Deriving distinguishing sequences for input/output automata

Igor Burdonov
Software Engineering department
Ivannikov Institute for System Programming of RAS
Moscow, Russia
igor@ispras.ru

Nina Yevtushenko
Software engineering deparment
Ivannikov Institute for System Programming
Moscow, Russia
nyevtush@gmail.com

Alexandr Kossachev
Software Engineering department
Ivannikov Institute for System Programming of RAS
Moscow, Russia
kos@ispras.ru

***Abstract* — Input/Output (I/O) automata are widely used when deriving high quality tests for (components of) complex discrete systems based on so called distinguishing sequences. For I/O automata, the number of distinguishability relations is much bigger than for classical deterministic Finite State Machines (FSM). In order to avoid submitting the same test sequence several number of times, i.e., avoid the "all weather conditions" assumption, the separability relation can be considered. If two Input/Output automata are separable then there exists an input sequence such that after submitting this sequence and observing produced outputs it can be uniquely concluded which automaton is under testing. In this paper, we modify the discipline of applying input sequences and discuss the derivation of separating sequences for automata with mixed states, i.e., states where transitions both under inputs and under outputs are defined, as well as with cycles labeled by outputs. We also illustrate how an adaptive separating sequence can be derived when both automata are input-enabled.**

Keywords—Finite State Machine (FSM), Input/Output (I/O) automata, distinguishing sequence

I. Introduction

When checking functional and non-functional requirements for complex programmable systems, a so-called mutation testing approach is widely used. The system specification is mutated using most crucial faults and an input sequence is derived that distinguishes the specification and its mutation if such a sequence exists, i.e., the distinguishability (separability) notion allows to formally define the conformance relation between the specification and an implementation under test. Distinguishing sequences are thoroughly studied for deterministic complete Finite State Machines (FSMs) [1] when at each state for each input there is exactly one transition. However, components of complex discrete systems can be partial and have the non-deterministic behavior. Moreover, the requirement to have an output directly after each input is too strong, since a system under test can response with an output (sequence) only to a sequence of inputs. Therefore, Input/Output (I/O) automata [2, 3] are more appropriate when describing the behavior of components of complex discrete systems. Nevertheless, the number of studies of distinguishing sequences for this model is much less.

For deterministic complete FSMs the distinguishability relation means that an implementation is not equivalent to the specification FSM that is there exists an input sequence such that the specification and implementation FSMs have different output responses to this sequence. For transition systems which have the nondeterministic behavior and are only partially specified, the number of distinguishability relations is much bigger. Moreover, in this case, a distinguishing sequence with respect to the nonequivalence or non-reduction relations has to be applied several times while its length can be exponential with respect the the number of states of the specification if the specification is not observable [4, 5]. In order to avoid submitting the same test sequence several number of times the separability relation can be considered [6]. If two Input/Output automata are separable then there exists an input sequence such that after submitting this sequence and observing produced outputs it can be uniquely concluded which automaton is under experiment. In [7], such sequences are considered for I/O automata without mixed states, i.e., states where transitions both under inputs and under outputs are defined and there are no cycles labeled by outputs. In this paper, we extend the set of considered automata modifying the discipline of applying input sequences. We also illustrate how an adaptive separating sequence can be derived when both automata are input-enabled.

The rest of the paper is structured as follows. Section 2 contains the preliminaries. Section 3 illustrates how an (adaptive) separating sequence can be derived for a proper class of automata when at each state either input actions or output actions are specified. Section 4 proposes a method for deriving a separating sequence for general I/O automata.

II. Preliminaries

A finite Input/Output (I/O) automaton or simply an automaton throughout this paper is a 4-tuple $\boldsymbol{S} = (S, s_0, I, O, h_S)$ where S is a nonempty finite set of states with the designated initial state s_0, I and O are finite input and output alphabets, $I \cap O = \varnothing$, and $h_S \subseteq S \times (I \cup O) \times S$ is the transition relation. There exists a transition from state s to state s' under action a if and only if a triple $(s, a, s') \in h_S$. A state of the automaton is a *mixed* state if at the state, transitions under both inputs and outputs are defined. The automaton is *observable*[1] if at each state under each action there exists at most one transition. The automaton is *nondeterministic* if at some state several output actions are specified [5]. In this paper, we consider only observable possibly nondeterministic automata if the converse is not directly stated. A trace of the automaton is a sequence of actions of $I \cup O$ that is permissible at the initial state. Given a trace σ, σ_{in} and σ_{out} are input and output projections of trace σ.

Denote S_{in} a subset of states where transitions under outputs are not specified. A trace at the initial state is *complete* if it is terminated at a state of the set S_{in}. In order to be able to observe such traces, a designated quiescence output $\delta \notin I \cup O$ is introduced [2, 3]; in other words, at each state where transitions under outputs are not specified a loop labeled by δ is added. The obtained automaton is denoted by $\mathbf{S}^{\delta}$ and by definition, δ is considered as an output. Therefore, a trace σ of $\mathbf{S}$ is complete if and only if $\mathbf{S}^{\delta}$ has a trace $\sigma\delta$, the latter corresponds to the fact that after this trace no output of the set O can be produced. Traces of $\mathbf{S}$ and $\mathbf{S}^{\delta}$ are closely related: given a trace of $\mathbf{S}^{\delta}$, after deleting δ a trace of the automaton $\mathbf{S}$ is obtained, and vice versa, given a trace σ of $\mathbf{S}$, if any number of δ are added after any prefix of σ that is complete then a trace of $\mathbf{S}^{\delta}$ is obtained.

III. AUTOMATA WITHOUT MIXED STATES AND CYCLES WITH OUTPUTS

A. *Preset separability*

When using "white model" based testing there is a need to distinguish two automata by an experiment. Two automata $\mathbf{S}$ and $\mathbf{P}$ are separable then an automaton under experiment can be uniquely recognized after applying a separating input sequence α and observing a corresponding output sequence. If automata $\mathbf{S}$ and $\mathbf{P}$ have no mixed states and cycles labeled by outputs then in [7], a method is proposed how to check if two automata are separable and if yes how to derive a separating sequence under the *following hypothesis about applying input sequences* [9]. Before applying the first or the next input the tester waits an appropriate timeout T_{out}, i.e., the separating experiment with an automaton is organized in the following way. The tester waits for the timeout T_{out}, if a system under test produces an output then the timer advances from zero and the tester again waits until the timeout T_{out} expires. If during the timeout there is no output produced then the system is assumed to produce δ. After this, the tester applies the next input and waits again for T_{out}. Under this assumption, the separability problem can be solved for FSMs $\boldsymbol{M_S}$ and $\boldsymbol{M_P}$ which can be constructed for automata $\mathbf{S}$ and $\mathbf{P}$.

The set of $\boldsymbol{M_S}$ states is the set $S_{in} \cup \{s_0\}$; the initial state of $\boldsymbol{M_S}$ is s_0. FSM $\boldsymbol{M_S}$ is a 5-tuple $(S_{in}, s_0, I \cup \{null\_in\}, O \cup O^2 \cup \ldots \cup O^{ns} \cup \{\delta\}, T_{MS})$, $null\_in \notin I$, where ns is maximum length of a trace of $\mathbf{S}$ that has only outputs. The transition relation T_{MS} is the following. For each state $s \in S_{in}$ such that $(s, i, s') \in T_S$, $s' \in S_{in}$, T_{MS} has the transition (s, i, δ, s'), and for each state $s \in S_{in}$ such that $(s, i, s') \in T_S$, $s' \notin S_{in}$, T_{MS} has a transition $(s, i, o_1\, o_2\ldots o_k, s'')$, $k \le ns$, where $s'' \in S_{in}$ is the $o_1\, o_2\ldots o_k$-successor of state s'. If the initial state of $\mathbf{S}$ is not in S_{in}, then T_{MS} has a transition $(s_0, null\_in, o_1\, o_2\ldots o_k, s)$, where $s \in S_{in}$, and s the $o_1\, o_2\ldots o_k$-successor of state s_0.

If the automaton $\mathbf{S}$ ($\mathbf{P}$) is observable then the corresponding FSM $\boldsymbol{M_S}$ ($\boldsymbol{M_P}$) is also observable but can be partial and nondeterministic. In [7], it is shown that automata $\mathbf{S}$ are $\mathbf{P}$ are separable if and only if FSMs $\boldsymbol{M_S}$ and $\boldsymbol{M_P}$ are separable. For checking the FSM separability and deriving a separating sequence, a method from [10] can be utilized. Let $\boldsymbol{M_S}$ and $\boldsymbol{M_P}$ be separable with a separating sequence α. If α is headed by an input of I, then α is a separating sequence for automata $\mathbf{S}$ and $\mathbf{P}$. If $\alpha = null\_in\ \beta$ where $null\_in$ is a so-called empty input then β is a separating sequence for $\mathbf{S}$ and $\mathbf{P}$. If a separating sequence is applied to an automaton under experiment that is $\mathbf{S}$ or $\mathbf{P}$, then under the above hypothesis for applying input sequences it is possible to uniquely determine which automaton is under experiment.

Example 1. For automata $\mathbf{S}$ and $\mathbf{P}$ in Figs. 1a and 2a with initial states 1 and *a*, the corresponding FSMs $\boldsymbol{M_S}$ and $\boldsymbol{M_P}$ are in Figs. 1б and 2б. These FSMs are not separable and thus, automata $\mathbf{S}$ and $\mathbf{P}$ also are not separable.

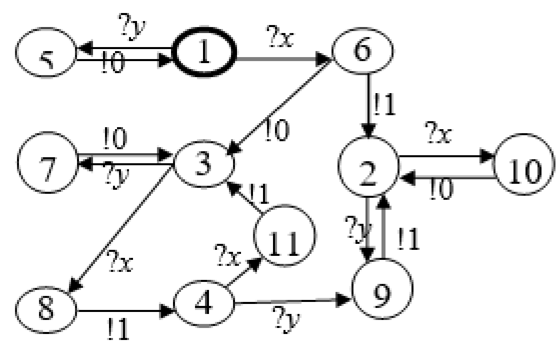

Fig. 1a. Automaton $\mathbf{S}$

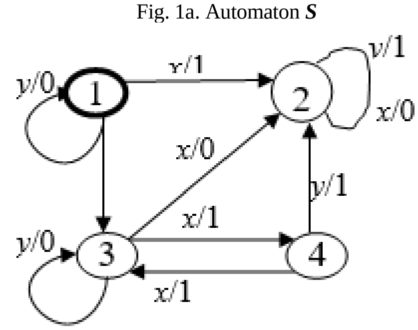

Fig. 1b. FSM M_S

---

[1]Very often such an automaton is called deterministic [8]. However, we use the word «deterministic» for observable automata where at any state at most one output is specified.

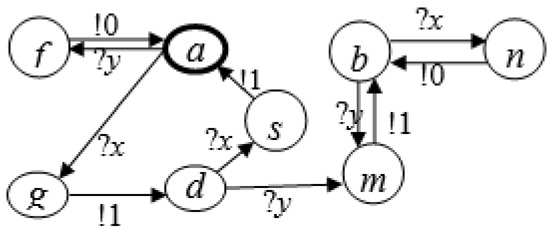

Fig. 2a. Automaton ***P***

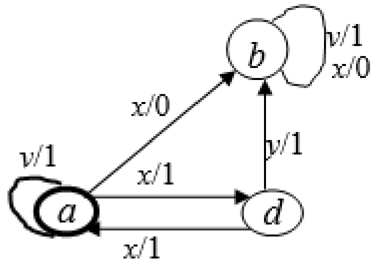

Fig. 2b. FSM M_P

B. Adaptive separability of automata without mixed states

In the paper [13], it is shown that the length of a separating sequence when it exists can exponentially depend on the number of states at least of one automaton. In order to reduce such length, an adaptive separating experiment with automata can be considered where a separating sequence becomes adaptive, i.e., the next input significantly depends on the outputs to the previous inputs. An adaptive separating sequence can be represented by an acyclic automaton without mixed states: at a state where outputs are not defined there is at most one input while at each intermediate state where outputs are defined there is a transition under each output including δ.

An automaton $\boldsymbol{T} = (T, t_0, I, O \cup \{\delta\}, h_T)$ without mixed states that has an acyclic transition graph is a *test case* for automata defined over input alphabet *I* and output alphabet *O* if at each non-deadlock state, either at most a single input or all the outputs are defined and each complete trace is tailed by an output or δ.

Automata ***S*** and ***P*** without mixed states which are defined over input alphabet *I* and output alphabet *O* are *adaptively separable* if there exists a test case such that each complete trace is at most in one of these automata. Automata are *adaptively nonseparable* if for each test case there exists a complete trace that is a trace of both automata.

Since there is one-to-one correspondence between traces of automaton $\boldsymbol{S}^{\delta}$ ($\boldsymbol{P}^{\delta}$) and FSM M_S (M_P) [7], automata ***S*** and ***P*** are adaptively separable if and only if FSMs M_S and M_P are adaptively separable. If at each state of sets of S_{in} и P_{in} of observable automata ***S*** and ***P*** the behavior is defined under each input then FSMs M_S and M_P are complete and observable, and for checking their adaptive separability the following theorem can be used.

Theorem 1 [11, 12]. Observable complete initialized FSMs are adaptively separable if and only if their intersection has no complete submachines.

Example 1 (continuing). The intersection of FSMs M_S and M_P in Figures 1b and 2b is shown in Figure 3a. Given a pair of states, there is an undefined transition in the intersection if FSMs at these states have no common outputs. A test case representing an adaptive separating sequence is shown in Figure 3b.

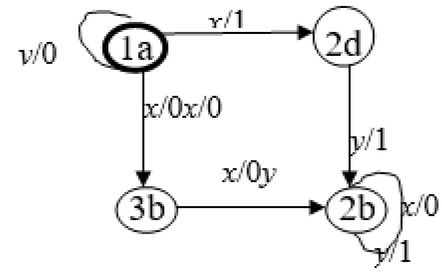

Fig. 3a. The intersection of M_S и M_P (a)

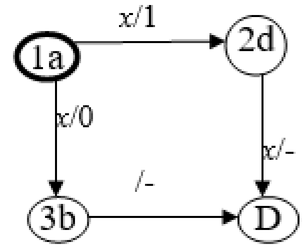

Fig. 3b. A test case representation of an adaptive separating sequence

IV. SEPARATING AUTOMATA WITH MIXED STATES

Consider an automaton that has a state where both inputs and outputs are defined. In order to avoid races between inputs and outputs at such a state, another input timeout T_{in} is introduced. Until this timeout expires, the automaton expects an input. If there is no input before the timeout T_{in} expires then the automaton produces one of the prescribed outputs and moves to the next state or produces a quiescence output δ when the timeout T_{out} expires. Thus, by definition, T_{out} is always bigger than T_{in}. Given the timeout T_{in}, the timer starts to advance from zero after submitting an input or observing an output. When δ is produced, an input can be applied at any time instance.

In order to derive a separating sequence for automata with mixed states or with cycles labeled by output actions, we propose to transform such an automaton ***S*** into an automaton $\boldsymbol{S}^{\omega}$ without mixed states. For this purpose, we add a special input ω into ***S***: the input ω means that we need to wait T_{out} without submitting any input. Correspondingly, at each state *s* where outputs are defined, a transition to state *s*' is added under new input ω. All the transitions under outputs at *s* and only they are moved to state *s*'. Thus, the number of states in $\boldsymbol{S}^{\omega}$ is increased by the number of states where there are transitions under outputs, i.e., at most twice. An automaton $\boldsymbol{S}^{\omega}$ has no mixed states neither cycles labeled by output actions. For each state of $\boldsymbol{S}^{\omega}$ where are no transitions under outputs and ω, a loop labeled by ω is added to $\boldsymbol{S}^{\omega}$. At the next step, FSM M^{ω}_S is derived for the automaton $\boldsymbol{S}^{\omega,\delta}$ (Section 3a) with a small exception. If there is a loop at state of $\boldsymbol{S}^{\omega,\delta}$ labeled by ω и δ, then at this state a loop labeled by ω/δ is added to the FSM M^{ω}_S.

Example 2. Consider an automaton ***Q*** in Figure 4a for which an automaton $\boldsymbol{Q}^{\omega}$ is shown in Figure 4b while the FSM M^{ω}_Q is in Figure 4c.

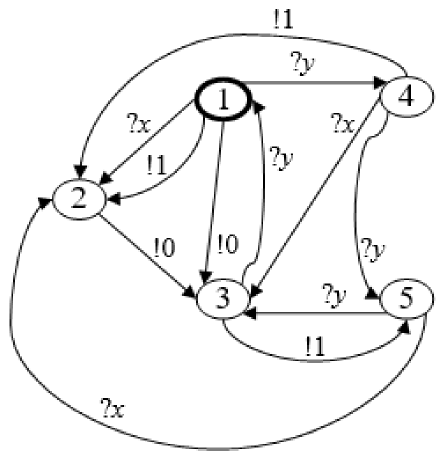

Fig. 4a. Automaton ***Q*** (a)

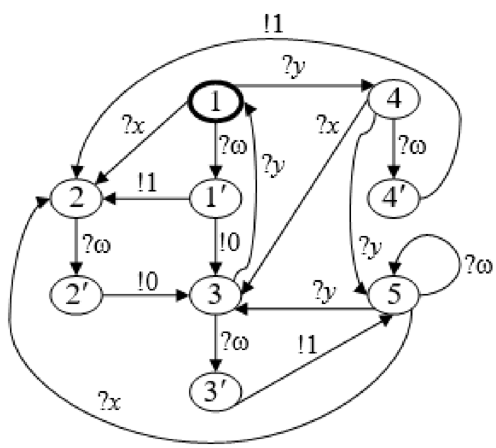

Fig. 4b. Automaton $\boldsymbol{Q}^{\omega}$

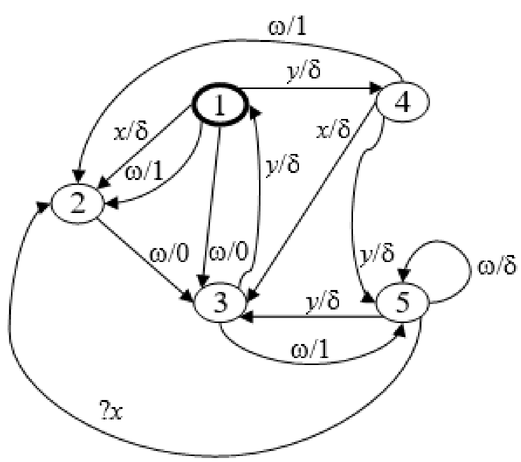

Fig. 4c. FSM M^{ω}_{Q}

Automata ***S*** and ***P*** with mixed states are *separable* if automata $\boldsymbol{S}^{\omega}$ and $\boldsymbol{P}^{\omega}$ are separable. If automata $\boldsymbol{S}^{\omega}$ and $\boldsymbol{P}^{\omega}$ are separable then a *separating sequence* for these automata is a separating sequence for automata ***S*** and ***P***. However, the *hypothesis for applying input sequences* becomes more complex. An automaton under experiment waits for an input until the input timeout T_{in} expires and thus, a tester has to apply inputs fast enough, and a timer for calculating T_{in} advances from zero after applying a current input or receiving an output different from δ. After producing δ *under input* ω, in general, the next input can be applied at any time instance but for having unique requirements for applying inputs we assume that the output timeout T_{out} has to be over. When a separating sequence has an input ω, this means that no input is applied to an automaton under experiment: we just wait for an output to be produced during the output timeout T_{out} and only after an output is produced or T_{out} expires the next input can be applied (in the T_{in} range).

As an example, consider how a sequence *yx*ωω is applied to automaton ***Q*** (Fig. 3a). An input *y* is applied and after this, the next input *x* is applied in the T_{in} range; thus, the automaton reaches state 3. After this an output is expected until the timeout T_{out} expires. In our example, the output !1 is produced. We wait another timeout T_{out} and obtain δ as an output. The experiment is over, since there are no more inputs in the input sequence. We cannot apply an input sequence *xy* at the initial state of ***Q*** since after accepting input *x* the automaton reaches state 2 where a transition under *y* is not defined. By construction of $\boldsymbol{S}^{\omega}$, the following statement holds.

Theorem 2. Given a trace σ^{ω} of $\boldsymbol{S}^{\omega\delta}$, a trace of $\boldsymbol{S}^{\delta}$ is obtained after deleting ω from the trace σ^{ω}, and vice versa, given a trace σ of $\boldsymbol{S}^{\delta}$, if ω is added in front of each output (except δ) and any number of ω in front of δ and after δ, then the obtained sequence is a trace of $\boldsymbol{S}^{\omega\delta}$.

Automata $\boldsymbol{S}^{\omega}$ and $\boldsymbol{P}^{\omega}$ have no mixed states and in order to check their separability and derive a separating sequence (if the automata are separable), FSMs M^{ω}_{S} and M^{ω}_{P} for automata $\boldsymbol{S}^{\omega,\delta}$ и $\boldsymbol{P}^{\omega,\delta}$ are derived. If FSMs M^{ω}_{S} and M^{ω}_{P} are separable then a separating sequence for these FSMs is a separating sequence for automata ***S*** and ***P***.

Indeed, if α is a separating sequence for automata ***S*** and ***P*** then α is a separating sequence for FSMs M^{ω}_{S} и M^{ω}_{P}. If α is a shortest separating sequence for FSMs M^{ω}_{S} и M^{ω}_{P}, then α = βω…ω and any proper prefix of α is not a separating sequence. Therefore, β takes both automata to states where the sets of output to ω…ω do not intersect, and thus, having a response to ω…ω we can conclude which automaton ***S*** or ***P*** is under experiment.

Given an automaton ***S*** in Figure 1a, construct a corresponding automaton $\boldsymbol{S}^{\omega}$ and derive a corresponding FSM M^{ω}_{Q} (Figure 4c) for the automaton ***Q*** in Figure 4a. FSMs M^{ω}_{Q} и M^{ω}_{S} are separated by an input sequence ω, since ***S*** after waiting for an output during T_{out} does not produce any output while ***Q*** produces an output !0.

V. Conclusions

In this paper, we have proposed a technique for separating Input/Output automata without a nonobservable action and in fact, this paper is the extension of [13] where separating sequences are derived for I/O automata without mixed states, i.e., states where transitions both under inputs and under outputs are defined and there are no cycles labeled by outputs. In this paper, we extend the set of considered automata modifying the discipline of applying input sequences and the next step is to extend the obtained results for checking the existence and derivation of an adaptive separating sequence for Input/Output automata as well as to study other kinds of state identification sequences such as homing and synchronizing sequences.

Acknowledgment

This work is partly supported by RFBR project N 19-07-00327/19.

References

[1] Kam, T., Villa, T., Brayton, K. R., Sangiovanni-Vincentelli, A. Synthesis of FSMs: Functional Optimization. Springer, 1997. 282 p.

[2] Tretmans J. A formal approach to conformance testing // The Intern. Workshop on Protocol Test Systems. 1993. P. 257–276.

[3] Burdonov I.B., Kossachev A.S., Kuliamin V.V. Conformance theory for systems with blocking and destruction. Nauka, 2008. 412 p.

[4] Iksoon Hwang, Nina Yevtushenko, Ana R. Cavalli: Tight bound on the length of distinguishing sequences for non-observable nondeterministic Finite-State Machines with a polynomial number of inputs and outputs. Inf. Process. Lett. 112(7). 2012. P. 298-301.

[5] Igor Burdonov, Alexandr Kossachev, Nina Yevtushenko, Alexey Demakovn. "Evaluating the Length of Distinguishing Sequences for Non-Deterministic Input/Output Automata", Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS), pp. 445-449

[6] Starke P. Abstract Automata. American Elsevier, 1972. 419 p.

[7] I. Burdonov, N. Yevtushenko, A. Kossachev. Distingusihing transitions systems with nondeterministic behavior. In Proc. Of Russian conference «Scientific Service in Internet», 2019, P. 177-187 (in Russian).

[8] Hopcroft J.E., Motwani, R., and Ullman J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, second edition. 2001.

[9] Kushik N., Yevtushenko N., Burdonov I., Kossachev A. Synchronizing and Homing Experiments for Input/output Automata // System Informatics. 2017. No. 10. P. 1–10.

[10] Kushik N., Yevtushenko N., Cavalli A.R. On Testing against partial nondeterministic machines // Intern. Conf. on the Quality of information and Communications Technology. 2014. P. 230–233.

[11] Petrenko A., Yevtushenko N. Conformance Tests as Checking Experiments for Partial Nondeterministic FSM // Lecture Notes in Computer Science. 2005. V. 3997. P. 118–133.

[12] Yevtushenko N., Kushik N. Some state identification problems for nondeterministic Finite State Machines. CTT, 2018. 190 p. (in Russian)

[13] Burdonov I., Yevtushenko N., Kossachev A. Separating Input/Output automata with nondeterministic behavior // Russian Digital Libraries Journal. 2020. T. 23, Vol. 2 (in Russian).