), \ \overline{\alpha}(s_1'), \ldots \overline{\alpha}(s_l'), T_1, \ldots, T_m\Big)$$

is equivalent to

$$\mathfrak{A}_1 \models \varphi^*\left(x_1^1, \ldots x_1^n, \ldots x_k^1, \ldots x_k^n, \ s_1', \ldots s_l', \overline{\alpha}^{-1}(T_1), \ldots \overline{\alpha}^{-1}(T_m), \bar{p}\right).$$

(3) *The transformation* $\varphi \mapsto \varphi^*$ *takes* $\exists$*–formulas into* $\exists$*–formulas.*

Proof. We assume that our formulas are in the prenex normal form, i.e., they have a form $Q_1 x_1 \ldots Q_s x_s \psi$, where $Q_i \in \{\forall, \exists\}$, $i = 1.\ldots, s$ and ψ is a disjunctive normal form. In addition, without loss of generality we may assume that our formula has no subformulas of kinds $u = v$ and $\neg(u = v)$ where u and v are record variables, since we could replace them with quantifier–free subformulas $\bigwedge_{N \in \tau}(u.N = v.N)$ and $\bigvee_{N \in \tau} \neg(u.N = v.N)$ respectively.

Define the required mapping by induction. Let

$$(s \in T)^* = s' \in T', \quad \neg(s \in T)^* = \neg(s' \in T').$$

If x and y are variables for elements then we let

$$\begin{aligned} (x = y)^* &= E^+\left(x^1, \ldots, x^n, y^1, \ldots, y^n, \bar{p}\right), \\ \neg(x = y)^* &= E^-\left(x^1, \ldots, x^n, y^1, \ldots, y^n, \bar{p}\right). \end{aligned}$$

If s and u are record variables and x is a variable for elements then we let

$$\begin{aligned} (s.N = x)^* = (x = s.N)^* &= E^+\left(s'.N^{(1)}, \ldots s'.N^{(n)}, x^1, \ldots x^n, \bar{p}\right), \\ \neg(s.N = x)^* = \neg(x = s.N)^* &= E^-\left(s'.N^{(1)}, \ldots s'.N^{(n)}, x^1, \ldots x^n, \bar{p}\right), \\ (s.N = u.M)^* &= E^+\Big(s'.N^{(1)}, \ldots s'.N^{(n)}, u'.M^{(1)}, \ldots u'.M^{(n)}, \bar{p}\Big) \end{aligned}$$

If an atomic formula has the form $P(A_1, \ldots, A_m)$ where $A_1, \ldots, A_m$ are simple expressions then we let

$$\begin{aligned} (P(A_1, \ldots, A_m))^* &= \exists x_1^1 \ldots x_1^n \ldots x_m^1 \ldots x_m^n \left(\bigwedge_{i=1}^{m} (x_i = A_i)^* \wedge \right. \\ &\qquad \left. \wedge P^+\left(x_1^1, \ldots, x_1^n, \ \ldots, x_m^1, \ldots, x_m^n, \bar{p}\right)\right), \\ \neg(P(A_1, \ldots, A_m))^* &= \exists x_1^1 \ldots x_1^n \ldots x_m^1 \ldots x_m^n \left(\bigwedge_{i=1}^{m} (x_i = A_i)^* \wedge \right. \\ &\qquad \left. \wedge P^-\left(x_1^1, \ldots, x_1^n, \ \ldots, x_m^1, \ldots, x_m^n, \bar{p}\right)\right). \end{aligned}$$

Further on, for any binary propositional connective $\beta \in \{\wedge, \vee\}$ we let

$$(A \beta B)^* = A^* \beta B^*.$$

If x is a variable for elements then we let

$$(\exists x\varphi)^* = \exists x^1 \dots x^n \left(U(x^1,\dots,x^n,\bar{p}) \wedge \varphi^*\right),$$

$$(\forall x\varphi)^* = \forall x^1 \dots x^n \left(\neg U(x^1,\dots,x^n,\bar{p}) \vee \varphi^*\right).$$

If s is a record variables of type τ then we let

$$(\exists s\varphi)^* = \exists s' \left(\bigwedge_{N\in\tau} U\left(s'.N^{(1)},\dots,s'.N^{(n)},\,\bar{p}\right) \wedge \varphi^* \right)$$

$$(\forall s\varphi)^* = \forall s' \left(\neg \bigwedge_{N\in\tau} U\left(s'.N^{(1)},\dots,s'.N^{(n)},\,\bar{p}\right) \vee \varphi^* \right).$$

A routine inductive check of the statements of the theorem is left to the reader. Theorem is complete.

Remark. The mapping $\varphi \mapsto \varphi^*$

(1) transforms Δ–formulas to formulas equivalent to Δ–formulas,
(2) for each $k > 0$, transforms Σ_k–formulas to formulas equivalent to Σ_k–formulas, Π_k–formulas to formulas equivalent to Π_k–formulas,
(3) transforms formulas without quantifiers over record variables into formulas without quantifiers over record variables,
(4) transforms formulas without record variables into formulas without record variables.

Corollary 1. *Let $\mathfrak{A}_0 \prec_\exists \mathfrak{A}_1$ and let $\langle \mathfrak{A}_1, \nu_1 \rangle$ be a constructive structure. Assume that the constructivization ν_0 of the structure $\mathfrak{A}_0$ is constructed by ν_1. Then there exists a $\overline{p} \in \mathfrak{A}_1^{<\omega}$ such that given any $\exists$–query $\varphi(v, \overline{H}, \overline{c})$ for $\mathfrak{A}_0$, one can uniformly construct a finite family of tables $\overline{H}'$ over $\mathfrak{A}_1$, parameters $\overline{c}'$, and an $\exists$–query $\varphi^*(v, \overline{H}', \overline{c}', \overline{p})$ for $\mathfrak{A}_1$ such that the realization of the query $\varphi(v, \overline{H}, \overline{c})$ equals the image by the mapping $\overline{\alpha}$ of the realization of the query $\varphi^*(v, \overline{H}', \overline{c}', \overline{p})$. Moreover, if the equivalence relation E is trivial and all $\bar{H}$ are finite then the family $\bar{H}'$ will consist of finite tables. In the rest cases, if all $\overline{H}$ are computable then one can take a computable family of tables $\overline{H}'$.*

3. Semilattice by $\exists$–interpretability

To avoid nontrivial set–theoretic considerations, we assume that elements of all structures we consider now are natural numbers and that we have fixed a countable computable predicate signature containing a countable number of m–ary predicate symbols, for each m. We also assume that all signatures of the structures we are considering here are finite subsets of this signature.

Theorem 3 immediately gives us

Corollary 2. *If $\mathfrak{A}_0 \prec_\exists \mathfrak{A}_1$ then the $\exists$–theory of the structure $\mathfrak{A}_0$ is m–reducible to the $\exists$–theory of some finite enrichment by constants of the structure $\mathfrak{A}_1$.*

The main result of the paper is the following theorem:

Theorem 4. (1) *An ordering by inclusion on finite subsets of a countable set is isomorphically embeddable into $\mathcal{L}^0_\exists$.*
(2) *Any finite partial ordering is isomorphically embeddable into $\mathcal{L}^0_\exists$.*

Proof of the theorem. P. 2 follows from p. 1, because any finite partial ordering $\langle L; \leqslant \rangle$ could be embedded into the ordering on finite subsets of a countable set by means of the mapping $x \mapsto \widehat{x} = \{y \mid y \leqslant x\}$.

Prove p. 1. Assume that we have a finite family of infinite computably enumerable sets $\overline{A} = A_1, \ldots, A_k$, $k > 0$ such that for all $i = 1, \ldots, k$ holds

$$\omega \setminus A_i \subseteq \{m^2 \mid m \in \omega\}.$$

Define the structure $\mathfrak{M}(\overline{A})$ as follows: its signature will consist of unary predicate symbols P, Q and binary predicate symbols S and R.

The basic set of this structure will be the set of natural numbers ω. Now we define the values of the above mentioned predicates.

The predicate S defines the successor function on the even numbers and thus it produces a copy of the natural numbers within our structure:

$$S(x, y) \quad \Leftrightarrow \quad \exists t\, (x = 2t \wedge y = 2t + 2).$$

Predicate P distinguishes its first element:

$$P(x) \quad \Leftrightarrow \quad x = 0.$$

Predicate Q is used to partition the copy of natural numbers formed by S on the even numbers into k–element pieces by marking the beginning of every such piece:

$$Q(x) \quad \Leftrightarrow \quad \exists t\, (x = 2kt).$$

Predicate R is defined as follows: fix some simultaneous enumerations of the sets $A_1, \ldots, A_k$ with the property that at each step only one element in only one of these enumeration is enumerated. At each step, if a new element m occurs in the enumeration of A_i then we take the minimal odd number z which is not used in the construction so far and add the pair $\langle 2(km + i), z\rangle$ to the predicate R. After this moment, the number z is considered as a number already used in the construction.

Note that the identity mapping ν on the natural numbers is a constructivization of $\mathfrak{M}(\overline{A})$. This statement might be nontrivial for predicate R only. The computability of this predicate follows from the fact that for each odd z there exists exactly one x with the property $\langle x, z\rangle \in R$, and such an x must be even.

The following lemma is obvious.

Lemma 1. *Each element $x \in \mathfrak{M}\left(\overline{A}\right)$ could be defined by an $\exists$–formula without parameters.*

Lemma 1 enables us to strengthen Corollary 2 for our class of models:

Lemma 2. *If $\mathfrak{M}(\overline{A}_0) \prec_{\exists} \mathfrak{M}(\overline{A}_1)$ then the $\exists$–theory of the structure $\mathfrak{M}(\overline{A}_0)$ is m–reducible to the $\exists$–theory of the structure $\mathfrak{M}(\overline{A}_1)$.*

We leave the proof to the reader.

Fix a computable family of computably enumerable sets A_i, $i \in \omega$ so that for each $i \in \omega$ holds $A_i \not\leqslant_T \oplus_{j \neq i} A_j$; this family exists by [1]. Without loss of generality we may assume that, in addition, all the sets A_i, $i \in \omega$ contain all non–squares.

Lemma 3. *Let $\{i_1 < \ldots < i_k\}$ be an arbitrary finite subset of ω and $k > 0$. Then the $\exists$–theory of $\mathfrak{M}(A_{i_1}, \ldots, A_{i_k})$ is Turing equivalent to $A_{i_1} \oplus \ldots \oplus A_{i_k}$.*

Proof of lemma. The redicubility of $A_{i_1} \oplus \ldots \oplus A_{i_k}$ to the $\exists$–theory of $\mathfrak{M}(\overline{A})$ is obvious.

Prove the computability of the $\exists$–theory of $\mathfrak{M}(\overline{A})$ in the set $A_{i_1} \oplus \ldots \oplus A_{i_k}$.

Any $\exists$–sentence of signature $\langle P, Q, S, R\rangle$ could be reduced to an equivalent form $\exists\bar{x}\psi$, where ψ is a perfect disjunctive normal form from all possible atomic formulas constructed from the signature symbols and variables $\bar{x}$ that occur in ψ. Here each disjunct of ψ describes a finite structure, up to isomorphism; and thus answering the question about the truth of the whole sentence $\exists\bar{x}\psi$ reduces to answering a question about the embeddability of at least one such structure into the structure $\mathfrak{M}(\overline{A})$.

Now we describe a sketch algorithm that answers this question with respect to the set $A_{i_1}\oplus\ldots\oplus A_{i_k}$. The embeddability of an arbitrary finite structure into $\mathfrak{M}(\overline{A})$ is equivalent to the satisfiability of the conjunction of the following conditions:

(1) The binary relation S should be embeddable into the successor relation on the natural numbers, which could be easily checked. If it is true then the basic set of our finite structure is a disjoint sum of the connectivity components with respect to S. In what follows, we call them simply *components*.
(2) At most one of these components could start with an element satisfying P and at most one element could satisfy P. We call the component containing this element *initial*.
(3) The only element satisfying P, if exists, must satisfy the predicate Q.
(4) Any sequence consisting of k successive elements with respect to S must contain exactly one element satisfying the predicate Q.
(5) The relations R and R^{-1} must be injective, i.e., $R(x, y_0)$ and $R(x, y_1)$ must imply $y_0 = y_1$, and $R(x_0, y)$ together with $R(x_1, y)$ must imply $x_0 = x_1$.
(6) The range of R must not contain elements neither of the domain nor of the range of S.
(7) The initial component must be embeddable into $\mathfrak{M}(\overline{A})$, which could be effectively checked using the oracle $A_{i_1}\oplus\cdots\oplus A_{i_k}$. (We don't need to worry about the rest components. Since the sets A_i, $i < k$ contain all non–squares, all these sets contain convex intervals of arbitrary lengths; by this, the rest components could be isomorphically embedded into $\mathfrak{M}(\overline{A})$ so that their images would be disjoint and have empty intersections with the image of the initial component.)

Lemma is complete.

Lemma 4. *Let* $I = \{i_1 < \ldots < i_k\} \subseteq \omega$, $J = \{j_1 < \ldots < j_s\} \subseteq \omega$, $k, s > 0$. *Then* $\mathfrak{M}(A_{i_1},\ldots,A_{i_k}) \prec_\exists \mathfrak{M}(A_{j_1},\ldots,A_{j_s})$ *is equivalent to* $I \subseteq J$.

Proof of lemma We omit the part ($\Leftarrow$), which is easy to prove.

Prove the converse implication. Assume that $\mathfrak{M}(A_{i_1},\ldots,A_{i_k}) \prec_\exists \mathfrak{M}(A_{j_1},\ldots,A_{j_s})$ but $I \nsubseteq J$. Fix an arbitrary $i_0 \in I \setminus J$. Then by the above, the $\exists$–theory of the structure $\mathfrak{M}(A_{i_1},\ldots,A_{i_k})$ m–reduces to the $\exists$–theory of $\mathfrak{M}(A_{j_1},\ldots,A_{j_s})$. Since they are T–equivalent to $A_{i_1}\oplus\ldots\oplus A_{i_k}$ and $A_{j_1}\oplus\ldots\oplus A_{j_s}$ respectively, we have $A_{i_1}\oplus\ldots\oplus A_{i_k} \leqslant_T A_{j_1}\oplus\ldots\oplus A_{j_s}$. From the properties of the sets A_i, $i<\omega$, we derive that $A_{i_0} \leqslant_T A_{j_1}\oplus\ldots\oplus A_{j_s}$. From here we have $i_0 \in J = \{j_1,\ldots,j_s\}$, which is a contradiction. Lemma is complete.

Theorem 5. *There exists a computable structure with the property that every computable structure of finite signature is $\exists$–interpretable in it.*

Proof of the theorem. The structure $\langle \omega, P_+, P_\times, P_0, P_1\rangle$, where P_+ and $P_\times$ are graphs of the addition and multiplication operations and P_0 are P_1 distinguish respectively 0 and 1, could serve as an example. It follows from the famous result by Yu.V. Matiyasevich [2] on presentations of computably enumerable predicates that any computably enumerable relation on ω in this structure is definable by an $\exists$–formula, which easily implies the theorem.

Remark. *It follows from this theorem, that $\mathcal{L}_\exists^0$ is a principal ideal of the semilattice $\mathcal{L}_\exists$.*

The authors are very grateful to the anonymous referee for useful remarks.

References

[1] A.A. Mučnik, *Solution of Post's reduction problem and of certain other problems in the theory of algorithms*, Trudy Moskov. Mat. Obšč, **7** (1958), 391–405. (Russian) MR0114751

[2] Ju.V. Matijasevič, *Enumerable sets are Diophantine*, Soviet Math. Doklady, **11**:20 (1970), 354–357; correction **11**:6 (1970), vi. Zbl 0212.33401

[3] Yu.L. Ershov, *Decidability Problems and Constructive Models*, Nauka, Moscow, 1980. (Russian) MR0598465

[4] Yu.L. Ershov, Σ*–Definabiility of algebraic systems*, Handbook of Recursive Mathematics, (Recursive Model Theory), ser. Studies in Logic and Foundations of Mathematics. Elsevier, Amsterdam, Lausanne, New York, Oxford, Shannon, Singapore, Tokyo, **1** (1998), 235–260. MR1673637

Andrey Sergeevich Morozov
Sobolev Institute of Mathematics,
pr. Koptyuga, 4,
630090, Novosibirsk, Russia
E-mail address: morozov@math.nsc.ru

Aizhan Zhanabekovna Satekbayeva
E.N.Gumilev Eurasian National University,
Pushkin str. 11,
010008, Astana, Kazakhstan
E-mail address: satekbayeva@gmail.com

Jamalbek Aliaskarovich Tussupov
E.N.Gumilev Eurasian National University,
Pushkin str. 11,
010008, Astana, Kazakhstan
E-mail address: tussupov@mail.ru