The Decision of a Goal-Directed Behavior Problem in a Class of Systems Described by one-to-one Transformations

A.A.Sytnik, N.S. Vagarina

State Socio-Economic University,
Radischeva Str. 89, Saratov, 410000, Russia
e-mail: vagarina@ssea.runnet.ru

Abstract
In the Theory of Systems attention is traditionally directed to the problems of control, diagnostics and correction of complex systems behavior. At present a devices and technologies are very complicate and the problem of generating goal-directed behavior is one of prior.
In the general case (that is, for an arbitrary complex system) the problem of generating goal-directed behavior is algorithmically insoluble, but it can be solved by imposing certain restrictions on the system behavior. Approach to a decision of referred above problem in class of intellectual systems described by one-to-one transformations (without loss of information) is offered in this scientific work.

Keywords: complex systems, goal-directed behavior, finite-state machine, systems described by one-to-one transformations.

1 Introduction

In the course of complex systems exploitation in time their original behavior transformation can occur. The character of systems material nature determines it. J.von Neumann pointed out in his work "...multipliers troubles... are essential and integral part of their work".

Nowadays a tendency to transfer functions of effective functioning security inside the system increases. More often for behavior modification two main superfluity types – structural and temporal - are used. However, death of structural reservation begets the question: "Is it possible to use characteristics of current automatic machine function law for formation of required reactions totality on exits?" To answer this question we should study functional system superfluity in existence and also possible variants of its goal-seeking creation on the point of designing. Behavior restoration, realizing on the indicated principles, is called functional. Functional behavior restoration task is not reduced only to the problem of proper behavior restoration, in more general respect, it's a task of a complex system arbitrary flowing behavior reduction to the specified state. Formal functional behavior restoration target setting was made by A.A.Sytnik. Investigations, carrying out in this work, deal with the questions of functional behavior restoration exactly, consisting in attempts to restore

proper functioning due to usage of only behavior characteristics and object's peculiarities.

In a general case for an arbitrary complex system the task of behavior restoration can't be solved algorithmically, but it can be solved at superposition of definite restrictions on system's behavior (and, consequently, on mathematical model of its behavior). In this work systems behavior restoration without loss of information is investigated, and as the system's without loss of information mathematical model a finite-state machine is chosen (FSM), which is the most commonly used mathematical model at description of discontinuous type complex systems. The works of such specialists as M.A.Izerman, M.Arbib, Ya.M.Barzdin, B.A.Trakhtenbrot, A.M.Bogomolov, D.V.Speransky, V.I.Varshavsky, M.A.Gavrilov, V.M.Glushkov, A.Gill, I.E.Kobrinsky, V.B.Kudryavtsev, O.P.Kuznetsov, V.G.Lazarev, M.Minsky, C.Shennon, J.von Neumann, A.A.Sytnik, V.A.Tverdokhlebov, J.Ulman, M.L.Tsetlin, S.V.Yablonsky etc. are devoted to investigation of general machines theory and possibility of its applied use. In FSM terms transfer from defective system's behavior to proper one realizes by machines behavior type variation.

Traditionally model-based objects behavior is considered from a converting point of view, i.e. mechanism of entrance successions (effects) transformation into exit successions (reactions) is studied. However, to get full and thorough idea of system's functional abilities, it is necessary to consider feedback too – finding of entrance effects totality, inducing the adjusted exit signal. If machine description (as formal system model) is multitude of exit successions, which it generates, then the question is of machine's behavior enumerative form. The phrase "to restore proper behavior", in this case, means substitution of a "defective" entrance signal for a "proper" entrance effect, which provokes the necessary reaction.

Fundamental basis for behavior restoration task solution is the universal machines theory. Its origin is connected with the works of C.Shennon, M.Minsky, J.von Neumann, who outlined the main investigation directions.

Behavior self-recovery suggests possibility of getting necessary functioning law from current law, originated as a result of a defect. That is, in terms of the universal machines theory, the universal machine describes behavior of the device, which can be tuned to modeling of family functioning machines laws. As A.A.Sytnik showed, the task of universal machine construction, regarding to an arbitrary FSM family, can't be solved algorithmically. However, the task of universal finite-state machine construction, regarding to a finite family of finite-state machines, can be solved algorithmically. Thus, possibility of behavior restoration, regarding to the adjusted defects class, suggests criteria finding of universal machine synthesis possibility for adjusted class. Synthesis task solution - universal machine – from the substantial point of view, can substitute any machine of an adjusted class at defects origin.

Since every FSM begets a certain subsemigroup of a symmetric semigroup (corresponding to the whole "spectrum" of finite-state machine transformations), then the corresponding universal machine semigroup must be symmetric. Thus, universal machine synthesis task is divided into two subtasks:

- construction of a symmetric semigroup generative multitude, which determines the number of universal machine entrance signals (each element of generative multitude is induced by a certain entrance signal) (possible approaches to the decision of this problem were studied at various times by J.N. Reni, F. Gecheg, S. Pikar and others);
- realization of a symmetric semigroup arbitrary transformation production mechanism from constitutive multitude elements in a universal machine.

Both from theoretical and from practical points of view, studying of the case, when a semigroup is a group, is rather interesting. Under this type of machines the so-called machines without loss of information (described by one-to-one transformations) fall, and they, exactly, are considered in this work. For this class of machines probably construction the special type of universal enumerator of a kind of $\tilde{M}'_I$=(S,X,(δ_1, δ_2)).

2. Method of universal enumerator synthesis for machines without loss of information class

Target setting.

A finite-state machine is given M=(S,X, δ) with states multitude S=(0,...n-1), entrance symbols multitude X=(x_0,...,x_p) and transformation function $\delta = \{\delta_x\}, x \in X$.

where $\delta_x = \begin{pmatrix} s_0 \ldots . s_{n-1} \\ s_{i_0} \ldots . s_{i_{n-1}} \end{pmatrix}$. Its possible defects are described by the machines class $\{\tilde{M}_i = (S,X,\tilde{\delta}^i)\}_{i \in I}$ (I – a certain indexes multitude) with a transformation function $\{\tilde{\delta}^i{}_x\}$, $x \in X$. It is necessary to construct a machine $\tilde{M}'_I$=(S,X,(δ_1, δ_2)), which is a universal enumerator for $I \cup \{M\}$.

Entrance.

Finite-state machine M=(S,X, δ) and its possible defects class $\{\tilde{M}_i = (S,X,\tilde{\delta}^i)\}_{i \in I}$

Exit.

Finite-state machine $\tilde{M}'_I$=(S,X,(δ_1, δ_2)), is a universal enumerator for $I \cup \{M\}$.

Step 1.

a) Suppose j=1

b) Choose a finite-state machine $\tilde{M}_j = (S, X, \tilde{\delta}_x)$ from defects class $\{\tilde{M}_i = (S,X,\tilde{\delta}^i)\}_{i \in I}$.

c) Choose arbitrary machine substitution $\delta_1 \in \{\tilde{\delta}_x\}_{x \in X}$

Step 2

a) if $\delta_1=\begin{pmatrix}0 & 1\ldots n-1\\ 1 & 2\ldots\ 0\end{pmatrix}$, then move to b), otherwise f).

b) Suppose k=0, move to c).

c) If $\exists x\in X$, such as $\delta_x=\begin{pmatrix}0 & 1\ldots & k & k+1\ldots n-1\\ 0 & 1\ldots k+1 & k & \ldots n-1\end{pmatrix}\in\{\tilde{\delta}_x\}$, then move to e), otherwise suppose k=k+1 and move to d).

d) If k≤n-1, then repeat c), otherwise move to STEP 3.

e) Suppose $\delta_2=\delta_x$ and move to Step 9.

f) If $\exists x\in X$, such as $\delta_x=\begin{pmatrix}s_o\ldots s_{i_1}\ldots s_{i_2}\ldots s_{i_3}\ldots s_{n-1}\\ s_0\ldots s_{i_2}\ldots s_{i_3}\ldots s_{i_1}\ldots s_{n-1}\end{pmatrix}\in\{\tilde{\delta}_x\}$, where s_{i_1},s_{i_2},s_{i_3} - three different states of the machine $\tilde{M}$ and number n is even, then suppose $\delta_2=\delta_x$ and move to Step 5.

Step 3

a) Suppose k=0, move to b).

b) If $\exists x\in X$, such as $\delta_x=\begin{pmatrix}0 & 1\ldots & k & k+1 & k+2\ldots n-1\\ 0 & 1\ldots k+1 & k+2 & k & \ldots n-1\end{pmatrix}\in\{\tilde{\delta}_x\}$ and $n-$ even number, then move to d), otherwise suppose k=k+1 and move to c).

c) If k≤n-1, then repeat b), otherwise move to STEP 4.

d) Suppose $\delta_2=\delta_x$ and move to Step 9.

Step 4

a) Suppose k=1, m=0 and move to b).

b) If $\exists x\in X$, such as $\delta_x=\begin{pmatrix}m & \ldots m+k-1\\ m+1\ldots & m\end{pmatrix}\in\{\tilde{\delta}_x\}$ and one of numbers n, k is even, then move to e), otherwise move to c).

c) Suppose $m=m+1$. If $m\le n-1$, then repeat b), otherwise move to d).

d) Suppose $m=0$, $k=k+1$. IF $k<n-1$, then repeat b), otherwise move to Step 4.

e) Suppose $\delta_2=\delta_x$ and move to Step 9.

Step 5

a) If $\exists x\in X$, such as $\delta_x=\begin{pmatrix}s_o\ldots s_{i_1}\ldots s_{i_2}\ldots s_{i_3}\ldots s_{n-1}\\ s_0\ldots s_{i_2}\ldots s_{i_3}\ldots s_{i_1}\ldots s_{n-1}\end{pmatrix}\in\{\tilde{\delta}_x\}$, where s_{i_1},s_{i_2},s_{i_3} - three different machine states $\tilde{M}$ and number n is even, then move to b), otherwise move to Step 6.

b) If $D(\overline{i_1i_2},\overline{i_2i_3},n-1)=1$, then move to c), otherwise suppose $\delta_2=\delta_x$ and move to Step 6.

c) Suppose $\delta_2 = \delta_x$ and move to step 9.

Step 6

a) If $\exists x \in X$, such as $\delta_x = \begin{pmatrix} 0 & 1 \ldots s_m \\ 1 & 2 \ldots 0 \end{pmatrix} \begin{pmatrix} s_m+1 \ldots n-1 \\ s_m+2 \ldots s_m+1 \end{pmatrix} \in \{\tilde{\delta}_x\}$ and s_m – is such machine state M_j that at least one of the numbers i_1, i_2, and i_3 in substitution δ_2 is less or equal to m and one is more than m, then move to b), otherwise to Step 7.

b) If $D(m,n\text{-}1\text{-}m,d)=1$, where d – is an absolute value of numbers difference of those states from s_{i_1}, s_{i_2}, and s_{i_3}, which are contained in one and the same cycle of machine substitution δ_x, then move to c), otherwise to Step 7.

c) Suppose $\delta_1 = \delta_x$ and move to Step 9.

Step 7

a) Suppose k=3, move to b).

b) If $\exists x \in X$, such as $\delta_x = \begin{pmatrix} 0 & 2 \ldots s_{k-1} \\ 1 & 2 \ldots 0 \end{pmatrix} \in \{\tilde{\delta}_x\}$ and k is odd, then suppose $\delta_1 = \delta_x$ and move to d), otherwise to c).

c) suppose $k=k+1$, and if $k<n$, then move to b), otherwise to Step 8.

d) suppose i=0 and move to e).

e) If $\exists x \in X$, such as $\delta_x = \begin{pmatrix} s_i & s_{i+1} \\ s_{i+1} & s_i \end{pmatrix} \begin{pmatrix} s_{k-1} & s_k \\ s_k & s_{k-1} \end{pmatrix} \in \{\tilde{\delta}_x\}$, then suppose $\delta_2 = \delta_x$ and move to g), otherwise to f).

f) suppose $i=i+1$, and if $i \le k$-3, then move to e), otherwise to Step 8.

If δ_1 and δ_2 is not equal

$$\delta_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \end{pmatrix} \text{ and } \delta_2 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 45 \\ 54 \end{pmatrix},$$

$$\delta_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 & 0 \end{pmatrix} \text{ and } \delta_2 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 67 \\ 76 \end{pmatrix},$$

$$\delta_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 & 0 \end{pmatrix} \text{ and } \delta_2 = \begin{pmatrix} 3 & 4 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} 67 \\ 76 \end{pmatrix},$$

then move to Step 9, otherwise to Step 10.

Step 8

a) Suppose k=1, move to b).

b) If $\exists x \in X$, such as $\delta_x = \begin{pmatrix} 0 & 2 \ldots s_k \\ 1 & 2 \ldots 0 \end{pmatrix} \in \{\tilde{\delta}_x\}$, then suppose $\delta_1 = \delta_x$ and move to d), otherwise to c).

c) suppose $k=k+1$, and if $k<n-1$, then move to b), otherwise to Step 10.

d) suppose $i=0$ and move to e).

e) If $\exists x \in X$, such as $\delta_x=\begin{pmatrix} s_i & s_{i+1} \dots s_k & s_{k+1} \dots s_{n-1} \\ s_{i+1} & s_{i+2} \dots s_{k+1} & s_{k+2} \dots s_i \end{pmatrix} \in \{\tilde{\delta}_x\}$, then suppose $\delta_2=\delta_x$ and move to g), otherwise to f).

f) suppose $i=i+1$, and if $i \le k$, then move to e), otherwise to Step 10.

g) If δ_1 and δ_2 is not is not equal

$$\delta_1=\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \end{pmatrix} \text{ and } \delta_2=\begin{pmatrix} 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 2 \end{pmatrix},$$

$$\delta_1=\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \end{pmatrix} \text{ and } \delta_2=\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix},$$

$$\delta_1=\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \end{pmatrix} \text{ and } \delta_2=\begin{pmatrix} 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 2 \end{pmatrix}$$

and at least one of the numbers k, $n-i$ is even, then move to Step 9, otherwise to Step 10.

Step 9. Result: machine $\tilde{M}_l'=(S,X,(\delta_1,\delta_2))$, is a universal enumerator for $I \cup \{M\}$.

Step 10.

a) suppose $j=j+1$

b) if $j \in I$, then move to b) of Step 1, otherwise to Step 11.

Step 11. Result: there is no finite-state machine of the form $\tilde{M}_l'=(S,X,(\delta_1,\delta_2))$, which is a universal enumerator for $I \cup \{M\}$.

3. Conclusion

In given clause the method of universal enumerator synthesis for machines without loss of information class is presented. It grows out scientific researches of authors [2],[3],[4]. It's Novelty consists in ability of construction all special type universal enumerator of a kind of $\tilde{M}_l'=(S,X,(\delta_1,\delta_2))$.

The main result of this work is the form finding of machine substitutions and machine universality conditions for model-based class systems without loss of information, that is of a theoretical interest and can be useful from the practical point of view at complex systems behavior restoration task solution. Namely, the results of carried out investigations allow to solve such important practical tasks as:

- complex systems proper functioning restoration, when it is impossible or inexpediently to carry out damage control immediately, and when it is impossible to attach a reserve onstituent or, if it fell out;

- complex systems behavior modification organization without their physical redesign, only at the expense of current functional possibilities usage;

- fault-tolerant systems working out, which use the whole spectrum of potential means for their behavior restoration – from traditional reservation to peculiarities with existing functioning law.

References

Vagarina N., Sytnik A. (2002) «The approach to the decision of intellectual systems behavior correction problem».V International Congress on mathematical modelling (30.09.02-06.10.02, Dubna),V.2, p. 132

Vagarina N. (2002) «Groups of automatic transformations at synthesis enumerators». «Theoretical problems of computer science and its appendices» № 5, Saratov, p. 42-47

Vagarina N., Sytnik A. (2004) «The problem of goal-seeking behavior in complex systems class without loss of information». «Artificial intellect», №4, p. 100-107.

Vagarina N., Sytnik A. (2004) «Models of enumerators at designing failure-safe discrete systems». V International Conference «Automation of designing of discrete systems», Minsk, V.1, p. 79-86.

Arbib M. A., ed., (1968) «Algebraic theory of machines, languages and semigroups», Acad. Press, New York \& London, 335 p.

Gill A. (1966) «Introduction to finite machines theory», Moskow, Nauka, 272 p.

Kudryavtsev V.B., Aleshin S.V., Podkolzin A.S. (1985) «Introduction to the machines theory», Moskow, Nauka, 319 p.