

THE LOGIC-ALGEBRAIC EQUATIONS METHOD IN SYSTEM DYNAMICS

N. V. NAGUL

ABSTRACT. An algorithm is proposed for constructing conditions that ensure the preservation of properties of many-sorted algebraic systems. The functions and relations of such systems are defined on arbitrary steps in the sense of Bourbaki. Preservation conditions are generated as solutions of logic-algebraic equations. The coupling mappings between systems have a nature of morphisms and their canonical expansions to steps. The method is illustrated by the example of analysis of a dynamical property of a discrete-event system.

INTRODUCTION

In mathematical systems theory, the productivity of interdisciplinary research is well known. In particular, in the context of this paper, we should note studies at the junction of algebra and logic (A. I. Maltsev, A. Tarski, Yu. L. Ershov, Yu. I. Zhuravlev, S. S. Goncharov, etc.), and system dynamics and algebra (S. Lee, L. V. Ovsyannikov, M. A. Arbib, and others). The present paper lies at the intersection of system dynamics, algebra, and logic, presenting a method of mathematical systems theory that allows one to algorithmically synthesize criteria for preservation of properties of systems under some special mappings — morphisms (these and other mappings between two systems under consideration will be referred to as *coupling mappings*, and the conditions *a priori* imposed on them as *coupling conditions*). One of the main applications of the preservation criteria is the reduction of some complex systems to much simpler ones.

In 1974, to algorithmically form certain criteria for preservation of properties of dynamical systems, Matrosov [10] proposed the logic based *comparison method*, which was later developed widely in his school (Anapol'sky, Vasil'ev, Kozlov). The comparison method provides preservation criteria for simple properties (the so-called properties of degree 1), but for a broad class of dynamical systems and in terms of the coupling mappings called *comparison vector functions* (CVF). In development of this method, Vasil'ev proposed a more general comparison method in the form of *logical equations* (LE, [4]) with the structure $\mathcal{X} \& \mathcal{M} \& \mathcal{F}' \rightarrow \mathcal{F}$, where \mathcal{M} , \mathcal{F}' , \mathcal{F} are known, \mathcal{F} is a property of the system \mathcal{S} , \mathcal{F}' is a property of the another system \mathcal{S}' , \mathcal{M} is the coupling condition, and \mathcal{X} is to be found. The LE method is suitable for the study of general properties \mathcal{F} (of any degree), and for various mathematical models. Moreover, in the case of the Matrosov method the coupling condition \mathcal{M} is the requirement of the majorizability of CVF at solutions of a dynamical system under consideration by the corresponding solutions of some auxiliary system (less complex *comparison system*). However, in the case of the LE method this condition may have another meaning: of a morphism, or of the requirement of topological equivalence for \mathcal{S} , \mathcal{S}' , and so on. The method of solving more general

2010 *Mathematics Subject Classification*. Primary 03C40; Secondary 08A99, 03C98.

Key words and phrases. Property preservation, many-sorted algebra, discrete-event system.

Supported by RFBR (grants nos. 12-08-90018-Bel-a and no. 11-07-00655-a) and by the Project no. 10 of the Presidium of SB RAS.

logical equations of the form $\mathcal{X} \& (\&\{C_i : i = 1, 2, \dots, n\}) \rightarrow \mathcal{F}$, where $n \geq 1$, is known as the *reduction method* [3].

It should be noted that the case of morphisms of dynamical systems is especially useful, for example, when we study stability and other dynamic properties that require a change of variables, because we must ensure that the property under consideration in old variables is equivalent to that in new variables, or at least guarantee its unidirectional preservation [6]. The case where the coupling mappings are morphisms is important in connection with the problem of stabilization of programmed motions for controlled systems via converting these systems into equivalent canonical forms [7]. Currently, morphisms are also used in the qualitative study of dynamical systems, including the controlled ones, in a number of papers; see [15]. So it makes sense to design algorithms for obtaining preservation criteria that only involve preservation conditions for operations and relations (i.e., criteria expressed in the form of traditional morphisms).

The starting point for this study was the method of solving logical equations of the form $\mathcal{X} \& \mathcal{F}' \rightarrow \mathcal{F}$, where \mathcal{F} and \mathcal{F}' are known properties of single-sorted algebraic systems (Vasil'ev [5]). Here coupling conditions are not specified *a priori*, and this fact increases algorithmizability of the method. To obtain fairly efficient preservation criteria, we increase the load on the algorithms that form conditions \mathcal{X} , by narrowing the diversity of Skolem functions corresponding to existential quantifiers in \mathcal{X} . Applying these algorithms to specific \mathcal{F} helps in the nonalgorithmical generating of preservation criteria for various classes of single-sorted algebraic systems properties. However, dynamical systems are specific in the following sense: algebraization of their models usually leads to many-sorted algebraic systems where the basic sets have the meaning of a state space, a time scale, etc. Therefore, it was necessary to extend the LE-method to the many-sorted case, what is done in this paper.

It should also be noted that the greater complexity, compared to traditional algebraic systems, of the definitions of the mathematical objects that are studied in the system dynamics requires to consider many-sorted algebraic systems (MASs), with functions and relations determined not only on the basic sets or their Cartesian products, but also on arbitrary steps in the sense of Bourbaki, and moreover, on steps with some additional extensions (in particular, this is needed to include the case of discrete-event systems). So, in view of the general nature of the MASs under study, additional expressing tools are included in the logical language, not only to represent schemes and steps (generated with these schemes) but also to represent canonical expansions of mappings (CEMs) of the basic sets to the steps. It is up to the class of these functions, CEMs, that we algorithmically narrow the variety of Skolem functions mentioned above, which are critical for the efficiency of preservation criteria. To specialize the equations in this extended (logical-algebraic) language, we call them *logical-algebraic equations* (LAE) and consider two types of such equations: $\mathcal{X} \& \mathcal{P} \rightarrow \mathcal{P}'$ and $\mathcal{X} \& \mathcal{P}' \rightarrow \mathcal{P}$ (the second equation corresponds to the study of preservation of properties in the direction opposite to coupling mappings acting from \mathbf{S} to \mathbf{S}').

In [11] we used the algorithm presented here to investigate dynamical properties of a state-machine net and Nemytsky's general dynamical systems. In the present paper (§4), we show an application of the LAE-method to the study of properties of *discrete-event systems* (DESSs), whose main distinctive feature is modeling the system evolution by considering the occurrence of some *events*. Development of the DESSs theory is driven by the rapid progress of manufacturing systems and communication networks, transportation networks and other, primarily man-made systems. At the same time, despite of the large number of studies devoted to DESSs, there is still no general apparatus for studying them, which eventually could turn into an analog of differential equations in the study

of dynamical systems with continuous variables. In §4 we explore a property of a DES as a dynamical system, by using the method of logic-algebraic equations.

§1. A GENERAL MANY-SORTED ALGEBRAIC SYSTEM OF FINITE TYPE

Let $A = \{A_\lambda \mid \lambda = 1, \dots, k\}$ be a finite family of sets A_λ . It is well known that the set theory operations of constructing Cartesian product and Boolean over the sets of A generate sets that are called *steps* over the family A ; see [1]. More precisely,

- (1) each set A_λ is a step;
- (2) if sets M_1 and M_2 are steps then each of the sets $M_1 \times M_2$ and 2^{M_1} is a step;
- (3) a set is a step if and only if this follows from (1) and (2).

To discuss properties of a larger class of dynamical systems and, in particular, discrete-event systems whose model includes phase and event trajectories as sequences of states and events, it is useful to extend the notion of the Bourbaki step to cover the third operation, namely, that constructing a set of sequences. Suppose B is a nonempty set. We denote by \mathbf{b} (lowercase bold letters) an infinite sequence of elements of B : $\mathbf{b} = (b_0, b_1, b_2, \dots)$, $b_i \in B$, $i \in \mathbb{N} = \{0, 1, 2, \dots\}$. It is obvious that \mathbf{b} is a mapping $\mathbf{b} : \mathbb{N} \rightarrow B$ of natural numbers into the set B . $B^{\mathbb{N}}$ denotes the set of all sequences of elements of B , i.e., $B^{\mathbb{N}} = \{\mathbf{b} \mid \mathbf{b} : \mathbb{N} \rightarrow B\}$.

Consider the set of k symbols $\sigma_a = \{a_\lambda \mid \lambda = 1, \dots, k\}$ ($k > 0$) and the signature $\sigma_0 = \{\sqcup \times \sqcup, \mathcal{P}, \mathcal{N}\}$. We define a *step scheme* over σ_a (or simply a *scheme* for brevity) as follows:

- (1) for all $\lambda = 1, \dots, k$, a_λ is a scheme;
- (2) if S_1, S_2 are schemes then $(S_1 \times S_2)$ is a scheme;
- (3) if S is a scheme then $\mathcal{P}(S)$ and $\mathcal{N}(S)$ are schemes;
- (4) an expression is a scheme if and only if this follows from rules (1)–(3).

$\text{St}[\sigma_a]$ will denote the *set of all step schemes* over σ_a . Given a family of sets $A = \{A_\lambda \mid \lambda = 1, \dots, k\}$, for any step scheme $S \in \text{St}[\sigma_a]$ we can naturally and unambiguously define the set $S[A]$ — a *step over the family A* (built using the scheme S). This is done by induction on the construction of S . Moreover,

- (1) the symbols $a_\lambda \in \sigma_a$ are interpreted as sets $A_\lambda \in A$ ($\lambda = 1, \dots, k$);
- (2) the binary symbol $\sqcup \times \sqcup \in \sigma_0$ is interpreted as the operation of constructing Cartesian product, i.e., for any schemes $S_1, S_2 \in \text{St}[\sigma_a]$ we put $(S_1 \times S_2)[A] = S_1[A] \times S_2[A]$;
- (3) the functional symbol $\mathcal{P} \in \sigma_0$ denotes the operation of forming the Boolean, i.e., for the scheme $\mathcal{P}(S) \in \text{St}[\sigma_a]$ we put $\mathcal{P}(S)[A] = 2^{S[A]}$;
- (4) the symbol $\mathcal{N} \in \sigma_0$ denotes the operation of forming the set of sequences, i.e., for the scheme $\mathcal{N}(S) \in \text{St}[\sigma_a]$ we put $\mathcal{N}(S)[A] = (S[A])^{\mathbb{N}}$.

Hence, the classical concepts of a scheme and a Bourbaki step are extended by introducing the extra character \mathcal{N} in σ_0 and the corresponding interpretation of schemes. By analogy with [1], the set of all steps generated as above is called the *generalized scale of steps (sets)* over A and is denoted by $\text{St}[A]$.

In the study of many-sorted algebraic systems, the language of the extended first-order predicate calculus is usually employed. Consider the signature $\sigma = \sigma_P \cup \sigma_F \cup \sigma_E$, where:

$\sigma_F = \{F_\beta^{n_\beta}, \beta = 1, \dots, k_F, n_\beta \in \{1, 2, \dots\}\}$ is the set of functional symbols, and to each symbol $F_\beta^{n_\beta}$ we assign $n_\beta + 1$ schemes $S_{1\beta}, S_{2\beta}, \dots, S_{n_\beta\beta}, S_{n_\beta+1,\beta} \in \text{St}[\sigma_a]$, where the last scheme allows us to fix the range of $F_\beta^{n_\beta}$ interpreted as an n_β -place function, and the other schemes allow us to fix the ranges of its arguments;

$\sigma_P = \{P_\gamma^{n_\gamma}, \gamma = 1, \dots, k_P, n_\gamma \in \{0, 1, 2, \dots\}\}$ is the set of relation symbols, together with the associated schemes $T_{1\gamma}, \dots, T_{n_\gamma\gamma} \in \text{St}[\sigma_a]$, $\gamma = 1, \dots, k_P$;

$\sigma_E = \{E_\delta, \delta = 1, \dots, k_E\}$ is the set of distinguished elements, together with the associated schemes $U_\delta \in \text{St}[\sigma_a]$, $\delta = 1, \dots, k_E$.

For each step scheme associated with symbols of the signature σ , we introduce (infinitely many) individual variables, and in what follows z_α will be used to denote variables bound by quantifiers, and x_μ will be free variables. We also use the logical connectives (\neg , \rightarrow , $\&$, etc.) with their common meaning, and auxiliary characters (parentheses, commas). In addition, we use symbols “=” (equality) and $\langle \sqcup, \sqcup \rangle$ (an ordered pair). Although these characters can connect elements of different pairs of steps, we use the same symbols for any such pair, i.e., we apply the “liberal option” (see [13]) to all characters implying substitution.

Now, we define the set of the well-formed expressions denoting objects, that are called *terms*; at the same time, we introduce the notion of the *sort* of a term:

- (1) each individual variable s is a term; the sort of this term is the corresponding step scheme S , this will be denoted by $|s| = S$; the range of possible values of the variable s under interpretation (i.e., under choosing basic sets) will be the step (set) $S[A]$;
- (2) each symbol E_δ of a distinguished element is a term, $|E_\delta| = U_\delta$;
- (3) if t_1, t_2 are terms of sorts $|t_1|, |t_2|$, respectively, then $\langle t_1, t_2 \rangle$ is a term of sort $|t_1| \times |t_2|$;
- (4) if t_i is term, $|t_i| = S_{i\beta}$, $i = 1, \dots, n_\beta$, and $t = \langle \dots \langle \langle t_1, t_2 \rangle, t_3 \rangle, \dots, t_{n_\beta} \rangle$, then $F_\beta^{n_\beta}(t)$ is a term of sort $S_{n_\beta+1, \beta}$, $\beta = 1, \dots, k_F$ (the inner brackets $\langle \rangle$ will be omitted);
- (5) an expression is a term only when this follows from rules (1)–(4).

We also define special expressions called *type quantifiers* (TQs): $\hat{\omega}_\alpha \stackrel{\text{def}}{=} \forall z_\alpha : Z_\alpha \stackrel{\text{def}}{=} \forall z_\alpha (Z_\alpha \rightarrow \sqcup)$ (universal type quantifier), $\check{\omega}_\alpha \stackrel{\text{def}}{=} \exists z_\alpha : Z_\alpha \stackrel{\text{def}}{=} \exists z_\alpha (Z_\alpha \& \sqcup)$ (existential type quantifier). Here Z_α is the so-called *type condition*.

To formulate statements about objects, we define well-formed expressions that are called *formulas* as follows:

- (1) if t_1, t_2 are terms and $|t_1| = |t_2|$, then the expression “ $t_1 = t_2$ ” is a formula;
- (2) if t_i is a term, $|t_i| = T_{i\gamma}$, $i = 1, \dots, n_\gamma$, $t = \langle \dots \langle \langle t_1, t_2 \rangle, t_3 \rangle, \dots, t_{n_\gamma} \rangle$, and $P_\gamma^{n_\gamma}$ is a relation symbol with the associated schemes $T_{1\gamma}, \dots, T_{n_\gamma\gamma}$, then $P_\gamma^{n_\gamma}(t)$ is a formula;
- (3) if $\mathcal{F}_1, \mathcal{F}_2$ are formulas, then the expressions $\neg(\mathcal{F}_1)$, $(\mathcal{F}_1 \& \mathcal{F}_2)$, $(\mathcal{F}_1 \vee \mathcal{F}_2)$, $(\mathcal{F}_1 \rightarrow \mathcal{F}_2)$ are formulas;
- (4) if \mathcal{F} and \mathcal{Z}_α are formulas and z_α is an individual variable, then the expressions $\hat{\omega}_\alpha \mathcal{F} = \forall z_\alpha (\mathcal{Z}_\alpha \rightarrow \mathcal{F})$, $\check{\omega}_\alpha \mathcal{F} = \exists z_\alpha (\mathcal{Z}_\alpha \& \mathcal{F})$ are formulas;
- (5) the expression is a formula if and only if this follows from rules (1)–(4).

Formulas built by rules (1) and (2) are said to be *atomic*. Let \mathcal{L} denote the language of terms and formulas built according to the rules stated. It includes only elements of σ as functional symbols, predicate symbols, and symbols of distinguished elements, and the additional symbols “=” and $\langle \sqcup, \sqcup \rangle$.

By a *general many-sorted algebraic system of finite type* (GMAS) and of signature σ , we shall mean an object $\mathfrak{A} = \langle A, \Omega_F, \Omega_P, \Omega_E \rangle$ that consists of a family of sets $A = \{A_\lambda | \lambda = 1, \dots, k\}$ and the following sets, corresponding to σ_F, σ_P , and σ_E : a set of functions $\Omega_F = \{F_\beta^{n_\beta} | F_\beta^{n_\beta} : S_{1\beta}[A] \times S_{2\beta}[A] \times \dots \times S_{n_\beta\beta}[A] \rightarrow S_{n_\beta+1, \beta}[A], \beta = 1, \dots, k_F\}$, a set of relations $\Omega_P = \{P_\gamma^{n_\gamma} | P_\gamma^{n_\gamma} \subseteq T_{1\gamma}[A] \times \dots \times T_{n_\gamma\gamma}[A], \gamma = 1, \dots, k_P\}$, and a set of distinguished elements $\Omega_E = \{E_\delta | E_\delta \in U_\delta[A], \delta = 1, \dots, k_E\}$. Note that, in contrast to [5], [9], [14], elements of the set $\Omega_F \cup \Omega_P \cup \Omega_E$ may be defined on extended steps over A . The generalization of the concepts of a Bourbaki scheme and a step allows

us to easily represent various models of dynamical systems and, in particular, DESs, in the form of a GMAS.

Consider two mappings $\psi_1 : B_1 \rightarrow B_1'$ and $\psi_2 : B_2 \rightarrow B_2'$. The mapping $\psi_1 * \psi_2 : B_1 \times B_2 \rightarrow B_1' \times B_2'$ defined by

$$(\psi_1 * \psi_2)(b_1, b_2) = \langle \psi_1(b_1), \psi_2(b_2) \rangle \quad (\forall b_1 \in B_1, \forall b_2 \in B_2),$$

is called the *canonical extension of the given mappings to the Cartesian product*; see [1].

The mapping $\widehat{\psi}_1 : 2^{B_1} \rightarrow 2^{B_1'}$ defined by

$$\widehat{\psi}_1(B) = \{b' \in B_1' \mid \exists b \in B : \psi_1(b) = b'\} \quad (\forall B \subseteq B_1),$$

is called the *canonical extension of ψ_1 to the Boolean*; see [1].

The mapping $\psi_1|_N : B_1^{\mathbb{N}} \rightarrow (B_1')^{\mathbb{N}}$ defined by

$$\forall \mathbf{b} \in B_1^{\mathbb{N}} \quad \psi_1|_N(\mathbf{b}) = \mathbf{b}', \text{ where } b'_i = \psi_1(b_i) \quad \forall i \in \mathbb{N},$$

is called the *canonical extension of ψ_1 to the set of sequences*.

Let $A' = \{A'_\lambda \mid \lambda = 1, \dots, k\}$. For a step $S[A]$, we denote by $S[A']$ the step over A' formed from the sets A'_λ by using the same scheme S that was employed when we formed $S[A]$ from the sets A . For any family of mappings

$$(1.1) \quad \varphi = \{\varphi_\lambda \mid \varphi_\lambda : A_\lambda \rightarrow A'_\lambda, \lambda = 1, \dots, k\}$$

and each step $S[A] \in \text{St}[A]$ we inductively define the *canonical extension of mappings* (CEM) (1.1) to $S[A]$ (i.e., $\langle \varphi \rangle^{S[A]} : S[A] \rightarrow S[A']$) as follows:

- (1) if $S[A] = A_\lambda$, then $\langle \varphi \rangle^{S[A]} = \varphi_\lambda$;
- (2) if $S[A] = S_1[A] \times S_2[A]$, then $\langle \varphi \rangle^{S[A]} = \langle \varphi \rangle^{S_1[A]} * \langle \varphi \rangle^{S_2[A]}$;
- (3) if $S[A] = 2^{S_1[A]}$, then $\langle \varphi \rangle^{S[A]} = \widehat{\langle \varphi \rangle^{S_1[A]}}$;
- (4) if $S[A] = S_1[A]^N$, then $\langle \varphi \rangle^{S[A]} = \langle \varphi \rangle^{S_1[A]}|_N$.

§2. SYNTHESIS OF CRITERIA FOR THE PRESERVATION OF PROPERTIES OF GMAS

Let $\mathcal{F}(\bar{x}) \stackrel{\text{def}}{=} \mathcal{F}(x_1, \dots, x_p)$ be an arbitrary formula of signature σ in the language \mathcal{L} , where x_μ is a free variable, $\mu = 1, \dots, p$, $p \geq 0$. Without loss of generality, we may assume that \mathcal{F} is formed of *literals* (*concluding statements*, or *c-formulas*) \mathcal{F}^ν , that is, atomic formulas \mathcal{F}_+^ν or their negations \mathcal{F}_-^ν , with the help of the connectives $\&$, \vee and the type quantifiers $\widehat{\omega}_\alpha, \check{\omega}_\alpha$, $\alpha = 1, \dots, n$, $\nu = 1, \dots, M$. Unlike positive formulas used, for example, in the theory of general algebraic systems [8], such formulas will be referred to as *generalized positive formulas*. This notion is similar to the notion of the “negation normal form” (see, e.g., [13]) except that generalized positive formulas are built with type quantifiers. The number M is called the *degree of a property*.

We write $\mathcal{J} \sqsubseteq \mathcal{F}$ to denote subformulas \mathcal{J} that arise in the process of building \mathcal{F} from its concluding statements. The formulas \mathcal{F}^ν and formulas that arise in the process of building \mathcal{F} by adding connectives and type quantifiers are called the *main subformulas* of the formula \mathcal{F} (for example, a type condition \mathcal{Z}_α is not a main subformula).

Since only TQs will be used in what follows, they are referred to simply as quantifiers for short. Assume that all quantifying variables z_α are pairwise different. Without loss of generality, the free variables x_μ of the formula \mathcal{F} are also assumed to be pairwise distinct and different from the z_α . Moreover, it is assumed that \mathcal{F} does not include unessential quantifiers, that is, those whose scope does not involve the variable of the quantifier.

For the given interpretation (choosing a GMAS \mathfrak{A} with the family of basic sets A) each formula of the language \mathcal{L} without free variables (a *closed* formula) represents a true or false sentence. The formula $\mathcal{F}(\bar{x})$ with free variables x_1, \dots, x_p expresses a *formula predicate* $\mathcal{F}_{\mathfrak{A}}$, i.e., some p -place relation over A that may be satisfied at some

collections $c_1 \in S_1[A], \dots, c_p \in S_p[A]$ (of values of the variables x_1, \dots, x_p) and not satisfied at the others. Respectively, in the given interpretation the formula $\mathcal{F}(\bar{x})$ is said to be *satisfied (not satisfied)* at the collection c_1, \dots, c_p . Unlike the traditional definition [8], $\mathcal{F}(\bar{x})$ is called a *property* of a GMAS even if $p \neq 0$. The formula $\mathcal{F}(\bar{x})$ is said to be *true (false)* in the interpretation in question if it is satisfied (respectively, not satisfied) at each collection $c_1 \in S_1[A], \dots, c_p \in S_p[A]$, and it is a *valid formula* (respectively, *contradiction*) if it is true (respectively, false) in any interpretation.

Let $\mathfrak{A}, \mathfrak{A}'$ be two GMASs of the same type and of signature σ with basic sets families $A = \{A_\lambda | \lambda = 1, \dots, k\}$ and $A' = \{A'_\lambda | \lambda = 1, \dots, k\}$, respectively. "Same type" means that the powers of the sets A and A' , Ω_F and Ω'_F , and so on coincide pairwise, and the step $S'_{1\beta}[A'](S'_{2\beta}[A'], T'_\gamma[A'], U'_\delta[A']$, respectively) is formed from the sets A'_λ by using the same scheme as was employed when $S_{1\beta}[A]$ ($S_{2\beta}[A], T_\gamma[A], U_\delta[A]$, respectively) was formed from the sets A_λ .

Consider the p -place formula predicate $\mathcal{F}_{\mathfrak{A}}$ defined by the formula $\mathcal{F}(\bar{x})$ on the system \mathfrak{A} . Our aim is to find a way of synthesizing conditions ensuring that $\mathcal{F}_{\mathfrak{A}}$ is preserved under the mappings (1.1). Note that in the general theory of algebraic systems [8], the term "stability" is often used instead of the term "preservation", and the term "transferring" is employed in the Bourbaki theory of structures [1].

In other words, it is necessary to construct an algorithm for finding nontrivial conditions \mathcal{X} that guarantee that, whenever $\mathcal{F}(\bar{x})$ is satisfied with some fixed values c_1, \dots, c_p of the variables x_1, \dots, x_p of the system \mathfrak{A} , it is also satisfied with the corresponding values $\langle \varphi \rangle^{S_1[A]}(c_1), \dots, \langle \varphi \rangle^{S_p[A]}(c_p)$ of the CEMs (1.1) on \mathfrak{A}' . Here S_μ is the sort of the variable $x_\mu, \mu = 1, \dots, p$.

It is convenient to seek \mathcal{X} also as a first-order language formula. To do so, to the signature σ we add a set $\sigma' = \sigma'_a \cup \sigma'_P \cup \sigma'_F \cup \sigma'_E$ of symbols that duplicate the symbols of the signature $\sigma, \sigma' \cap \sigma = \emptyset$. The formula $\mathcal{F}' = \mathcal{F}(P'/P, F'/F, E'/E, x'_\mu/x_\mu, z'_\alpha/z_\alpha)$ will be of signature σ' and the expression X'/X means the substitution of the symbol X' for all entries of the symbol X , i.e., each symbol $P \in \sigma_P, F \in \sigma_F, E \in \sigma_E, \dots$, is replaced with a new corresponding symbol $P' \in \sigma'_P, F' \in \sigma'_F, E' \in \sigma'_E, \dots, \mu = 1, \dots, p, \alpha = 1, \dots, n$. It is assumed that the new variables x'_μ, z'_α are pairwise different and do not coincide with the variables of $\mathcal{F}(\bar{x})$.

Let f_λ be new functional symbols, which for all $\lambda = 1, \dots, k$ are always interpreted on a set of single-valued functions $\varphi_\lambda : A_\lambda \rightarrow A'_\lambda$ total on the basic set A_λ .

Denote $\sigma^* = \sigma \cup \sigma' \cup \{f_\lambda\}$. Let the language \mathcal{L}^* be built on the basis of σ^* in the same way as \mathcal{L} was built on the basis of σ but with the additional expansions mentioned above. The list of them is augmented with three functional symbols $\sqcup \bullet \sqcup, B(\sqcup), \sqcup \parallel^N$, which will be interpreted as the operations $\sqcup * \sqcup, \hat{\sqcup}, \sqcup \parallel^N$ of constructing CEMs of (1.1) to the Cartesian product, the Boolean, and sequences sets, respectively. The term f^S is formed from the symbols f_λ by using the scheme S of CEM building and these three operations. It is defined in a natural way.

So, the problem stated above can be formulated as that of finding a nontrivial solution of the logic-algebraic equation (LAE)

$$(2.1) \quad \mathcal{X} \ \& \ \mathcal{F}(\bar{x}) \rightarrow \mathcal{F}'(\overline{f(x)})$$

where $\mathcal{F}'(\overline{f(x)}) = \mathcal{F}'(f^{|x_1|}(x_1)/x'_1, \dots, f^{|x_p|}(x_p)/x'_p)$, and conditions \mathcal{X} should be constructed in the signature σ^* .

If $\sigma_F \cup \sigma_E \neq \emptyset$, we transform $\mathcal{F}(\bar{x})$ into some new formula in such a way that its concluding statements do not contain entries of symbols of $\sigma_F \cup \sigma_E$. The transformation is defined in the following way: first, in $\mathcal{F}^\nu, \nu = 1, \dots, M$, we look for a term of the form either a^o) E_1 , or b^o) $F_1(s_{11}, s_{12}, \dots, s_{1n_1})$, where $F_1 \in \sigma_F$, the s_{1i} are variables x_μ or

z_α (i.e., the term with the only functional symbol F_1), $i = 1, \dots, n_1$, $n_1 > 0$. Denote the term found by t_1 (say). Out of the variables z_α in t_1 , if any, we choose a variable $s_{1\alpha_1}$ that is a quantifying variable of TQ $\tilde{\omega}_{1\alpha_1}$ with a minimal scope \mathcal{D}_1 , and then set

$$\Phi_1 \stackrel{\text{def}}{=} \begin{cases} \mathcal{F}(\widehat{\omega}_{1\alpha_1} \forall y_1 (y_1 = t_1 \rightarrow \mathcal{D}_1(y_1/t_1)) / \widehat{\omega}_{1\alpha_1} \mathcal{D}_1) & \text{if } \tilde{\omega}_{1\alpha_1} = \widehat{\omega}_{1\alpha_1}; \\ \mathcal{F}(\check{\omega}_{1\alpha_1} \exists y_1 (y_1 = t_1 \& \mathcal{D}_1(y_1/t_1)) / \check{\omega}_{1\alpha_1} \mathcal{D}_1) & \text{if } \tilde{\omega}_{1\alpha_1} = \check{\omega}_{1\alpha_1}. \end{cases}$$

Here $\mathcal{D}_1(y_1/t_1)$ is the result of replacing in \mathcal{D}_1 of all entries of the term t_1 by a variable y_1 not involved in \mathcal{F} before.

If there are no variables z_α among the s_{1i} (if t_1 is E_1 , this is so indeed), then

$$\Phi_1 \stackrel{\text{def}}{=} \exists y_1 (y_1 = t_1 \& \mathcal{F}(y_1/t_1)).$$

In the process of consecutive transformation of the formula \mathcal{F} , the literals \mathcal{F}^ν are transformed into expressions that are called the *traces* of these literals (in contrast to the added literals, such as $y_1 = t_1$).

Consider Φ_1 . If there are still symbols $F \in \sigma_F$ in the traces of the literals \mathcal{F}^ν , then we choose a term t_2 either of the form a^o) E_2 , or of the form b^o) $F_2(s_{21}, s_{22}, \dots, s_{2n_2})$, where $F_2 \in \sigma_F$, the s_{2i} are variables x_μ , z_α or y_1 , $i = 1, \dots, n_2$, $n_2 > 0$. Out of the variables z_α in t_2 , if any, again we choose a variable $s_{2\alpha_2}$ that is a quantifying variable of TQ $\tilde{\omega}_{2\alpha_2}$ with a minimal scope \mathcal{D}_2 , and set

$$\Phi_2 \stackrel{\text{def}}{=} \begin{cases} \Phi_1(\widehat{\omega}_{2\alpha_2} \forall y_2 (y_2 = t_2 \rightarrow \mathcal{D}_2(y_2/t_2)) / \widehat{\omega}_{2\alpha_2} \mathcal{D}_2) & \text{if } \tilde{\omega}_{2\alpha_2} = \widehat{\omega}_{2\alpha_2}, \\ \Phi_1(\check{\omega}_{2\alpha_2} \exists y_2 (y_2 = t_2 \& \mathcal{D}_2(y_2/t_2)) / \check{\omega}_{2\alpha_2} \mathcal{D}_2) & \text{if } \tilde{\omega}_{2\alpha_2} = \check{\omega}_{2\alpha_2}, \end{cases}$$

where y_2 is a new variable.

Continuing this process if necessary, we obtain a formula $\Psi(\bar{x}) = \Psi(x_1, \dots, x_p)$ without symbols $E \in \sigma_E$ and $F \in \sigma_F$ in the traces of the literals \mathcal{F}^ν , $\nu = 1, \dots, M$. If all \mathcal{F}^ν do not include elements of $\sigma_F \cup \sigma_E$ from the outset, then $\Psi(\bar{x}) = \mathcal{F}(\bar{x})$.

Let y_{ε^*} be the last of the added variables y_ε , i.e., $\varepsilon = 1, \dots, \varepsilon^*$. Denote by \hat{y}_ε (respectively, \check{y}_ε) the quantifiers $\forall y_\varepsilon (y_\varepsilon = t_\varepsilon \rightarrow \sqcup)$ (respectively, $\exists y_\varepsilon (y_\varepsilon = t_\varepsilon \& \sqcup)$). For any formula Θ , denote by $|\Theta, e|$ a word composed of all quantifiers controlling an element e in the formula Θ , and denote by $\|\Theta, e\|$ the same word where all existential quantifiers are replaced by the corresponding universal quantifiers. $H\Theta$ denotes the result of deleting all unessential (in the above sense) quantifiers from Θ .

Also, we introduce the notation

$$\begin{aligned} \mathcal{C} &= \& \{ H \|\Psi, \check{y}_\varepsilon\| \exists y_\varepsilon (y_\varepsilon = t_\varepsilon) : \varepsilon = 1, \dots, \varepsilon^* \& \check{y}_\varepsilon \in \Psi \}, \\ \mathcal{C}' &= \& \{ H \|\Psi', \hat{y}_\varepsilon\| \exists y'_\varepsilon (y'_\varepsilon = t'_\varepsilon) : \varepsilon = 1, \dots, \varepsilon^* \& \hat{y}_\varepsilon \in \Psi' \}, \end{aligned}$$

where

$$\Psi'(\bar{x}') = \Psi'(x'_1, \dots, x'_p) = \Psi(P'/P, F'/F, E'/E, z'_\alpha/z_\alpha, x'_\mu/x_\mu, y'_\varepsilon/y_\varepsilon),$$

$t'_\varepsilon = t_\varepsilon(U'_\varepsilon/U_\varepsilon, z'_\alpha/z_\alpha, y'_\tau/y_\tau, f^{|x_\mu|}(x_\mu)/x_\mu)$ (replacement of all $P \in \sigma_P$, $F \in \sigma_F$, $E \in \sigma_E$, $\alpha = 1, \dots, n$, $\mu = 1, \dots, p$, $\tau = 1, \dots, \varepsilon^*$, $\tau \neq \varepsilon$), $U_\varepsilon = E_\varepsilon$ or $U_\varepsilon = F_\varepsilon$ in accordance with cases a^o and b^o, $\varepsilon = 1, \dots, \varepsilon^*$. We agree that if there is no ε with $\check{y}_\varepsilon \in \Psi$ (respectively, $\hat{y}_\varepsilon \in \Psi'$), then we consider \mathcal{C} (respectively, \mathcal{C}') as being identically true.

Let

$$\begin{aligned} \Psi'(\overline{f(\bar{x})}) &= \Psi'(f^{|x_1|}(x_1)/x'_1, \dots, f^{|x_p|}(x_p)/x'_p), \\ \mathcal{F}'(\overline{f(\bar{x})}) &= \mathcal{F}'(f^{|x_1|}(x_1)/x'_1, \dots, f^{|x_p|}(x_p)/x'_p). \end{aligned}$$

Lemma 1. $\mathcal{C} \& \mathcal{C}' \rightarrow ((\Psi(\bar{x}) \rightarrow \Psi'(\overline{f(\bar{x})})) \rightarrow (\mathcal{F}(\bar{x}) \rightarrow \mathcal{F}'(\overline{f(\bar{x})})))$.

Proof. It is easy to show that

$$\begin{aligned}\mathcal{C} &\rightarrow (\mathcal{F}(\bar{x}) \rightarrow \Psi(\bar{x})), \\ \mathcal{C}' &\rightarrow (\Psi'(\overline{f(x)}) \rightarrow \mathcal{F}'(\overline{f(x)})),\end{aligned}$$

and we are done. \square

The following notation will be used below. Ψ^ν denotes the trace of the literal \mathcal{F}^ν in Ψ , $\nu = 1, \dots, M$, and \tilde{v} (respectively, \hat{v}) denotes TQ \tilde{w}_α (respectively, \hat{w}_α) if v is a variable z_α . If v is a variable y_ε , then \tilde{v} (respectively, \hat{v}) denotes TQ \tilde{y}_ε (respectively, \hat{y}_ε). Also, the symbols $\Psi'_+, \Psi'_-,$ and \check{v}', \hat{v}' are employed to denote quantifiers over variables v' of the formula $\Psi'(\bar{x}')$.

Let $V_j \stackrel{\text{def}}{=} (f^{|v_j|}(v_j) = v'_j)$, $j = 1, \dots, n^*$ (obviously, $n^* = n + \varepsilon^*$, i.e., the sum of the number of the variables bounded with quantifiers in the initial formula, and the number of new variables),

$$(\check{v}_j : V_j) \stackrel{\text{def}}{=} \begin{cases} \exists v_j (Z_\alpha \ \& \ V_j \ \& \ \sqcup) & \text{if } v_j \text{ is a variable } z_\alpha; \\ \exists y_\varepsilon (y_\varepsilon = t_\varepsilon \ \& \ V_j \ \& \ \sqcup) & \text{if } v_j \text{ is a variable } y_\varepsilon. \end{cases}$$

The expression $(\check{v}'_j : V_j)$ is understood in a similar way. Next Ξ'_+ (respectively, Ξ'_-) denotes the implication $\Psi'_+ \rightarrow (\Psi')'_+$ (respectively, $(\Psi')'_+ \rightarrow \Psi'_+$), where $(\Psi')'_+$ is the literal in Ψ' corresponding to the literal Ψ'_+ in Ψ .

We modify the formula Ψ as follows:

- (1) consider the maximal length chains of neighbor quantifiers of the same type, i.e., maximal chains $\hat{v}_1 \dots \hat{v}_m$ or $\check{v}_1 \dots \check{v}_m$ of neighbor quantifiers not separated in the structure of Ψ by quantifiers \exists and \forall , respectively, and by logical connectives $\&$, \vee ; replace all these chains with

$$\begin{aligned}\hat{v}'_1 \dots \hat{v}'_m (\check{v}_1 : V_1) \dots (\check{v}_m : V_m), \\ \hat{v}_1 \dots \hat{v}_m (\check{v}'_1 : V_1) \dots (\check{v}'_m : V_m),\end{aligned}$$

respectively;

- (2) replace all logical connectives \vee with $\&$;
- (3) replace all literals Ψ^ν_+ , $\nu = 1, \dots, M$, with formulas Ξ^ν_+ , and all literals Ψ^ν_- with formulas Ξ^ν_- ;
- (4) after the transformation described in (1)–(3), we replace all x'_μ with $f^{|x_\mu|}(x_\mu)$ and denote the result by \mathcal{R}_1 .

Lemma 2. $\mathcal{C} \ \& \ \mathcal{C}' \rightarrow (\mathcal{R}_1 \rightarrow (\mathcal{F}(\bar{x}) \rightarrow \mathcal{F}'(\overline{f(x)})))$.

Proof. First, we show the validity of the implication

$$(2.2) \quad \mathcal{R}_1 \rightarrow (\Psi(\bar{x}) \rightarrow \Psi'(\overline{f(x)})),$$

and then use Lemma 1.

The validity of the implication will be proved by induction on the length of the formula \mathcal{R}_1 . Note that the algorithm proposed above builds the formulas Ψ' and \mathcal{R}_1 over the formula Ψ in such a way that the structures of the formulas Ψ and Ψ' are identical, and the only TQs entering \mathcal{R}_1 are those that enter Ψ and Ψ' . Also, the quantifiers in \mathcal{R}_1 follow in the same order as the corresponding quantifiers in Ψ and Ψ' , and the c-formulas Ξ^ν in \mathcal{R}_1 correspond to the c-formulas Ψ^ν and $(\Psi')^\nu$ (by obvious semantic considerations, we assume that Ξ^ν are c-formulas of \mathcal{R}_1). Therefore, some main subformulas Ψ^* , $(\Psi')^*$ of Ψ , Ψ' correspond unambiguously to each main subformula $\mathcal{R}_1^* \sqsubseteq \mathcal{R}_1$.

Denoting by K the number of TQs and logical connectives in \mathcal{R}_1 (the induction parameter), suppose that there exists $K^* \geq 0$ such that for all $K \leq K^*$ the implication

(2.2) holds true:

$$(2.3) \quad \mathcal{R}_1^* \rightarrow (\Psi(\bar{x})^* \rightarrow (\Psi'(f(\bar{x})))^*).$$

Now we want to show that the role of K^* can be played, for example, by 0. In fact, the case where $K^* = 0$ means that either $\mathcal{R}_1 = \Xi_+^\nu$ or $\mathcal{R}_1 = \Xi_-^\nu$, for some $\nu = 1, \dots, M$. Then the formula \mathcal{R}_1 is put in correspondence to $\Psi_+^\nu, (\Psi')_+^\nu$ in the first case and to $\Psi_-^\nu, (\Psi')_-^\nu$ in the second case, and also $\Psi_-^\nu = \neg\Psi_+^\nu, (\Psi')_-^\nu = \neg(\Psi')_+^\nu$. The validity of

$$\begin{aligned} (\Psi_+^\nu \rightarrow (\Psi')_+^\nu) &\rightarrow (\Psi_+^\nu \rightarrow (\Psi')_+^\nu), \\ ((\Psi')_+^\nu \rightarrow \Psi_+^\nu) &\rightarrow (\neg\Psi_+^\nu \rightarrow \neg(\Psi')_+^\nu) \end{aligned}$$

is obvious.

The inductive step from \mathcal{R}_1^* to a overformula $\tilde{\mathcal{R}}_1$ with $K = \tilde{K} = K^* + 1$ consists of consideration of two cases. In virtue of isomorphic embeddings, some subformulas $\tilde{\Psi}$ and $\tilde{\Psi}'$ correspond to the formula $\tilde{\mathcal{R}}_1$ in $\Psi(\bar{x})$ and $\Psi'(\bar{x})$.

Case 1. The formula $\tilde{\mathcal{R}}_1$ is built of \mathcal{R}_1^* by adding the logical connective $\&$ (by definition, there are no other logical connectives in \mathcal{R}_1) connecting the subformulas \mathcal{R}_1^* and \mathcal{R}_1^{**} , and for both of them (2.3) holds true, i.e.,

$$\begin{aligned} (2.4) \quad &\mathcal{R}_1^* \rightarrow (\Psi(\bar{x})^* \rightarrow (\Psi'(f(\bar{x})))^*), \\ (2.5) \quad &\mathcal{R}_1^{**} \rightarrow (\Psi(\bar{x})^{**} \rightarrow (\Psi'(f(\bar{x})))^{**}). \end{aligned}$$

It is required to prove that

$$\mathcal{R}_1^* \& \mathcal{R}_1^{**} \rightarrow (\Psi(\bar{x})^* L\Psi(\bar{x})^{**} \rightarrow (\Psi'(f(\bar{x}))^* L(\Psi'(f(\bar{x})))^{**})),$$

where L is either \vee or $\&$. This easily follows from (2.4) and (2.5).

Case 2. $\tilde{\mathcal{R}}_1$ is built from \mathcal{R}_1^* by adding a vertex v_j corresponding to either a TQ variable y_ε or a variable z_α . Four subcases are possible in this case.

Case 2.1. Let $\tilde{\mathcal{R}}_1 = \hat{v}_j \mathcal{R}_1^*$. Then, by the construction of \mathcal{R}_1 , a subformula $\check{v}_j \Psi(\bar{x})^* \sqsubseteq \Psi(\bar{x})$ corresponds to the subformula $\tilde{\mathcal{R}}_1$. We must prove that

$$\hat{v}_j \mathcal{R}_1^* \rightarrow (\check{v}_j \Psi(\bar{x})^* \rightarrow (\Psi'(f(\bar{x})))^*),$$

given the inductive hypothesis (2.3). Note that the formula $(\Psi'(f(\bar{x})))^*$ does not depend on the variable v_j . The proof is obvious.

Case 2.2. $\tilde{\mathcal{R}}_1 = \hat{v}'_j \mathcal{R}_1^*$. Then the subformula $\hat{v}'_j (\Psi'(f(\bar{x})))^* \sqsubseteq \Psi'(f(\bar{x}))$ corresponds to $\tilde{\mathcal{R}}_1$. The validity of

$$\hat{v}'_j \mathcal{R}_1^* \rightarrow (\Psi(\bar{x})^* \rightarrow \hat{v}'_j (\Psi'(f(\bar{x})))^*)$$

is obvious, because $\Psi(\bar{x})^*$ does not depend on the variable v_j .

Case 2.3. If $\tilde{\mathcal{R}}_1 = (\check{v}_j : V_j) \mathcal{R}_1^*$, then a subformula $\hat{v}_j \Psi(\bar{x})^* \sqsubseteq \Psi(\bar{x})$ corresponds to $\tilde{\mathcal{R}}_1$. We must prove that

$$(\check{v}_j : V_j) \mathcal{R}_1^* \rightarrow (\hat{v}_j \Psi(\bar{x})^* \rightarrow (\Psi'(f(\bar{x})))^*)$$

under condition (2.3). Let a be a value of the variable v_j such that conditions V_j and \mathcal{R}_1^* are satisfied. Then, by the inductive hypothesis, $\Psi(\bar{x})^* \rightarrow (\Psi'(f(\bar{x})))^*$ is true for a . For TQ-formulas, the validity of the implication $\check{w}_\alpha (\mathcal{F}_1 \rightarrow \mathcal{F}_2) \rightarrow (\hat{w}_\alpha \mathcal{F}_1 \rightarrow \mathcal{F}_2)$ is known, where \mathcal{F}_2 does not depend on z_α . This readily implies the claim.

Case 2.4. $\tilde{\mathcal{R}}_1 = (\check{v}'_j : V_j)\mathcal{R}_1^*$. Then the subformula $\check{v}'_j(\Psi'(f(x)))^* \sqsubseteq \Psi'(f(x))$ corresponds to $\tilde{\mathcal{R}}_1$ by construction. We must prove that

$$(\check{v}'_j : V_j)\mathcal{R}_1^* \rightarrow (\Psi(\bar{x})^* \rightarrow \check{v}'_j(\Psi'(f(x)))^*)$$

under condition (2.3). In this case we use the well-known implication $\check{w}_\alpha(\mathcal{F}_1 \rightarrow \mathcal{F}_2) \rightarrow (\mathcal{F}_1 \rightarrow \check{w}_\alpha\mathcal{F}_2)$ with \mathcal{F}_1 not depending on z_α .

Now, from (2.2) and Lemma 1 we immediately obtain $\mathcal{C} \ \& \ \mathcal{C}' \rightarrow (\mathcal{R}_1 \rightarrow (\mathcal{F}(\bar{x}) \rightarrow \mathcal{F}'(f(x))))$. This finishes the proof of Lemma 2. □

The above method of constructing conditions \mathcal{R}_1 for a given formula $\mathcal{F}(\bar{x})$ determines an algorithm of synthesis for solutions

$$\mathcal{X} \stackrel{\text{def}}{=} \mathcal{C} \ \& \ \mathcal{C}' \ \& \ \mathcal{R}_1$$

of the LAE (2.1) that guarantee the preservation of the truth values of formula predicates under mappings of many-sorted algebraic systems to one another. Lemma 2 proves that this algorithm is consistent.

It is easy to realize that, in a similar way, we can obtain conditions sufficient for preservation of GMAS properties in the “reverse” direction, i.e., when we pass from images to preimages. Indeed, suppose, given mappings (1.1), we want to find nontrivial conditions that guarantee that, whenever the formula $\mathcal{F}(\bar{x}) \stackrel{\text{def}}{=} \mathcal{F}(x_1, \dots, x_p)$ on the system \mathfrak{A} is satisfied at the values of CEMs $\langle \varphi \rangle^{S_1[A]}(c_1), \dots, \langle \varphi \rangle^{S_p[A]}(c_p)$ corresponding to some values c_1, \dots, c_p of the variables x_1, \dots, x_p on the system \mathfrak{A} (where $S_\mu[A]$ is the sort of the variable $x_\mu, \mu = 1, \dots, p$), the formula $\mathcal{F}(\bar{x})$ is satisfied on \mathfrak{A} .

We will seek preservation conditions as the solution of the LAE

$$(2.6) \quad \bar{\mathcal{X}} \ \& \ \mathcal{F}'(f^{|x_1|}(x_1), \dots, f^{|x_p|}(x_p)) \rightarrow \mathcal{F}(x_1, \dots, x_p).$$

The algorithm of constructing the formula $\bar{\mathcal{X}}$ preserves the main features of the algorithm of constructing \mathcal{X} . However, on the stage of excluding functional symbols and symbols of distinguished elements from the c -formulas of \mathcal{F} , we choose the following auxiliary formulas:

$$\begin{aligned} \bar{\mathcal{C}} &= \&\{H \parallel \Psi, \hat{y}_\varepsilon \parallel \exists y_\varepsilon (y_\varepsilon = t_\varepsilon) : \varepsilon = 1, \dots, \varepsilon^* \ \& \ \hat{y}_\varepsilon \in \Psi\}, \\ \bar{\mathcal{C}}' &= \&\{H \parallel \Psi', \check{y}'_\varepsilon \parallel \exists y'_\varepsilon (y'_\varepsilon = t'_\varepsilon) : \varepsilon = 1, \dots, \varepsilon^* \ \& \ \check{y}'_\varepsilon \in \Psi'\}, \end{aligned}$$

where the entire notation has the same meaning.

Lemma 3. $\bar{\mathcal{C}} \ \& \ \bar{\mathcal{C}}' \rightarrow ((\Psi'(f(x)) \rightarrow \Psi(\bar{x})) \rightarrow (\mathcal{F}'(f(x)) \rightarrow \mathcal{F}(\bar{x})))$.

Proof. It is easy to show that

$$\begin{aligned} \bar{\mathcal{C}} &\rightarrow (\Psi(\bar{x}) \rightarrow \mathcal{F}(\bar{x})), \\ \bar{\mathcal{C}}' &\rightarrow (\mathcal{F}'(f(x)) \rightarrow \Psi'(f(x))), \end{aligned}$$

and the claim follows. □

The formula Ψ will be transformed in the following way:

- (1) the maximal length chains of the same type neighbor quantifiers $\hat{v}_1 \dots \hat{v}_m$ and $\check{v}_1 \dots \check{v}_m$ are replaced with $\hat{v}_1 \dots \hat{v}_m(\check{v}'_1 : V_1) \dots (\check{v}'_m : V_m)$ and $\hat{v}'_1 \dots \hat{v}'_m(\check{v}_1 : V_1) \dots (\check{v}_m : V_m)$, respectively;
- (2) all connectives \vee are replaced with $\&$;
- (3) all literals $\Psi'_\nu, \nu = 1, \dots, M$, are replaced with the formulas Ξ'_ν , and all literals Ψ''_ν are replaced with the formulas Ξ''_ν ;
- (4) after the transformation described in (1)–(3), all variables x'_μ are replaced with $f^{|x_\mu|}(x_\mu)$, and the result is denoted by $\bar{\mathcal{R}}_1$.

Lemma 4. $\bar{C} \& \bar{C}' \rightarrow (\bar{\mathcal{R}}_1 \rightarrow (\mathcal{F}'(\overline{f(x)}) \rightarrow \mathcal{F}(\bar{x})))$.

The proof of this lemma is the same as that of Lemma 2 and is omitted for this reason. Lemma 4 justifies the admissibility of the solution

$$\bar{\mathcal{X}} \stackrel{\text{def}}{=} \bar{C} \& \bar{C}' \& \bar{\mathcal{R}}_1$$

of the LAE (2.6), where \bar{C} , \bar{C}' , and $\bar{\mathcal{R}}_1$ are built by using the modified algorithms and present conditions sufficient for the preservation of GMAS properties when we pass from images to preimages.

§3. THE TRANSFORMATION OF CONDITIONS INTO MORE EFFICIENT CRITERIA

The preservation conditions \mathcal{R}_1 and $\bar{\mathcal{R}}_1$, obtained by the algorithm described differ from the morphisms used traditionally in the general algebraic system theory, including universal algebra and model theory. The stage of excluding functional symbols ensures that no functional and relation symbols will appear simultaneously in any concluding (atomic) formula \mathcal{R}_1 . Yet, intense alternation of quantifiers of different types in the condition \mathcal{R}_1 hardens its verification compared to conditions imposed on standard morphisms. Therefore, we seek certain conditions \mathcal{R} that ensure \mathcal{R}_1 and represent more usual morphism-like conditions or obviously follow from them. For this, we use the algorithm for obtaining comparison theorems from comparison lemmas [5, 16]. Like the algorithm in [10], it consists in splitting \mathcal{R}_1 into simpler formulas that include (in some sense) a “minimal” number of existential quantifiers. However, this algorithm is formulated in a more general (in particular, logic but not set theory) form, which deals with formulas with type conditions and concluding statements taken from some fixed variety in the language of systems of processes [10] (and then the logic symbolism \exists, \forall is used merely to shorten expressions of a natural language).

Moreover, the procedure described below allows us to pass eventually to the conjunction of conditions that, when fulfilled simultaneously, are sufficient for \mathcal{R}_1 . Each of them represents a condition of preserving a single symbol of the signature, functional, or relation, in accordance with the traditional definitions of morphisms of algebraic systems.

Henceforth, the following notation will be used. Let Θ^* be some subset of concluding statements of Θ . The expression $[\Theta, \Theta^*]$ denotes the result of excluding from Θ all concluding statements Θ^ν such that $\Theta^\nu \notin \Theta^*$, and also the quantifiers that became unessential during this process, and the logical connectives that lose at least one subformula joined by them in Θ . Let (Θ, Θ^*) denote the result of the replacement in Θ of all concluding statements Θ^ν such that $\Theta^\nu \in \Theta^*$ by an identically true predicate P_t . $\mathcal{A}\{\Theta : \Delta\}$ denotes the result of inversion of the sense of all quantifiers \tilde{w}_α in Θ satisfying the condition Δ . Let $\mathbf{W}(\Theta)$ denote the set of all occurrences of unessential quantifiers in the formula Θ .

The first transformation is splitting the formula \mathcal{R}_1 into parts with respect to its concluding statements. Let $\mathbf{E}(\mathcal{R}_1)$ be the set of all concluding statements of \mathcal{R}_1 , and let a collection of pairwise disjoint sets $E_1, E_2, \dots, E_\rho, \emptyset \neq E_\delta \subseteq \mathbf{E}(\mathcal{R}_1), \delta = 1, \dots, \rho$, be given, with $\rho \geq 0$. For each δ , we specify some condition Δ_δ^A .

The transformation is defined recursively:

$$\begin{aligned} \tilde{\mathcal{U}}_0 &= \mathcal{R}_1, \\ \mathcal{U}_\delta &= \mathcal{A}\{\tilde{\mathcal{U}}_{\delta-1}, \Sigma_\delta\} : \Delta_\delta^A\}, \\ \tilde{\tilde{\mathcal{U}}}_\delta &= \mathcal{A}\{\tilde{\mathcal{U}}_{\delta-1}, \Sigma_\delta\} : \tilde{w}_\alpha \in \mathcal{U}_\delta\} \end{aligned}$$

$\tilde{\tilde{\mathcal{U}}}_\delta$ is the result of replacing of the c-formulas $(\mathcal{R}_1)^\nu \in \Sigma_\delta$ in $\tilde{\mathcal{U}}_{\delta-1}$ by the identically true predicate P_t , followed by application of the operation \mathcal{A} to existential quantifiers that

occur in \mathcal{U}_δ),

$$\tilde{\mathcal{U}}_\delta = H\tilde{\mathcal{U}}_\delta(\mathcal{J}/\mathcal{J} \ \& \ P_t)$$

(in the formula $\tilde{\mathcal{U}}_\delta$ all subformulas $\mathcal{J} \ \& \ P_t$ are replaced with \mathcal{J} , where \mathcal{J} denotes any formula differing from P_t , and then all unessential quantifiers are eliminated in the resulting formula),

$$I_\delta = \&\{H\|\tilde{\mathcal{U}}_\delta, \tilde{w}_\alpha\| \exists z_\alpha Z_\alpha : \tilde{w}_\alpha \in W(\tilde{\mathcal{U}}_\delta)\}$$

(I_δ is the conjunction of the formulas corresponding to the existential quantifiers that were eliminated while constructing conditions $\tilde{\mathcal{U}}_\delta$ from \mathcal{U}_δ).

The following implication holds [5, 16]:

$$\tilde{\mathcal{U}}_\rho \ \& \ \bigwedge_{\delta=1}^\rho \mathcal{U}_\delta \ \& \ I_\delta \rightarrow \mathcal{R}_1.$$

Next, we form the set $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_\rho, \tilde{\mathcal{U}}_\rho\}$. In general, the formulas in \mathcal{U} admit splitting into conjunctions of simpler conditions by means of the transformation below applied to some universal TQs.

In formula $\mathcal{U} \in \mathcal{U}$ we choose some existential quantifiers $\tilde{w}^\tau = \exists w^\tau : Z^\tau$ and for each of them specify a condition Δ_τ , $\tau = 1, \dots, \eta$. We put $\tilde{\mathcal{V}}_0 = \mathcal{U}$,

$$\mathcal{V}_\tau = \mathcal{V}_\tau(\mathcal{U}) = \mathcal{A}\{H \mid \tilde{\mathcal{V}}_{\tau-1}, \tilde{w}^\tau \mid \exists w^\tau Z^\tau : \Delta_\tau\}$$

(in $\tilde{\mathcal{V}}_{\tau-1}$ we select all elements controlling the quantifier \tilde{w}^τ , then build the formula $\mid \tilde{\mathcal{V}}_{\tau-1}, \tilde{w}^\tau \mid \exists w^\tau Z^\tau$, eliminate all unessential quantifiers, and then replace with universal quantifiers only those existential quantifiers that satisfy the condition Δ_τ),

$$\tilde{\mathcal{V}}_\tau = \tilde{\mathcal{V}}_\tau(\mathcal{U}) = \mathcal{A}\{\tilde{\mathcal{V}}_{\tau-1} : \tilde{w} = \tilde{w}^\tau \vee \tilde{w} \in \mathcal{V}_\tau\}$$

(in the formula $\tilde{\mathcal{V}}_{\tau-1}$ we invert the quantifier \tilde{w}^τ and all universal quantifiers that occur in \mathcal{V}_τ), and

$$\mathcal{V} = \mathcal{V}(\mathcal{U}) = \&\{\mathcal{V}_\tau : \tau = 1, \dots, \eta\} \ \& \ \tilde{\mathcal{V}}_\eta.$$

Since $\mathcal{V}(\mathcal{U}) \rightarrow \mathcal{U}$ for all $\mathcal{U} \in \mathcal{U}$ (see, [5, 16]), we take

$$\mathcal{R} = \mathcal{V}(\tilde{\mathcal{U}}_\rho) \ \& \ \bigwedge_{\delta=1}^\rho I_\delta \ \& \ \mathcal{V}(\mathcal{U}_\delta).$$

For $\mathcal{C}, \mathcal{C}', \mathcal{R}_1$ built by the algorithms described in §2, the following theorem is true.

Theorem 1. $\mathcal{C} \ \& \ \mathcal{C}' \rightarrow (\mathcal{R} \rightarrow (\mathcal{F}(\bar{x}) \rightarrow \mathcal{F}'(\overline{f(x)})))$.

So, we have constructed the required formula

$$\mathcal{X} \stackrel{\text{def}}{=} \mathcal{C} \ \& \ \mathcal{C}' \ \& \ \mathcal{R},$$

which is a solution of the LAE (2.1) in question and provides conditions ensuring that, whenever the formula $\mathcal{F}(\bar{x})$ is satisfied at some values of the variables x_1, \dots, x_p in the system \mathfrak{A} , the formula $\mathcal{F}(\bar{x})$ is also satisfied at the corresponding values of CEMs $\langle \varphi \rangle^{|x_1|}(x_1), \dots, \langle \varphi \rangle^{|x_p|}(x_p)$ in \mathfrak{A}' .

Similar theorem may be stated for preservation of properties “in the opposite direction”.

Theorem 2. $\bar{\mathcal{C}} \ \& \ \bar{\mathcal{C}}' \rightarrow (\bar{\mathcal{R}} \rightarrow (\mathcal{F}'(\overline{f(x)}) \rightarrow \mathcal{F}(\bar{x})))$.

Here the formula $\bar{\mathcal{R}}$ is built by the same algorithms but on the basis of $\bar{\mathcal{R}}_1$. So, the formula

$$\bar{\mathcal{X}} \stackrel{\text{def}}{=} \bar{\mathcal{C}} \ \& \ \bar{\mathcal{C}}' \ \& \ \bar{\mathcal{R}}$$

presents a solution of the LAE (2.3) and provides conditions ensuring that, whenever the formula $\mathcal{F}(\bar{x})$ is satisfied at the values of CEMs $\langle \varphi \rangle^{|x_1|}(x_1), \dots, \langle \varphi \rangle^{|x_p|}(x_p)$ in \mathfrak{A}' , the formula $\mathcal{F}(\bar{x})$ is also satisfied at the corresponding values of the variables x_1, \dots, x_p in \mathfrak{A} .

§4. A PROPERTY OF A DISCRETE-EVENT SYSTEM

A (logical) discrete-event system is a quintuple [15]

$$G = (\mathcal{X}, \mathcal{E}, \{f_e\}_{e \in \mathcal{E}}, g, E_v).$$

Here \mathcal{X} is the set of states of the system; \mathcal{E} is the set of events denoted by the letter e with some index; $g : \mathcal{X} \rightarrow 2^{\mathcal{E}} \setminus \{\emptyset\}$ is a so-called *enable function*: $g(x)$ is the set of all events enabled at a current state $x \in \mathcal{X}$. The definition of a logical DES does not involve time (in the common sense, like, e.g., systems of differential equations). However, a kind of analog of the time variable is a counter that shows how many events have passed and, therefore, how many times states have changed. So, let there be a discrete “time” scale in the form of a scale of the counter values $\mathbb{N} = \{0, 1, 2, \dots\}$, and let $x_k \in \mathcal{X}$ present the state of DES at the time moment $k \in \mathbb{N}$ (i.e., when the counter is in the state k).

For each $e \in \mathcal{E}$, the operator $f_e : \mathcal{X} \rightarrow \mathcal{X}$ transfers the system from one state to another whenever an event e occurs. It is natural to require that $f_e(x)$ be defined only if $e \in g(x)$. If at this moment an event $e_k \in \mathcal{E}$ occurs (of course, in the case of $e_k \in g(x_k)$), then the next state x_{k+1} is given by application of the operator f_{e_k} , i.e., $x_{k+1} = f_{e_k}(x_k)$.

Any sequence $\mathbf{x} = \{x_0, x_1, x_2, \dots\} \in \mathcal{X}^{\mathbb{N}}$ such that $x_{k+1} = f_{e_k}(x_k)$, where $e_k \in g(x_k)$ for all k , is called a *state trajectory* (*phase trajectory*). Any sequence $\mathbf{e} = \{e_0, e_1, e_2, \dots\} \in \mathcal{E}^{\mathbb{N}}$ such that there exists a state trajectory $\mathbf{x} \in \mathcal{X}^{\mathbb{N}}$ (where $e_k \in g(x_k)$ and $x_{k+1} = f_{e_k}(x_k)$ for all k) is an *event trajectory*. Therefore, each initial state x_0 and each event trajectory \mathbf{e} determining the order of applying the operators f_{e_k} , give rise to a unique phase trajectory provided $e_0 \in g(x_0)$ (the converse is not true in general).

E ($E \subset \mathcal{E}^{\mathbb{N}}$) denotes the set of all event trajectories. $E_v \subset E \subset \mathcal{E}^{\mathbb{N}}$ is the set of all *valid event trajectories*, i.e., the trajectories that are physically possible in G . Therefore, if $x_k \in \mathcal{X}$ and $e_k \in g(x_k)$, an event e_k may occur only if it lies on a valid event trajectory. In some cases it is appropriate to consider a set $E_a \subset E_v$ of *allowed event trajectories*. It helps to limit the occurrence of events to some required sequence of events.

In the general form, a DES G may be regarded as a triple-based algebraic system $\mathfrak{A} = \langle A, \Omega_F, \Omega_P, \Omega_E \rangle$ with $A = \{\mathcal{X}, \mathcal{E}, \mathbb{N}\}$, $\Omega_F = \{h\}$, $\Omega_P = \{E_v\}$, $\Omega_E = \emptyset$, but representation of a DES as a GMAS may change depending on the formula expressing a given property. It should be noted that, to formalize phase and event trajectories of DESs, the step with the operation of forming sets of sequences is used.

We fix a DES G and consider the property of the existence, for any initial state, of a valid event trajectory such that k^* is the first moment of attaining the given set \mathcal{X}^* by the corresponding phase trajectory, while any other valid event trajectory attains this set later (the property degree is $M = 3$):

$$\begin{aligned} \mathcal{F}(k^*) = & \forall x_0 : \mathcal{X}(x_0) \ \exists \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \ (\mathcal{X}_*(h(x_0, \mathbf{e}^*, k^*))) \ \& \\ (4.1) \quad & (\forall k_1 : k_1 < k^* \ \neg \mathcal{X}_*(h(x_0, \mathbf{e}^*, k_1))) \ \& \\ & (\forall \mathbf{e} : \check{E}_v(x_0, \mathbf{e}) \ \& \ \mathbf{e} \neq \mathbf{e}^* \ \forall k_2 : k_2 \leq k^* \ \neg \mathcal{X}_*(h(x_0, \mathbf{e}, k_2))). \end{aligned}$$

In this case G can be regarded as a general triple-based algebraic system of finite type $\mathfrak{A} = \langle A, \Omega_F, \Omega_P, \Omega_E \rangle$, where $A = \{\mathcal{X}, \mathcal{E}, \mathbb{N}\}$, $\Omega_F = \{h\}$, $\Omega_P = \{\mathcal{X}, \mathcal{X}_*, \check{E}_v, <\}$, $\Omega_E = \emptyset$. Note that $\mathcal{X}^* \subset \mathcal{X}$, and $\mathcal{X}_*(x)$ is true if and only if $x \in \mathcal{X}^*$. Here \mathcal{X} is the unary predicate identically coincident with the state space \mathcal{X} , essential to represent the formula in the constructed type-quantifiers language \mathcal{L} . Assume $\mathcal{X}(x)$ to be true for all $x \in \mathcal{X}$. For the binary predicate \check{E}_v , a pair (x_0, \mathbf{e}) belongs to \check{E}_v if and only if $\mathbf{e} \in E_v(x_0)$, i.e., if \mathbf{e} is a valid trajectory started at the state x_0 ($\check{E}_v \in 2^{\mathcal{X} \times \mathcal{E}^N}$). Finally, $<$ is the relation “less than” on natural numbers.

Let $G' = (\mathcal{X}', \mathcal{E}', \{f'_{e'}\}_{e' \in \mathcal{E}'}, g', E'_v)$ be another DES, and suppose we are given mappings

$$\varphi_1 : \mathcal{X} \rightarrow \mathcal{X}', \quad \varphi_2 : \mathcal{E} \rightarrow \mathcal{E}', \quad \varphi_3 : \mathbb{N} \rightarrow \mathbb{N}'.$$

It is required to find conditions ensuring that, whenever G has property (4.1), the system G' possesses it as well. We are going to use the above-described algorithm of constructing conditions that preserve GMAS properties.

First, we exclude the functional symbol h out of the c-formulas \mathcal{F} , introducing new variables y_1, y_2 , and y_3 , and building the formula Ψ by the algorithm described previously. Then we build the formula \mathcal{R}_1 taking into account that $|k^*| = \mathbb{N}$. Furthermore,

$$\Xi_+^1 = \mathcal{X}_*(y_1) \rightarrow \mathcal{X}'_*(y'_1), \quad \Xi_-^2 = \mathcal{X}'_*(y'_2) \rightarrow \mathcal{X}_*(y_2), \quad \Xi_-^3 = \mathcal{X}'_*(y'_3) \rightarrow \mathcal{X}_*(y_3).$$

Now, $\mathbf{E}(\mathcal{R}_1) = \{\Xi_+^1, \Xi_-^2, \Xi_-^3\}$. Let $E_1 = \{\Xi_+^1\}$, $E_2 = \{\Xi_-^2\}$, $\Delta_1^A = \Delta_2^A = P_f$:

$$\begin{aligned} \mathcal{U}_1 &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \exists x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0 \quad \forall \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \\ &\quad \forall y_1 = h(x_0, \mathbf{e}^*, k^*) \quad \exists (\mathbf{e}^*)' : \check{E}'_v(x'_0, (\mathbf{e}^*)') \ \& \ f_2 \parallel^N(\mathbf{e}^*) = (\mathbf{e}^*)' \\ &\quad \exists y'_1 = h'(x'_0, (\mathbf{e}^*)', f_3(k^*)) : f_1(y_1) = y'_1 \quad (\mathcal{X}_*(y_1) \rightarrow \mathcal{X}'_*(y'_1)), \\ \mathcal{U}_2 &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \forall x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0 \quad \forall \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \\ &\quad \forall (\mathbf{e}^*)' : \check{E}'_v(x'_0, (\mathbf{e}^*)') \ \& \ f_2 \parallel^N(\mathbf{e}^*) = (\mathbf{e}^*)' \\ &\quad \forall k'_1 : k'_1 < f_3(k^*) \quad \forall y'_2 = h'(x'_0, (\mathbf{e}^*)', k'_1) \quad \exists k_1 : k_1 < k^* \ \& \ f_3(k_1) = k'_1 \\ &\quad \exists y_2 = h(x_0, \mathbf{e}^*, k_1) : f_1(y_2) = y'_2 \quad (\mathcal{X}'_*(y'_2) \rightarrow \mathcal{X}_*(y_2)), \\ \tilde{\mathcal{U}}_2 &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \forall x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0 \quad \forall \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \\ &\quad \forall (\mathbf{e}^*)' : \check{E}'_v(x'_0, (\mathbf{e}^*)') \ \& \ f_2 \parallel^N(\mathbf{e}^*) = (\mathbf{e}^*)' \quad \forall \mathbf{e}' : \check{E}'_v(x'_0, \mathbf{e}') \ \& \ \mathbf{e}' \neq (\mathbf{e}^*)' \\ &\quad \forall k'_2 : k'_2 \leq f_3(k^*) \quad \forall y'_3 = h'(x'_0, \mathbf{e}', k'_2) \\ &\quad \exists \mathbf{e} : \check{E}_v(x_0, \mathbf{e}) \ \& \ \mathbf{e} \neq \mathbf{e}^* \ \& \ f_2 \parallel^N(\mathbf{e}) = \mathbf{e}' \\ &\quad \exists k_2 : k_2 \leq k^* \ \& \ f_3(k_2) = k'_2 \quad \exists y_3 = h(x_0, \mathbf{e}, k_2) : f_1(y_3) = y'_3 \\ &\quad (\mathcal{X}'_*(y'_3) \rightarrow \mathcal{X}_*(y_3)). \end{aligned}$$

In this case, $\mathcal{U}_1 \ \& \ \mathcal{U}_2 \ \& \ \tilde{\mathcal{U}}_2 \rightarrow \mathcal{R}_1$ and $\mathbf{U} = \{\mathcal{U}_1, \mathcal{U}_2, \tilde{\mathcal{U}}_2\}$. For splitting the formulas from \mathbf{U} , choose $\Delta_\tau = P_f$ as the parameters for all existential TQs. Then for \mathcal{U}_1 we obtain the

following conditions:

$$\begin{aligned}\mathcal{V}_1(\mathcal{U}_1) &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \exists x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0, \\ \mathcal{V}_2(\mathcal{U}_1) &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \forall x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0 \quad \forall \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \\ &\quad \exists (\mathbf{e}^*)' : \check{E}'_v(x'_0, (\mathbf{e}^*)') \ \& \ f_2\|^{N}(\mathbf{e}^*) = (\mathbf{e}^*)', \\ \mathcal{V}_3(\mathcal{U}_1) &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \forall x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0 \quad \forall \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \\ &\quad \forall y_1 = h(x_0, \mathbf{e}^*, k^*) \quad \forall (\mathbf{e}^*)' : \check{E}'_v(x'_0, (\mathbf{e}^*)') \ \& \ f_2\|^{N}(\mathbf{e}^*) = (\mathbf{e}^*)' \\ &\quad \exists y'_1 = h'(x'_0, (\mathbf{e}^*)', f_3(k^*)) : f_1(y_1) = y'_1,\end{aligned}$$

and the condition $\check{\mathcal{V}}_3(\mathcal{U}_1)$ constructed by inverting all existential TQs in \mathcal{U}_1 .

Now consider the case of $\check{\mathcal{V}}_0 = \mathcal{U}_2$:

$$\begin{aligned}\mathcal{V}_1(\mathcal{U}_2) &= \forall k'_1 : k'_1 < f_3(k^*) \quad \exists k_1 : k_1 < k^* \ \& \ f_3(k_1) = k'_1, \\ \mathcal{V}_2(\mathcal{U}_2) &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \forall x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0 \quad \forall \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \\ &\quad \forall (\mathbf{e}^*)' : \check{E}'_v(x'_0, (\mathbf{e}^*)') \ \& \ f_2\|^{N}(\mathbf{e}^*) = (\mathbf{e}^*)' \quad \forall k'_1 : k'_1 < f_3(k^*) \\ &\quad \forall y'_2 = h'(x'_0, (\mathbf{e}^*)', k'_1) \quad \forall k_1 : k_1 < k^* \ \& \ f_3(k_1) = k'_1 \\ &\quad \exists y_2 = h(x_0, \mathbf{e}^*, k_1) : f_1(y_2) = y'_2,\end{aligned}$$

and the formula $\check{\mathcal{V}}_2(\mathcal{U}_2)$ contains no existential quantifiers.

In the case of $\check{\mathcal{V}}_0 = \check{\mathcal{U}}_2$:

$$\begin{aligned}\mathcal{V}_1(\check{\mathcal{U}}_2) &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \forall x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0 \quad \forall \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \\ &\quad \forall (\mathbf{e}^*)' : \check{E}'_v(x'_0, (\mathbf{e}^*)') \ \& \ f_2\|^{N}(\mathbf{e}^*) = (\mathbf{e}^*)' \quad \forall \mathbf{e}' : \check{E}'_v(x'_0, \mathbf{e}') \ \& \ \mathbf{e}' \neq (\mathbf{e}^*)' \\ &\quad \exists \mathbf{e} : \check{E}_v(x_0, \mathbf{e}) \ \& \ \mathbf{e} \neq \mathbf{e}^* \ \& \ f_2\|^{N}(\mathbf{e}) = \mathbf{e}', \\ \mathcal{V}_2(\check{\mathcal{U}}_2) &= \forall k'_2 : k'_2 \leq f_3(k^*) \quad \exists k_2 : k_2 \leq k^* \ \& \ f_3(k_2) = k'_2, \\ \mathcal{V}_3(\check{\mathcal{U}}_2) &= \forall x'_0 : \mathcal{X}'(x'_0) \quad \forall x_0 : \mathcal{X}(x_0) \ \& \ f_1(x_0) = x'_0 \quad \forall \mathbf{e}^* : \check{E}_v(x_0, \mathbf{e}^*) \\ &\quad \forall (\mathbf{e}^*)' : \check{E}'_v(x'_0, (\mathbf{e}^*)') \ \& \ f_2\|^{N}(\mathbf{e}^*) = (\mathbf{e}^*)' \quad \forall \mathbf{e}' : \check{E}'_v(x'_0, \mathbf{e}') \ \& \ \mathbf{e}' \neq (\mathbf{e}^*)' \\ &\quad \forall k'_2 : k'_2 \leq f_3(k^*) \quad \forall y'_3 = h'(x'_0, \mathbf{e}', k'_2) \\ &\quad \forall \mathbf{e} : \check{E}_v(x_0, \mathbf{e}) \ \& \ \mathbf{e} \neq \mathbf{e}^* \ \& \ f_2\|^{N}(\mathbf{e}) = \mathbf{e}' \\ &\quad \forall k_2 : k_2 \leq k^* \ \& \ f_3(k_2) = k'_2 \quad \exists y_3 = h(x_0, \mathbf{e}, k_2) : f_1(y_3) = y'_3,\end{aligned}$$

So,

$$\begin{aligned}\mathcal{V}(\mathcal{U}_2) &= \mathcal{V}_1(\mathcal{U}_2) \ \& \ \mathcal{V}_2(\mathcal{U}_2) \ \& \ \check{\mathcal{V}}_2(\mathcal{U}_2), \\ \mathcal{V}(\mathcal{U}_2) &\rightarrow \mathcal{U}_2, \\ \mathcal{R} &= \mathcal{V}(\mathcal{U}_1) \ \& \ \mathcal{V}(\mathcal{U}_2) \ \& \ \mathcal{V}(\check{\mathcal{U}}_2).\end{aligned}$$

We interpret the conditions obtained. The first of them, $\mathcal{V}_1(\mathcal{U}_1)$, means the requirement that the mapping $\varphi_1 : \mathcal{X} \rightarrow \mathcal{X}'$ be a surjection.

The conditions $\mathcal{V}_1(\mathcal{U}_2)$ and $\mathcal{V}_2(\check{\mathcal{U}}_2)$ impose constrains on the correspondence of time indices of the trajectories. They can be satisfied automatically by choosing φ_3 to be an identity mapping, which seems natural. Therefore, assume $\varphi_3 = \mathbf{1}$.

The condition $\mathcal{V}_3(\mathcal{U}_1)$ is a requirement for the system operation to be preserved:

$$\varphi_1(h(x_0, \mathbf{e}, k)) = h'(\varphi_1(x_0)\varphi_2\|^{N}(\mathbf{e}), k);$$

it is satisfied if φ is a homomorphism of the GMAS \mathfrak{A} to \mathfrak{A}' . Then the similar conditions $\mathcal{V}_2(\mathcal{U}_2)$, $\mathcal{V}_3(\check{\mathcal{U}}_2)$ are also satisfied.

The condition $\tilde{\mathcal{V}}_3(\mathcal{U}_1)$ may be rewritten in the form $\varphi_1(\mathcal{X}_*) \subseteq \mathcal{X}'_*$. On the other hand, $\tilde{\mathcal{V}}_2(\mathcal{U}_2)$ and $\tilde{\mathcal{V}}_3(\tilde{\mathcal{U}}_2)$ are equivalent to the requirement $\mathcal{X}'_* \subseteq \varphi_1(\mathcal{X}_*)$.

The conditions $\mathcal{V}_2(\mathcal{U}_1), \mathcal{V}_1(\tilde{\mathcal{U}}_2)$ impose requirements on the relationship of event trajectories of the systems under consideration. $\mathcal{V}_2(\mathcal{U}_1)$ may be rewritten in the form $\varphi_1 * \widehat{\varphi_2}^N(E_v) \subseteq E'_v$. The condition $\mathcal{V}_1(\tilde{\mathcal{U}}_2)$ is a little bit more complicated. We rewrite it in an equivalent form in a less formal language:

$$\forall x_0 \in \mathcal{X} \quad \forall \mathbf{e}^* \in E_v(x_0) \quad \forall \mathbf{e}' \in E'_v(f_1(x_0)) \setminus \{f_2\|^N(\mathbf{e}^*)\} \\ (f_2\|^N)^{-1}(\mathbf{e}') \cap E_v(x_0) \setminus \{\mathbf{e}^*\} \neq \emptyset.$$

It is satisfied if $E'_v \subseteq \widehat{\varphi_1 * \varphi_2}^N(E_v)$ or, to be precise, if for all $x_0 \in \mathcal{X}$ we have $E'_v(\varphi_1(x_0)) \subseteq \varphi_2\|^N|_{E_v(x_0)}$, and moreover, $\varphi_2\|^N|_{E_v(x_0)}$ is the injection from $E_v(x_0)$ to $E'_v(\varphi_1(x_0))$. Here $\varphi_2\|^N|_{E_v(x_0)}$ denotes the function whose domain is the restriction of the domain of the function $\varphi_2\|^N$ down to the set $E_v(x_0)$. Taking the condition $\mathcal{V}_2(\mathcal{U}_1)$ into account, we can change the above to the requirement that $\varphi_2\|^N|_{E_v(x_0)}$ be a bijection from $E_v(x_0)$ to $E'_v(\varphi_1(x_0))$.

A family of mappings $\varphi = \{\varphi_\lambda | \varphi_\lambda : A_\lambda \rightarrow A'_\lambda, \lambda = 1, \dots, k\}$ is called a *homomorphism* of a GMAS \mathfrak{A} to \mathfrak{A}' if:

- (1) $\langle \varphi \rangle^{S_{n_\beta+1, \beta}[A]}(\mathbf{F}_\beta^{n_\beta}(z_1, \dots, z_{n_\beta})) = (\mathbf{F}_\beta^{n_\beta})'(\langle \varphi \rangle^{S_{1\beta}[A]}(z_1), \dots, \langle \varphi \rangle^{S_{n_\beta\beta}[A]}(z_{n_\beta}))$
 $(z_1 \in S_{1\beta}[A], \dots, z_{n_\beta} \in S_{n_\beta\beta}[A], \beta = 1, \dots, k_F)$;
- (2) $\langle \varphi \rangle^{T_{1\gamma}[A] \times \dots \times T_{n_\gamma\gamma}[A]}(\mathbf{P}_\gamma^{n_\gamma}) \subseteq (\mathbf{P}_\gamma^{n_\gamma})'$ ($\gamma = 1, \dots, k_P$);
- (3) $\langle \varphi \rangle^{U_\delta[A]}(\mathbf{E}_\delta) = \mathbf{E}'_\delta$ ($\delta = 1, \dots, k_E$).

Theorem 3. *Suppose that $G = (\mathcal{X}, \mathcal{E}, \{f_e\}_{e \in \mathcal{E}}, g, E_v)$ has property (4.1). Suppose $G' = (\mathcal{X}', \mathcal{E}', \{f'_{e'}\}_{e' \in \mathcal{E}'}, g', E'_v)$ is another DES and there exists a family of mappings $\varphi = \langle \varphi_1, \varphi_2, \varphi_3 \rangle$, where $\varphi_1 : \mathcal{X} \rightarrow \mathcal{X}'$, $\varphi_2 : \mathcal{E} \rightarrow \mathcal{E}'$, $\varphi_3 : \mathbb{N} \rightarrow \mathbb{N}'$, and $\varphi_3 = \mathbf{1}$. If φ is a homomorphism of GMASs corresponding to G and G' and surjective with respect to φ_1 , $\mathcal{X}'_* \subseteq \varphi_1(\mathcal{X}_*)$, and for any $x_0 \in \mathcal{X}$ the mapping $\varphi_2\|^N|_{E_v(x_0)}$ is a bijection from $E_v(x_0)$ to $E'_v(\varphi_1(x_0))$, then the system G' also possesses property (4.1).*

The following definition generalizes the well-known definition of a strong homomorphism of single-sorted algebraic systems to many-sorted algebraic systems. We call φ a *strong homomorphism* of GMAS \mathfrak{A} on \mathfrak{A}' if φ is a homomorphism of this GMAS and for all $z_{\alpha_1} \in T_{1\gamma}[A], \dots, z_{\alpha_{n_\gamma}} \in T_{n_\gamma\gamma}[A]$ we have

$$(\mathbf{P}_\gamma^{n_\gamma})'(\langle \varphi \rangle^{T_{1\gamma}[A]}(z_{\alpha_1}), \dots, \langle \varphi \rangle^{T_{n_\gamma\gamma}[A]}(z_{\alpha_{n_\gamma}})) \rightarrow \mathbf{P}_\gamma^{n_\gamma}(z_{\alpha_1}, \dots, z_{\alpha_{n_\gamma}}), \gamma = 1, \dots, k_P.$$

In the formulas $\tilde{\mathcal{V}}_2(\mathcal{U}_2)$ and $\tilde{\mathcal{V}}_3(\tilde{\mathcal{U}}_2)$, let us keep the meaning of the quantifiers of variables y_2 and y_3 unchanged, not including them in the sets of quantifiers \tilde{w}^τ for the formulas \mathcal{U}_2 and $\tilde{\mathcal{U}}_2$. Then these formulas present a condition of strong homomorphism for the systems G and G' .

Theorem 4. *Suppose $G = (\mathcal{X}, \mathcal{E}, \{f_e\}_{e \in \mathcal{E}}, g, E_v)$ possesses property (4.1). Suppose also that $G' = (\mathcal{X}', \mathcal{E}', \{f'_{e'}\}_{e' \in \mathcal{E}'}, g', E'_v)$ is another DES and there exists a family of mappings $\varphi = \langle \varphi_1, \varphi_2, \varphi_3 \rangle$, where $\varphi_1 : \mathcal{X} \rightarrow \mathcal{X}'$, $\varphi_2 : \mathcal{E} \rightarrow \mathcal{E}'$, $\varphi_3 : \mathbb{N} \rightarrow \mathbb{N}'$, and $\varphi_3 = \mathbf{1}$. If φ is a strong homomorphism of GMASs corresponding to G and G' and surjective with respect to φ_1 , and for all $x_0 \in \mathcal{X}$ the mapping $\varphi_2\|^N|_{E_v(x_0)}$ is a bijection from $E_v(x_0)$ to $E'_v(\varphi_1(x_0))$, then the system G' also possesses property (4.1).*

CONCLUSION

In this paper, a new method is presented for generating conditions sufficient for the preservation of system properties under its mappings to another (auxiliary) system. This is done by combining algebraic and logical approaches. A prerequisite for its applicability to dynamical systems is the algebraization of their models. After that, criteria for preservation of properties are generated algorithmically and are formulated in terms of morphisms. The important stage of the algorithm presented is the choice of parameters of splitting the formula \mathcal{R}_1 . In forthcoming papers, it will be shown that the simplest choice of parameters of splitting (called a “standard splitting”) allows us to formulate preservation criteria merely on the basis of the structure of the formula and without numerous formal manipulations. Classes of properties will be determined that are preserved under morphisms of similar type. We shall analyze the relationship between the LAE-method and the representability method [2], which also allows one to generate conditions for preservations of properties of dynamical systems.

The example of studying the DES property has been given only to illustrate the stages of the algorithm. In [12], we investigate the DES-model for the public railway transportation network by using the approach described. The interpretation of the generated conditions made it possible to obtain criteria of nondeterioration for the properties of the timetable when new railway paths are added to the network.

REFERENCES

- [1] N. Bourbaki, *Elements of mathematics. Theory of sets*, Addison-Wesley Publ. Co., Reading, MA etc., Hermann, Paris, 1968. MR0237342 (38:5631)
- [2] S. N. Vasil'ev, *A representability method in logico-algebraic approach to qualitative analysis of dynamical systems*, Optim. Upravl. Intellekt **2005**, no. 3(11), 248–254. (Russian)
- [3] ———, *A reduction method and qualitative analysis of dynamical systems*. I, II, Izv. Ross. Akad. Nauk Teor. Sist. Upr. **2006**, no. 1, 21–29; **2006**, no. 2, 5–17; English transl., J. Comput. Syst. Sci. Int. **45** (2006), 17–25, 167–179. MR2258536 (2007e:93034a); MR2258537 (2007e:93034b)
- [4] ———, *The comparison method in systems analysis*. I–IV, *Differentsial'nye Uravneniya* **17** (1981), no. 9, 1562–1573; **17** (1981), no. 11, 1945–1954; **18** (1982), no. 2, 197–205; **18** (1982), no. 6, 938–947; English transl., *Differ. Equ.* **17** (1981), no. 9, 985–993 (1982); **17** (1981), no. 11, 1222–1230 (1982); **18** (1982), no. 2, 155–161; **18** (1982), no. 6, 657–665. MR0631770 (84a:49056a); MR0638520 (84a:49056b); MR0649663 (84a:49056c); MR0663798 (84a:49056d)
- [5] ———, *Synthesis of theorems with VFL in mathematical systems theory*, Dokt. diss., Irkutsk, 1988. (Russian)
- [6] V. F. Zhuravlev, *Foundations of theoretical mechanics*, 2nd ed., Fizmatlit, Moscow, 2001. (Russian) MR2057304 (2005d:70002)
- [7] A. V. Kavinov and A. P. Krishchenko, *Stability of solutions in different variables*, *Differentsial'nye Uravneniya* **43** (2007), no. 11, 1470–1473; English transl., *Differ. Equ.* **43** (2007), no. 11, 1505–1509. MR2435951 (2009f:34116)
- [8] A. I. Mal'tsev, *Algebraic systems*, Nauka, Moscow, 1970; English transl., Grundlehren Math. Wiss., Bd. 192, Springer-Verlag, New York–Heidelberg, 1973. MR0282908 (44:142); MR0349384 (50:1878)
- [9] ———, *Model correspondences*, *Izv. Akad. Nauk SSSR Ser. Mat.* **23** (1959), no. 3, 313–336. (Russian) MR0120152 (22:10909)
- [10] V. M. Matrosov, *The comparison method in the dynamics of systems*. I, II, *Differentsial'nye Uravneniya* **10** (1974), no. 9, 1547–1559; **11** (1975), no. 3, 403–417; English transl. in *Differ. Equ.* **10** (1974), no. 9; **11** (1975), no. 3. MR0374574 (51:10774)
- [11] N. V. Nagul, *On the preservation of the properties of many-sorted algebraic systems*, *Vestn. Samar. Gos. Univ. Estestvennonauchn. Ser.* **2007**, no. 6, 223–241. (Russian) MR2387789 (2009a:03013)
- [12] ———, *Preservation of properties of schedules in a model of public transport*, *Sovrem. Tekhnol. Sistem. Anal. Modelir.* **2010**, no. 4 (28), 150–159. (Russian)
- [13] S. Feferman, *Harmonious logic: Craig's interpolation theorem and its descendants*, *Synthese* **164** (2008), 341–357. MR2438874 (2009j:03002)
- [14] P. J. Higgins, *Algebras with a scheme of operators*, *Math. Nachr.* **27** (1963), 115–132. MR0163940 (29:1239)

- [15] A. N. Michel, K. Wang, and Bo Hu, *Qualitative theory of dynamical systems*, Monogr. Textbooks Pure Appl. Math., vol. 239, Marcel Dekker, Inc., New York, 2001. MR1816096 (2002e:34001)
- [16] S. N. Vassilyev, *Machine synthesis of mathematical theorems*, J. Logic Programming **9** (1990), no. 2–3, 235–266. MR1072279 (91g:68145)

INSTITUTE FOR SYSTEM DYNAMICS AND CONTROL THEORY, SIBERIAN BRANCH, RUSSIAN ACADEMY OF SCIENCES, LERMONTOVA STR. 134, IRKUTSK, RUSSIA

E-mail address: `sapling@icc.ru`

Received April 1, 2007

Translated by THE AUTHOR