UDK 519.713

Synchronizing and Homing Experiments for Input/output Automata

Kushik N.G. (SAMOVAR, CNRS, Télécom SudParis / Université Paris-Saclay), Yevtushenko N.V. (Tomsk State University, Institute for System Programming of the Russian Academy of Sciences)

Burdonov I.B. (Institute for System Programming of the Russian Academy of Sciences)

Kossatchev A.S. (Institute for System Programming of the Russian Academy of Sciences)

The paper is devoted to studying the (‘gedanken’) experiments with input/output automata. We propose how to derive proper input sequences for identifying the final (current) state of the machine under experiment, namely synchronizing and homing sequences. The machine is non-initialized and its alphabet of actions is divided into disjoint sets of inputs and outputs. In this paper, we consider a specific class of such machines for which at each state the transitions only under inputs or under outputs are defined, and the machine transition diagram does not contain cycles labeled by outputs, i.e. the language of the machine does not contain traces with infinite postfix of outputs. Moreover, for each state where the transitions under inputs are defined, the machine has a loop under a special quiescence output. For such class of input/output automata, we define the preset synchronizing and homing experiments, establish necessary and sufficient conditions for their existence and propose techniques for their derivation. The procedures for deriving the corresponding (‘gedanken’) experiments for input/output automata are based on the well-studied solutions to these problems for Finite State Machines.

Keywords: *Input/Output Automata, Synchronizing Sequence, Homing Sequence*

1. Introduction

The state identification problem using ‘gedanken’ experiments with Finite State Machines (FSMs) is a long standing problem. The first results were obtained by Moore [10] and have been then improved by many researchers [3, 6, 9]. A (‘gedanken’) state identification experiment with an FSM consists of applying an input sequence to a machine under investigation, observing the output response and drawing a conclusion about initial or current state. If the conclusion is drawn about the initial state (a state before the experiment) then the experiment is called *distinguishing*. When the conclusion is drawn about the current FSM state (a state after the experiment) then the experiment is called *homing* or *synchronizing*. Depending on the way how an input sequence is applied an experiment can be *preset* or *adaptive*. In this paper, we discuss only preset experiments when an applied input sequence is derived in advance.

There are many applications of such 'gedanken' experiments and a big body of work is developed for constructing preset and adaptive experiments [4, 7, 11]. Most applications are related to decreasing the complexity of deriving a test suite with the guaranteed fault coverage when the specification FSM has homing/synchronizing/distinguishing sequences and there are many papers how such sequences can be derived for deterministic and nondeterministic, complete and partial FSMs [2, 5, 7, 11]. In [8], the authors propose how homing and synchronizing sequences can accelerate/optimize the monitoring of communicating systems. When the initial/current state of an Implementation Under Test (IUT) is known, the set of properties that should be verified at a given IUT state can be dramatically reduced.

However, FSMs have limited capacity when describing software component behavior. The reason is that the next input can be applied only when the FSM under investigation produced an output to the previous input. On one hand, this allows to escape races between inputs and outputs and it is one of the reasons why test suites with the guaranteed test coverage are derived mostly against FSMs. On the other hand, FSM notion does not allow to consider the situations when an output can be produced only after a sequence of inputs has been applied to an IUT and moreover, not a single output can be produced but a sequence of outputs. Such situations can be described when using Input/Output automaton as a model; an Input/Output automaton has the finite number of states but differently from FSMs, transitions between states are labeled not by a pair <input, output> but by a single input or output. To the best of our knowledge there are no investigations on homing/synchronizing sequences for such model.

In this paper, we study the state identification problem for Input/Output automata when at each state, only inputs or only outputs are allowed. We define the notions of homing/synchronizing sequences for such Input/Output automata and adapt the known techniques for deriving such sequences for a new model. Therefore, the main contribution of this paper is the definition of homing/synchronizing sequences for Input/Output automata and the development of techniques for the existence check and derivation.

The rest of the paper is structured as follows. Section 2 contains preliminaries. Techniques for deriving homing and synchronizing sequences for input/output automata are proposed in Section 3. Section 4 concludes the paper and has a brief discussion on the directions of the future work.

We note that this work is partially supported by the Russian Science Foundation (RSF), project № 16-49-03012.

2. Preliminaries

An Input/Output *Automaton* (or an *automaton* in this paper) is a 4-tuple $\boldsymbol{S} = (S, I, O, T_S)$ where S is a finite set of states; I and O are finite non-empty disjoint sets of inputs and outputs, respectively; $T_S \subseteq S \times I \times S \cup S \times O \times S$ is a *transition relation* where 3-tuples $(s, i, s') \in T_S$ and $(s, o, s') \in T_S$ are *transitions*.

In this paper, we consider a specific class of automata for which the following holds:

i) At each state only inputs or only outputs are allowed, i.e. $S = S_1 \cup S_2$, $S_1 \cap S_2 = \varnothing$ and $T_S \subseteq S_1 \times I \times S \cup S_2 \times O \times S$;

ii) The transition diagram does not contain cycles/loops labeled with outputs, i.e. the language of the machine does not contain traces with infinite postfix of outputs;

iii) The machine has a special output $\delta \notin O$ that represents the quiescence [12] at the states where the transitions under inputs are defined; at each state $s \in S_1$, there is a loop under δ, namely $(s, \delta, s) \in T_S$.

As an example of an Input/Output automaton, consider a machine in Fig. 1. The automaton $\boldsymbol{S}$ has five states, namely $S = \{s_1, \ldots, s_5\}$, where $S_1 = \{s_1, s_2, s_5\}$ and $S_2 = \{s_3, s_4\}$. At each state from the set S_1 the automaton accepts inputs i_1 and i_2. However, when the machine is at state s_3 or s_4 no inputs can be accepted and only outputs o_1 or o_2 can be produced.

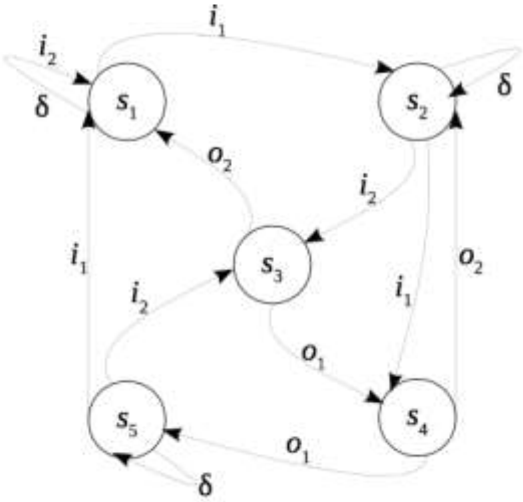

Fig. 1 – An input/output automaton $\boldsymbol{S}$

As usual, synchronizing and homing experiments are used to identify the final (current) state of the machine under experiment, i.e., the state reached by the machine after an appropriate input sequence has been applied when the initial state of the machine was unknown. In this paper, we adapt the notion of a ‘gedanken’ experiment in the way that it can be used for the input/output automata for which corresponding synchronizing and homing sequences can be defined.

The experiment is performed under the following **hypothesis**:

We assume that before applying any input, a tester (or any experimenting entity) waits for a given maximal output timeout t. The experiment is performed as follows: the tester expects an output in t time units; if the machine produces one, then the timer is reset and the tester waits for another t time units. If no output is produced by the system in t time units then the tester applies the next input (if any) and resets the timer.

The latter explains the necessity of introducing the specific output $\delta \notin O$, namely whenever the output is not observed we assume that the system/machine produced the output δ. Such extension of the output alphabet allows to define the corresponding synchronizing and homing sequences for an Input/Output automaton.

As usual, a synchronizing sequence is an input trace such that after its application independent of the initial state, the current state of the machine is known. In other words, a sequence $\alpha = i_1 i_2 \ldots i_k$ is *synchronizing* for the automaton $\boldsymbol{S}$ if there exists a state $s \in S$ such that for each trace $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$ where p is the length of a longest sequence of consecutive outputs and $\beta_j \in (O \cup \{\delta\})^p$, $j = 1,\ldots, k + 1$, it holds that the $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$-successor of the set S ($\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$-state-after-$S$) is either empty or equals $\{s\}$. We note that hereafter the γ-successor of the state $s \in S$ is the set of states that can be reached from state s through the trace γ while the γ-successor of S has every state that is reached from some state of S through the trace γ.

A homing sequence allows to determine the final (current) state of the machine under experiment via the observation of its output response. Therefore, a sequence $\alpha = i_1 i_2 \ldots i_k$ is *homing* for the automaton S if for each trace $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$, $\beta_j \in (O \cup \{\delta\})^p$, $j = 1,\ldots, k + 1$, it holds that the $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$-successor of the set S is either empty or is a singleton.

For an automaton $\boldsymbol{S}$ in Fig. 1 a homing sequence α is $\alpha = i_1 i_1$.

3. Deriving synchronizing and homing sequences for input/output automata

In this section, we discuss how homing and synchronizing sequences defined above can be derived against input/output automata. We also establish necessary and sufficient conditions for the existence of such sequences for the machines of the class/type described above.

3.1. Deriving synchronizing experiments

We propose to derive a synchronizing sequence for an automaton $\boldsymbol{S}$ where actions are divided into inputs and outputs via an iterative elimination of the transitions labeled by outputs. Such transition can always be omitted as for the automata class considered in this paper, there does not exist a state where transitions under inputs and outputs are defined at the same time. In other words, we propose to derive an automaton where only the transitions under inputs are left. Synchronizing sequences for such kind of automata are well studied [2, 5, 11, 13] and thus, classical methods for their derivation can be further applied.

Procedure 1

Input: Input/Output automaton $\boldsymbol{S} = (S, I, O, T_S)$

Output: Synchronizing sequence α or a message “The automaton $\boldsymbol{S}$ is not synchronizing”

Step 1. Derive an automaton $\boldsymbol{A} = (S_1, I, T_A)$ with the empty set of transitions, i.e. $T_A = \varnothing$.

Step 2. For each transition $(s, i, s'') \in T_S$, where $s, s'' \in S_1$, add to T_A the transition (s, i, s''); for each transition (s, i, s''), where $s \in S_1$ and $s'' \in S_2$, add to T_A the transition (s, i, s''') where state $s''' \in S_1$ and s''' is in a β-successor of s'' in the automaton $\boldsymbol{S}$, $\beta \in O^*$.

Step 3. Check the existence and derive, if possible, a synchronizing sequence α for the automaton $\boldsymbol{A}$:

If the sequence α is derived then **Return** α;

Else Return the message “The automaton $\boldsymbol{S}$ is not synchronizing”

□

Proposition 1. The automaton $\boldsymbol{S}$ is synchronizing if and only if the automaton $\boldsymbol{A}$ in Procedure 1 is synchronizing. Moreover, each synchronizing sequence α for $\boldsymbol{A}$ is a synchronizing sequence for $\boldsymbol{S}$.

□

As an example, check the existence of a synchronizing sequence for an automaton in Fig. 1. The corresponding automaton ***A*** derived over the inputs of ***S*** is presented in Fig. 2.

By direct inspection, one can assure that the automaton ***A*** in Fig. 2 is not synchronizing. Therefore, due to Proposition 1, the automaton ***S*** does not have a synchronizing sequence.

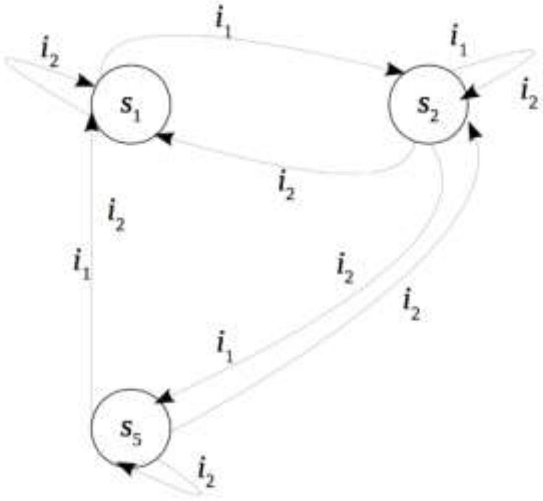

Fig. 2 – An automaton ***A*** derived through the application of Procedure 1

3.2. Deriving homing experiments

Similar to the derivation of synchronizing sequences we propose to reduce the problem of checking the existence and derivation of a homing sequence for an input/output automaton to that one for classical Finite State Machines (FSMs) as there are a number of such techniques for FSMs [see, for example 6-8].

Procedure 2

Input: Input/Output automaton ***S*** = (S, I, O, T_S)

Output: Homing sequence α or the message “The automaton ***S*** is not homing”

Step 1. Derive an FSM ***M*** = $(S_1, I, O \cup O^2 \cup$□… $\cup\ O^p \cup$□$\{\delta\}, T_M)$ with the empty set of transitions, i.e., $T_M = \varnothing$, where p is the length of a longest output trace of the automaton ***S***.

Step 2. For each state $s \in S_1$, such that $(s, i, s') \in T_S$, $s' \in S_1$, add to the T_M the transition (s, i, δ, s').

Step 3. For each state $s \in S_1$, such that $(s, i, s') \in T_S$, $s' \in S_2$, add to the T_M the transition $(s, i, o_1 o_2 \ldots o_k, s'')$, $k \le p$, where $s'' \in S_1$ is the $o_1 o_2 \ldots o_k$-successor of state s'.

Step 4. Check the existence and derive, if possible, a homing sequence α for the FSM ***M***:

If the sequence α is derived then **Return** α;

Else Return the message "The automaton ***S*** is not homing"

□

Proposition 2. A sequence α is homing for the automaton ***S*** if and only if α is a homing sequence for the FSM ***M***.

□

As an example, check the existence of a homing sequence of an automaton ***S*** in Fig. 1. This automaton does not have a synchronizing sequence, nevertheless, a homing one can still exist. In order to check the existence of a homing sequence we derive an FSM ***M*** of Procedure 2 for the automaton ***S*** in Fig. 1. The obtained FSM ***M*** is shown in Fig. 3.

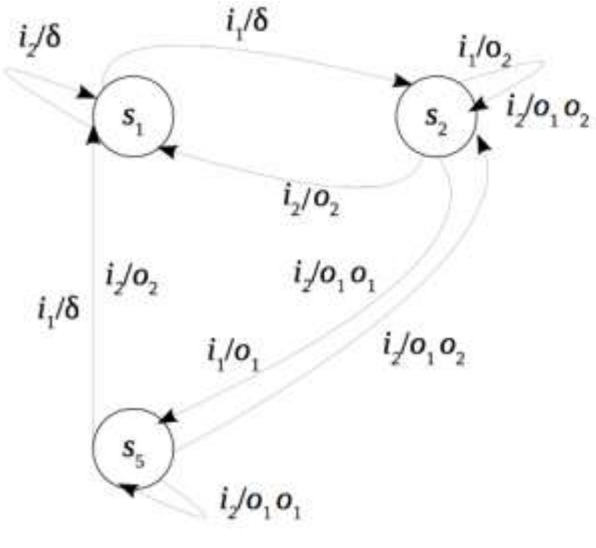

Fig. 3 – An FSM ***M*** derived from ***S*** using Procedure 2

The application of classical methods [6, 7, 11] for the homing sequence derivation for the FSM ***M*** can return a sequence $\alpha = i_1i_1$. Therefore, the sequence $\alpha = i_1i_1$ is a homing sequence for the automaton ***S*** in Fig. 1.

We note that the computational complexity of the existence check is the same (or at least not better than) as for corresponding automata and FSMs. The reason is that when deriving a synchronizing sequence an automaton that does not have output actions can be considered while for deriving a homing sequence a classical FSM can be represented as an Input/Output automaton by presenting each FSM transition *i*/*o* as a pair of consecutive transitions under *i* and *o*. The latter means that for input/output automata that lead to the derivation of synchronizing/homing sequences for (partial) nondeterministic automata/FSMs the length of such sequences is exponential with respect to the number of states of the machine. Therefore, one of interesting issues for the future work is the development of the approaches for decreasing this complexity or to specify classes of Input/Output automata where the complexity can be reduced.

4. Conclusion

In this paper, we have studied the problem of deriving homing/synchronizing sequences for Input/Output automata. We have limited the class of such automata with automata where at each state, only inputs or only outputs are defined. We have shown how for such automata, the known techniques for deriving homing/synchronizing sequences for automata and FSMs can be adapted. To the best of our knowledge there are no papers where homing/synchronizing sequences are derived for Input/Output automata. We also mention that a number of examples of using Input/Output automata for describing the behavior of (components) of discrete event systems can be found in [1, 12].

As for the future work, we are mostly concerned how to extend the obtained results to adaptive sequences including distinguishing sequences as well as how to expand the class of Input/Output automata for such ‘gedanken’ experiments.

An interesting question is about Input/Output automata with a nonobservable action τ. In this case, the nondeterminism degree of the automaton will be increased and more assumptions on the automaton behavior should be made when deriving homing/synchronizing sequences.

Another question is about considering Input/Output automata where both inputs and outputs are specified at some state. It is quite possible that in this case, more assumptions have to be made about the implementation/execution of a ‘gedanken’ experiment and we are going to try our hand in establishing such minimum assumptions.

As most problems of checking the existence and derivation of synchronizing and homing sequences for partial and non-deterministic automata and FSMs are PSPACE-complete, there is an interesting question about defining Input/Output automata classes where the complexity can be decreased. One possible way to decrease this complexity can be to consider adaptive synchronizing and homing experiments instead of preset, however this issue needs more investigation.

Based on the experience of constructing adaptive state identification experiments for FSMs, we suppose that in this case, we will face the same problems as discussed above and it is our first priority to define and construct adaptive homing/synchronizing experiments at least for the class of Input/Output automata described in this paper.

The problems listed above, as well as many others, form the directions for the future work.

References

1. Бурдонов И.Б. Теория конформности для функционального тестирования программных систем на основе формальных моделей : дис. … док. физ.-мат. наук. Москва, 2008. 596 с.
2. Мартюгин П.В. Нижние оценки длины кратчайших бережно синхронизирующих слов для двух- и трёхбуквенных частичных автоматов // Дискретн. анализ и исслед. опер. 2008. №4. Т. 15. С. 44-56.
3. Gill A. State-identification experiments in finite automata // Information and Control. 1961. P. 132-154.
4. Hierons R. M., Jourdan G.-V., Ural H., Yenigun H. Using adaptive distinguishing sequences in checking sequence constructions // ACM symposium on Applied computing : proceedings. 2008. P. 682-687.
5. Ito M., Shikishima-Tsuji K. Some Results on Directable Automata // Theory Is Forever. LNCS. 2004. №3113. P. 125-133.
6. Kohavi Z. Switching and Finite Automata Theory. McGraw-Hill: New York, 1978.
7. Kushik N., El-Fakih K., Yevtushenko N., Cavalli A. R. On Adaptive Experiments for Nondeterministic Finite State Machines // Software Tools for Technology Transfer. 2016. 18 (3). P. 251-264.
8. Kushik N., López J., Cavalli A.R., Yevtushenko N. Improving Protocol Passive Testing through "Gedanken" Experiments with Finite State Machines // QRS : proceedings. 2016. P. 315-322.
9. Lee D., Yannakakis M. Testing finite-state machines: state identification and verification // IEEE Trans. on Computers. 1994. 43(3). P. 306-320.
10. Moore E.F. Gedanken-experiments on sequential machines // In Automata Studies (Annals of Mathematical Studies no.1). Princeton University Press. 1956. P. 129-153.
11. Sandberg S. Homing and Synchronization Sequences // Model Based Testing of Reactive Systems. LNCS. 2005. №3472. P. 5-33.
12. Tretmans J. Test Generation with Inputs, Outputs and Repetitive Quiescence // Software - Concepts and Tools. 1996. 17 (3). P. 103-120.
13. Volkov M. Synchronizing Automata and the Černý Conjecture // 2nd Int'l. Conf. Language and Automata Theory and Applications : proceedings. 2008. P. 11-27.