FORMALIZED ANALYSIS OF COMPLEX SYSTEMS. I

P. M. Ivanov

UDC 519.8

Complex system design is treated as a multilevel iterative procedure. An approach to formalized description of the components of a complex system and their interconnections is considered.

INTRODUCTION

The insufficient development of formalized methods of control system design substantially reduces system efficiency and the productivity of personnel responsible for algorithmic and logic design of control systems. The theory of design of complex control systems is considered here as a special subdivision of the general theory of discrete system design. This point of view determines the range of mathematical topics that constitute the theoretical foundation of the proposed methodology of structural and algorithmic design of control systems: problems of analysis and synthesis of discrete transformers, development and analysis of the properties of Glushkov’s system of algorithmic algebras (SAA), and decidability of equivalence in SAA. These problems have been solved by the author in [1, 2]. The use of abstract-algebraic methods for algorithm optimization that characterizes the current stage in the development of the applied theory of algorithms is of fundamental importance for the development of optimal control systems and their algorithmic structures.

Complex system design usually involves the decomposition of the design process into two basic stages: system analysis and system synthesis. We are just now making the first steps on the road to formalization of these stages. An essential issue for the formalization of complex system analysis is formal description of the system in three steps: system structuring, formalization of the system components, and formalization of component interactions.

These topics have been studied by Buslenko and Kovalenko [3], whose theory relies on the description of complex system components as a subassembly and a piecewise-linear subassembly. However, this formalization of system components and their interactions produces a "nonformula" specification of the interconnection scheme (in the form of drawings) and the interconnection operators (in the form of tables). It is thus impossible to formalize the domain of equivalent structural transformations of the interconnection scheme.

The formalization of complex system components and their interactions in terms of so-called R-models proposed in our paper is free from these shortcomings. The R-models are used not only for the description of a deterministic dynamical system operating in discrete time, but also for the description of stochastic dynamical systems that are commonly representable by probabilistic automata. The component interconnection scheme defined in the form of an R-model can be easily used also to find the input and output terminals (the set of contacts of a particular component connected by elementary channels with the contacts of another component). The information about the terminals is used to estimate the intensity of data flows in the control system, to organize data exchange between the components, to estimate the throughput of data transmission channels, to evaluate the coding techniques, etc.

FORMALIZATION OF SYSTEM COMPONENTS

The proposed theory is based on an extended system of algorithmic algebras [2]. Discrete-time deterministic and stochastic systems are usually represented respectively by finite-automaton and probabilistic-automaton systems, which are very popular in system engineering, operations research, and system analysis. These automaton systems obviously do not cover all...
the relevant problems, and once the potential of these models has been exhausted, we have to turn to new mathematical schemes. The regular model scheme appears to be helpful for these purposes. Given a fixed information set and a fixed collection of basis operators and conditions, regular models can be used to describe a wider class of transformations of the information environment than finite discrete transformers.

An analysis algorithm has been obtained for deterministic automata [1]. If we now manage to develop a regular model for a probabilistic automaton, it would be possible to describe complex discrete systems whose components are described by different mathematical schemes (finite and probabilistic automata) in the form of a uniform standard mathematical scheme—the regular model. This (combined with the component interconnection scheme) will lead to an automated simulation model of a complex system.

This problem is solved in the following way. We introduce in SAA a random selection operator $X = \alpha(A \lor B \lor C)$ (where $\alpha$ is a conditional expression, $A$, $B$, $C$ are random variables), which functions as follows: if $\alpha$ is true, then follow the path corresponding to the random process $A$; if $\alpha$ is false, then follow the path of the process $B$; if $\alpha$ is undefined, then follow the process $C$. Then using the probabilistic automaton defined in this way, we construct the table of conditional transition functionals (CTFT) by the method of [4]. A specimen CTFT is shown in Table 1.

In Table 1, $A$ is a probabilistic automaton, $\mu_1$, $\mu_2$, $\mu_3$ are mutually independent random variables taking the values 0, 1, 2 according to their distributions, $a(t)$ and $x(t)$ respectively are the state and the output signal of the automaton at time $t$. When the condition $a(t) = 0 \land x(t) = 0$ is true, $a(t + 1)$ is one of the realizations of the random variable $\mu_2$. We may thus write $a(t + 1) = \mu_2$. If $(a(t) = 0 \land x(t) = 1) = 1$, then the automaton goes with certainty to state 2. The remaining cells in the table are described similarly.

Then, given such a transition table, we construct the regular expression in SAA which is defined by this automaton. This CTFT is described by the regular model

$$t = 0 \{ a(t + 1) = (\mu_2 \lor (\mu_1 \lor (2 \lor (\mu_3 \lor \emptyset))) \} (t : = t + 1),$$

where $a_i$, $x_j$ are respectively denoted by $a(t) = i$, $x(t) = j$ ($i = 0, 1, 2, j = 0, 1$) and the dot stands for conjunction. It thus becomes possible to optimize the automaton by different criteria through equivalent transformations of this expression.

If we recall the process of execution of a regular model and omit the assignment operators, then using the corresponding SAA formulas we can transform this expression into an essentially more compact process:

$$\{ (\mu_2 \lor 2) \lor (\mu_1 \lor (\mu_3 \lor \emptyset)) \}.$$