

# Reducing the Complexity of Checking the Existence and Derivation of Adaptive Synchronizing Experiments for Nondeterministic FSMs

Natalia Kushik<sup>1,2</sup>, Nina Yevtushenko<sup>2</sup> and Husnu Yenigun<sup>3</sup>

<sup>1</sup>Telecom SudParis, Evry, France

<sup>2</sup>Tomsk State University, Tomsk, Russia

<sup>3</sup>Sabanci University, Istanbul, Turkey

Keywords: Nondeterministic Finite State Machines, Adaptive Sequences, Synchronizing Experiments.

Abstract: In this paper, we address the problem of setting a discrete system specified as a Finite State Machine (FSM) to a known initial state. As the system behavior can be nondeterministic, we discuss the complexity of problems related to deriving synchronizing sequences for nondeterministic FSMs and propose a way for decreasing such complexity. In fact, we suggest to use adaptive synchronizing sequences instead of preset ones, and prove that for non-initialized nondeterministic FSMs the problem of checking the existence of an adaptive synchronizing sequence is in P.

## 1 INTRODUCTION

As the complexity of hardware/software systems increases, new methods and techniques for efficient checking of their functional and non-functional requirements need to be developed. Such checking can be effectively performed when the formal description of a system under test is provided as a finite state model. Finite state models are widely used to derive tests with the guaranteed fault coverage for checking functional requirements for various types of digital systems (see, for example, Gill, 1961; Chow, 1978).

A Finite State Machine (FSM) has finite non-empty sets of states, inputs and outputs; when an input is applied, the FSM moves to the next state producing an output. In other words, FSMs include a ‘natural reactivity’ and that is the reason why they are widely used when modeling systems working in request-response mode. The behavior of such a machine is described as a set of available input/output sequences (traces). In general, the set of traces is infinite. However, under certain conditions, it is possible to identify a *finite* subset of traces such that the application of input sequences from this subset and the observation of the expected output responses from an implementation under test (IUT) allows one to conclude that the system meets the corresponding requirements. Such a finite set of

traces is used to form, as it is called in the literature, a *checking sequence* or a *test suite*.

Complex systems can usually have a nondeterministic behavior, i.e. there can be several output responses specified for an FSM at a given state when an input sequence is applied. Meanwhile, test generation techniques against nondeterministic FSMs have higher complexity (Petrenko, Yevtushenko, and Bochmann, 1996; Hierons, 1998) and they mostly remain non-applicable to ‘real-life’ digital systems. Moreover, most of the test derivation techniques for nondeterministic FSMs, including the derivation of a single checking sequence applied to an IUT rely on the fact that the initial state of an IUT is known. Nevertheless, it is not always the case and thus, effective methods and techniques for setting up an IUT to a known (current) state still remain one of the important research topics for machines that can have a nondeterministic behavior. Usually this ‘set up’ is made by an application of an input sequence such that after observing the output response (in which case the sequence is called a *homing sequence*) or without observing the output response (in which case the sequence is called a *synchronizing sequence*), one can conclude about the current state of an IUT. Homing and synchronizing sequences can either serve as preambles of checking sequences or they can be used to set up a system into a particular critical state. After such a set up, the

required input sequences can be applied for checking an IUT against appropriate (functional or non-functional) test purposes such as safety, security, robustness, etc. However, for nondeterministic FSMs the length of homing and synchronizing sequences can be exponential w.r.t. the number of states (Ito and Shikishima-Tsuji, 2004; Kushik and Yevtushenko, 2013). Correspondingly, additional research should be performed for efficient derivation of homing and synchronizing sequences for nondeterministic FSMs.

In order to simplify the derivation of homing and synchronizing sequences researchers turn their attention to so called *adaptive* sequences that in many cases can be shorter than the ordinary (*preset*) ones. A sequence is *adaptive* if the next input to be applied to an IUT is chosen based on the previously observed outputs, and a sequence is *preset* if the outputs need to be observed only after the application of the whole sequence, or need not to be observed at all.

Homing and synchronizing sequences are well studied for deterministic FSMs and deterministic automata in which outputs are omitted. Both types of sequences allow to determine the current state of the machine after the application of the input sequence. In the case of a homing sequence, the conclusion about the current FSM state is made based on the observed output response, whereas for a synchronizing sequence, the final state is unique independently of the initial state of the FSM and the observed output sequence.

For deterministic FSMs as well as for deterministic automata the length of both homing and synchronizing sequences is polynomial (Klyachko, Rystov and Spivak, 1987; Cerny, 1964; Hibbard, 1961) when at each state a transition under each input is defined. Moreover, it has been shown that for deterministic FSMs the length of a homing sequence cannot be decreased when considering adaptive sequences instead of preset (Hibbard, 1961). For nondeterministic machines, preset homing and synchronizing sequences are known to have exponential length (Ito and Shikishima-Tsuji, 2004; Kushik and Yevtushenko, 2013). Nevertheless, it has been also shown that the length of a homing sequence can be decreased up to polynomial when considering complete nondeterministic FSMs (Kushik et al., 2014).

The authors are not aware of any results regarding the existence check or the derivation of adaptive synchronizing sequences for nondeterministic FSMs, except for an idea presented in (Kushik and Yevtushenko, 2012). In this paper, a

method is presented for checking the existence and for deriving such adaptive synchronizing sequences for a nondeterministic FSM that can have an arbitrary number of initial states. In this case, an adaptive synchronizing sequence is represented as a corresponding single-input output-complete acyclic FSM, introduced in (Petrenko and Yevtushenko, 2005) and hereafter referred to as a *Test Case*.

In this paper, necessary and sufficient conditions for the existence of a synchronizing test case are established. The obtained criterion allows to assess the complexity of the corresponding procedure as well as to determine the maximal length (height) of a corresponding synchronizing test (if it exists). As this bound is exponential, we show how this complexity can be decreased and show that for non-initialized FSMs the problem of checking the existence of an adaptive synchronizing sequence is in P. Furthermore, we propose an algorithm for deriving a synchronizing test case for a nondeterministic non-initialized FSM with the polynomial length. The results listed above form the main contribution of the paper.

The rest of the paper is organized as follows. Section 2 contains preliminaries. The necessary and sufficient conditions for the existence of an adaptive synchronizing test case for a weakly initialized nondeterministic FSM are established in Section 3. The complexity of the corresponding procedure is given in the same section. As this complexity is shown to be exponential, a novel method is proposed in Section 4 for nondeterministic non-initialized FSMs for which the problem of checking the existence of a synchronizing test case can be solved in polynomial time. Section 5 concludes the paper.

## 2 PRELIMINARIES

A *weakly initialized Finite State Machine* (FSM)  $\mathbf{S}$  is a 5-tuple  $(S, I, O, h_S, S_{in})$ , where  $S$  is a finite set of states with the set  $S_{in} \subseteq S$  of initial states;  $I$  and  $O$  are finite non-empty disjoint sets of inputs and outputs, respectively;  $h_S \subseteq S \times I \times O \times S$  is a *transition relation*, where a 4-tuple  $(s, i, o, s') \in h_S$  is a *transition*. If  $|S_{in}| = 1$  then the FSM  $\mathbf{S}$  is an *initialized* FSM. If  $S_{in} = S$  the machine is called a *non-initialized* machine. An input  $i$  is *defined* at a state  $s$  if there exists a transition  $(s, i, o, s') \in h_S$ . FSM  $\mathbf{S} = (S, I, O, h_S, S_{in})$  is *complete* if for each pair  $(s, i) \in S \times I$  there exists a pair  $(o, s') \in O \times S$  such that  $(s, i, o, s') \in h_S$ ; otherwise, the machine is *partial*. FSM  $\mathbf{S}$  is *nondeterministic* if for some pair

$(s, i) \in S \times I$ , there exist at least two transitions  $(s, i, o_1, s_1), (s, i, o_2, s_2) \in h_S$ , such that  $o_1 \neq o_2$  and/or  $s_1 \neq s_2$ . FSM  $\mathbf{S}$  is *observable* if for each two transitions  $(s, i, o, s_1), (s, i, o, s_2) \in h_S$  it holds that  $s_1 = s_2$ . Note that in this paper, we consider complete observable nondeterministic FSMs if the contrary is not explicitly stated.

FSM  $\mathbf{S}$  is *single-input* if at each state there is at most one defined input at the state, i.e., for each two transitions  $(s, i_1, o_1, s_1), (s, i_2, o_2, s_2) \in h_S$  it holds that  $i_1 = i_2$ , and FSM  $\mathbf{S}$  is *output-complete* if for each pair  $(s, i) \in S \times I$  such that the input  $i$  is defined at state  $s$ , there exists a transition from  $s$  under  $i$  for every output in  $O$  (Petrenko and Yevtushenko, 2005).

A *trace* of  $\mathbf{S}$  at state  $s$  is a sequence of input/output pairs of consecutive transitions starting from state  $s$ . Given a trace  $i_1 o_1 \dots i_k o_k$  at state  $s$ , the input projection  $i_1 \dots i_k$  of the trace is a *defined* input sequence at state  $s$ . For an observable nondeterministic FSM, if  $\gamma = i_1 o_1 \dots i_k o_k$  is a trace at a state  $s$ , then there exists a unique sequence of consecutive transitions  $(s, i_1, o_1, s_1)(s_1, i_2, o_2, s_2) \dots (s_{k-1}, i_k, o_k, s_k)$ . As usual, for state  $s$  and a sequence  $\gamma \in (IO)^*$  of input/output pairs, the  $\gamma$ -*successor* of state  $s$  is the set of all states that are reached from  $s$  by trace  $\gamma$ . If  $\gamma$  is not a trace at state  $s$  then the  $\gamma$ -successor of state  $s$  is empty or we simply say that the  $\gamma$ -successor of state  $s$  does not exist. For an observable FSM  $\mathbf{S}$ , for any string  $\gamma \in (IO)^*$ , the cardinality of the  $\gamma$ -successor of state  $s$  is at most one. Given a subset  $S'$  of states, the  $\gamma$ -*successor* of  $S'$  is the union of  $\gamma$ -successors over all states of the set  $S'$ .

Given an input alphabet  $I$  and an output alphabet  $O$ , a *test case*  $TC(I, O)$  is an initially connected single-input output-complete observable initialized FSM  $\mathbf{P} = (P, I, O, h_P, \{p_0\})$  with an acyclic transition graph. Given a complete FSM  $\mathbf{S}$  over alphabets  $I$  and  $O$ , a test case  $TC(I, O)$  represents an adaptive experiment with the FSM  $\mathbf{S}$  (Kushik et al., 2014).

If  $|I| > 1$  then a test case is a partial FSM. A state  $p \in P$  is a *deadlock* state of the FSM  $\mathbf{P}$  if there are no defined inputs at this state. In general, given a test case  $\mathbf{P}$ , the *length (height)* of the test case  $\mathbf{P}$  is defined as the length of a longest trace from the initial state to a deadlock state of  $\mathbf{P}$  and it specifies the length of the longest input sequence that can be applied to an FSM  $\mathbf{S}$  during the experiment. As usual, for complexity reasons, one is interested in deriving a test case with minimal length.

A test case  $\mathbf{P}$  is a *homing* test case for an FSM  $\mathbf{S}$  if for every trace  $\gamma$  of  $\mathbf{P}$  from the initial state to a

deadlock state, the  $\gamma$ -successor of  $S_{in}$  has at most one state. A *homing test case* is a *synchronizing test case* for an FSM  $\mathbf{S}$ , if there exists a state  $s$  such that for every trace  $\gamma$  of  $\mathbf{P}$  from the initial state to a deadlock state,  $\gamma$ -successor of  $S_{in}$  is either  $\{s\}$  or the empty set. If there exists a homing/synchronizing test case for the FSM  $\mathbf{S}$  then the set  $S_{in}$  is a *homing/synchronizing* set and the test case  $\mathbf{P}$  is a *homing/synchronizing test case* for the set  $S_{in}$ . Otherwise, the set  $S_{in}$  is not homing/synchronizing.

We further discuss how to check whether there exists a synchronizing test case for a given complete observable nondeterministic FSM in polynomial time. When there exists such a synchronizing test case, we propose a technique to derive one. We also provide the upper bound on the length of a synchronizing test case. As this technique relies on the presence of so called definitely-reachable state (Petrenko and Yevtushenko, 2011) in the equivalent initialized FSM, we repeat this notion here and briefly sketch the procedure to check the existence of such state in a nondeterministic FSM.

Given a complete weakly initialized FSM  $\mathbf{S} = (S, I, O, h_S, S_{in})$ , state  $s' \in S$  is *definitely-reachable (d-reachable)* from state  $s \in S$  if there exists a test case  $\mathbf{P}(s, s')$  over alphabets  $I$  and  $O$  such that for every trace  $\gamma$  of  $\mathbf{P}(s, s')$  from the initial state to a deadlock state, the  $\gamma$ -successor of state  $s$  is either the empty set or is the set  $\{s'\}$ . We hereafter refer to such a test case as a *d-transfer* test case.

If the machine is initialized and state  $s' \in S$  is definitely reachable from the initial state then we simply say that state  $s' \in S$  is *definitely-reachable*.

In (Petrenko and Yevtushenko, 2011), necessary and sufficient conditions are established that allow to check if state  $s' \in S$  is definitely reachable from state  $s \in S$ . In particular, it is proven that state  $s' \in S$  is definitely reachable from state  $s$  if and only if  $\mathbf{S}$  has a single-input acyclic submachine  $\mathbf{S}'$  with the initial state  $s$  and the only deadlock state  $s'$  such that for each input defined in some state of  $\mathbf{S}'$ , the state has all the transitions of  $\mathbf{S}$  labeled with this input. Moreover, in the same paper, an efficient method is proposed for checking whether state  $s'$  is definitely reachable from a state  $s$ , and if it is, then it is proposed how to derive a corresponding test case (see Procedure 1 given below).

Note that since any d-transfer test case  $\mathbf{P}(s, s')$  is an acyclic submachine of the machine  $\mathbf{S}$ , then the length of any trace in  $\mathbf{P}(s, s')$  does not exceed the number  $n$  of states of  $\mathbf{S}$ ; in other words, one needs at most  $n - 1$  inputs to adaptively transfer the possibly nondeterministic machine from state  $s$  to state  $s'$ .

Therefore, the length of a longest trace in a shortest test case  $\mathbf{P}(s, s')$  is polynomial and is at most  $n - 1$ .

**Procedure 1** for constructing a d-transfer test case  $\mathbf{P}(s, s')$

**Input:** An FSM  $\mathbf{S} = (S, I, O, h_S, S_{in})$ , states  $s, s' \in S$ ,  $s' \neq s$ .

**Output:** a d-transfer test case  $\mathbf{P}(s, s')$  if the state  $s'$  is definitely reachable from  $s$  or a message “the state  $s'$  is not definitely reachable from  $s$ ”.

Construct an FSM  $\mathbf{P}(s, s') = (P, I, O, h_P, \{p_0\})$  as follows

**Step 1**  $P := \{s'\}; h_P := \emptyset;$

**Step 2**

**While** there exist a state  $\hat{s} \notin P$  and a set of inputs  $I_{\hat{s}}$ , such that for each input  $i \in I_{\hat{s}}$ ,  $(\hat{s}, i, o, s'') \in h_S$ ,  $s'' \in P$  for every possible output  $o$  to an input  $i$  at state  $\hat{s}$  in FSM  $\mathbf{S}$

$P := P \cup \{\hat{s}\}; h_P := h_P \cup \{(\hat{s}, i, o, s'')\}$

**EndWhile**

**If**  $s \notin P$

**Then return** the message “the state  $s'$  is not definitely reachable from  $s$ ”.

**Step 3**  $p_0 := s;$

In the FSM  $\mathbf{P}(s, s')$ , remove from each state with several defined inputs all outgoing transitions with the same input until each such state has a single defined input.

Delete states which are unreachable from the initial state, add transitions to the designated deadlock state  $D$  for each missing output at each state (if any).

**Return**  $\mathbf{P}(s, s')$ . □

### 3 DERIVING SYNCHRONIZING TEST CASES FOR NONDETERMINISTIC FSMs

In this section, we discuss how a synchronizing test case can be derived for a complete observable nondeterministic FSM. In fact, the corresponding procedure is based on necessary and sufficient conditions for the existence of such test case that are stated in Proposition 1. The conditions rely on checking the existence of definitely-reachable states in the corresponding initialized machine. Such an equivalent initialized machine  $\mathbf{E}_S$  can be obtained through the determinization of the underlying

automaton. The set of traces of  $\mathbf{E}_S$  is the union of sets of traces over all states of the set  $S_{in}$ .

**Proposition 1.** There exists a synchronizing test case for a complete observable FSM  $\mathbf{S} = (S, I, O, h_S, S_{in})$ , if and only if the FSM  $\mathbf{E}_S$  has a definitely-reachable state  $\{s\}$  for some  $s \in S$ .

**Proof.** There is a trace  $\gamma$  at the initial state of the equivalent initialized machine  $\mathbf{E}_S$  if and only if such a trace is a trace at some initial state of FSM  $\mathbf{S}$ . Correspondingly, the  $\gamma$ -successor of the initial state of  $\mathbf{E}_S$  is the  $\gamma$ -successor of the set  $S_{in}$  in FSM  $\mathbf{S}$ . If the machine  $\mathbf{E}_S$  has a definitely-reachable state  $\{s\}$  then each trace of the FSM  $\mathbf{P}(e_0, \{s\})$  from the initial state to a deadlock state where  $e_0 = S_{in}$  is the initial state of  $\mathbf{E}_S$ , takes the FSM  $\mathbf{S}$  from any initial state to state  $s$ . On the other hand, let there exist some state  $s \in S$  and a synchronizing test case  $\mathbf{P}$  for FSM  $\mathbf{S}$  such that each trace  $\gamma$  of the FSM  $\mathbf{P}$  from the initial state to a deadlock state takes the FSM  $\mathbf{S}$  from any initial state to state  $s$ . In this case, the  $\gamma$ -successor of the set  $S_{in}$  is the set  $\{s\}$  and the state  $\{s\}$  is a definitely-reachable state in the FSM  $\mathbf{E}_S$  with a corresponding d-transfer test case  $\mathbf{P}$ . □

Proposition 1 establishes necessary and sufficient conditions for the existence of a synchronizing test case  $\mathbf{P}$  for a nondeterministic complete FSM  $\mathbf{S}$ . Moreover, it also gives a hint for a procedure to derive such a test case (when it exists). The first step of this procedure is the derivation of an initialized complete FSM  $\mathbf{E}_S$  that is equivalent to the machine  $\mathbf{S}$  under experiment. At the second step, Procedure 1 is called, and for each state  $\{s\}$  of the equivalent FSM  $\mathbf{E}_S$ , it is checked if  $\{s\}$  is definitely-reachable.

As the number of states of the machine  $\mathbf{E}_S$  does not exceed  $2^n - 1$  when  $|S| = n$ , and by construction, the FSM  $\mathbf{E}_S$  is always observable, the following statement holds.

**Proposition 2.** For each synchronizing FSM  $\mathbf{S} = (S, I, O, h_S, S_{in})$ ,  $|S| = n$ , there exists a synchronizing test case with the length that does not exceed  $2^n - n - 1$ .

As an example, consider an FSM  $\mathbf{S}$  with a flow table in Table 1 (Kushik and Yevtushenko, 2012). Table 2 represents the flow table for an initialized observable FSM  $\mathbf{E}_S$ . Note, that for the sake of simplicity we denote a subset  $\{s_1, \dots, s_k\}$  of FSM states as  $\overline{s_1, \dots, s_k}$ .

By direct application of Procedure 1, one can check that state  $\overline{3}$  is definitely-reachable in the initialized FSM  $\mathbf{E}_S$  (Table 2).



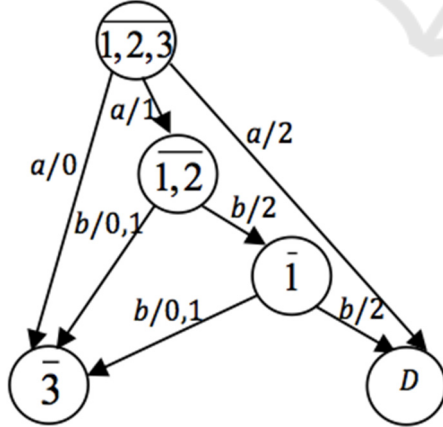
Table 1: The flow table of an FSM  $S$ .

$i/s$	1	2	3
$a$	2/1	3/0; 2/1	1/1
$b$	3/0, 1	3/0; 1/2	2/0, 1
$c$	2/0, 1	3/0; 1/2	3/0, 1

 Table 2: The flow table of the equivalent FSM  $E_s$ .

$i/s$	$\bar{1}, \bar{2}, \bar{3}$	$\bar{1}, \bar{2}$	$\bar{2}, \bar{3}$	$\bar{2}$	$\bar{3}$	$\bar{1}$
$a$	$\bar{1}, \bar{2}/1$ $\bar{3}/0$	$\bar{3}/0$ $\bar{2}/1$	$\bar{3}/0$ $\bar{1}, \bar{2}/1$	$\bar{3}/0$ $\bar{2}/1$	$\bar{1}/1$	$\bar{2}/1$
$b$	$\bar{2}, \bar{3}/0$ $\bar{2}, \bar{3}/1$ $\bar{1}/2$	$\bar{3}/0$ $\bar{3}/1$ $\bar{1}/2$	$\bar{2}, \bar{3}/0$ $\bar{2}/1$ $\bar{1}/2$	$\bar{3}/0$ $\bar{1}/2$	$\bar{2}/0$ $\bar{2}/1$	$\bar{3}/0$ $\bar{3}/1$
$c$	$\bar{2}, \bar{3}/0$ $\bar{2}, \bar{3}/1$ $\bar{1}/2$	$\bar{2}, \bar{3}/0$ $\bar{2}/1$ $\bar{1}/2$	$\bar{3}/0$ $\bar{3}/1$ $\bar{1}/2$	$\bar{3}/0$ $\bar{1}/2$	$\bar{3}/0$ $\bar{3}/1$	$\bar{2}/0, 1$

In fact, at the first iteration the set  $P = \{\bar{3}\}$  is updated with state  $\bar{1}$  due to input  $b$ . Then state  $\bar{2}$  is added to the set  $P = \{\bar{3}, \bar{1}\}$  due to and input  $b$  or  $c$ , etc. In the end, the set  $P$  equals the set of states of the FSM  $E_s$ . Therefore, there exists a synchronizing test case for the FSM  $S$ , and this test case is represented in Fig. 1. Note, that in this example the length of the test case equals three, i.e. the worst complexity case is not reached.


 Figure 1: A synchronizing test case  $P$  for an FSM  $S$  (Method 1).

In general, the upper bound on the length of a synchronizing test case is exponential, and we further discuss how this complexity can be decreased.

## 4 REDUCING THE LENGTH OF SYNCHRONIZING TEST CASES FOR NONDETERMINISTIC FSMs

In this section, we discuss how the length of synchronizing test cases can be reduced for nondeterministic FSMs. We show that proposed conditions are necessary and sufficient for non-initialized FSMs, i.e., for complete and observable machines where each state can be initial state. However, for weakly initialized machines the conditions are only sufficient when checking the existence of a synchronizing test case.

As each synchronizing test case is a homing test case with additional constraints, the existence check as well as the derivation of such test cases relies on such procedures for homing test cases. In (Kushik and Yevtushenko, 2015), it has been shown that when each state of a complete observable nondeterministic FSM can be initial, the existence of a homing test case can be checked in polynomial time. The procedure for deriving such test cases and potential heuristics improving the performance of the corresponding algorithm are discussed in (Kushik and Yenigun, 2015). We further briefly sketch this procedure (Procedure 2) as it is used for deriving a synchronizing test case.

Not every homing test case is a synchronizing test case for an FSM  $S$ , and moreover, even if a homing test case exists for  $S$ , a synchronizing test case may not exist for it. However, if for an adaptively homing non-initialized FSM  $S$  there exists a state  $s'$  such that for any state  $s \in S$ , there exists a d-transfer test case  $P(s, s')$ , then a synchronizing test case can always be derived (and vice versa).

**Proposition 3.** There exists a synchronizing test case for a complete observable FSM  $S = (S, I, O, h_s, S_m)$ ,  $S_m = S$ , if and only if FSM  $S$  is homing and there exists a state  $s' \in S$  such that for each state  $s \in S$  state  $s'$  is definitely reachable from  $s$ .

**Proof.** Indeed, given a homing FSM  $S$  with a homing test case  $R$ , consider a trace  $\gamma$  that takes  $R$  from the initial state to a deadlock state. Since  $R$  is a homing test case, the  $\gamma$ -successor of each state of the set  $S$  either does not exist or contains a unique state  $s$ . Since for any state  $s \in S$ , there exists a d-transfer test case  $P(s, s')$ , then each trace  $\sigma$  that takes  $P(s, s')$  from the initial state to a deadlock state takes the FSM  $S$  from state  $s$  to state  $s'$ , i.e., the  $\gamma\sigma$ -successor

of each state of the set  $S$  either does not exist or contains a unique state  $s'$ . On the other hand, if a synchronizing test case  $\mathbf{P}$  exists for the FSM  $\mathbf{S}$  then there exists a state  $s'$  such that for each state  $s \in S$  the test case  $\mathbf{P}$  is a d-transfer test case  $\mathbf{P}(s, s')$ .  $\square$

**Procedure 2 for deriving a homing test case for a complete observable FSM**

**Input:** A non-initialized complete observable FSM  $\mathbf{S} = (S, I, O, h_S, S)$

**Output:** A homing test case  $\mathbf{P}$  for the FSM  $\mathbf{S}$  or a message “FSM  $\mathbf{S}$  is not homing”

**Step 1** Derive a test case  $\mathbf{R}_{jk}$  for each pair  $\{s_j, s_k\}$  of different states of  $\mathbf{S}$  such that the set of states of different test cases do not intersect except for the designated deadlock state. If they do, without loss of generality, rename them. These test cases can be derived by calling a corresponding procedure from (Kushik and Yevtushenko, 2015). Represent them as tree-like FSMs (where only the leaf nodes are allowed to have an indegree greater than 1) with terminal nodes  $r_1, \dots, r_n$  and designated deadlock state  $D$ , such that there exists a trace  $\gamma$  from the initial state of  $\mathbf{R}_{jk}$  to state  $r_j$  if and only if  $\{r_j\}$  is the  $\gamma$ -successor of the pair  $\{s_j, s_k\}$ .

If at least one pair of states is not homing

**Then Return** the message “FSM  $\mathbf{S}$  is not homing”.

**Step 2**  $k := 3$

**While**  $k \leq n$

**For** each trace  $\gamma = \alpha(io)$  that takes  $\mathbf{R}_{12\dots(k-1)}$  from the initial state to the deadlock state  $r_a, a \in \{1, \dots, n\}$ , while taking  $S$  from state  $s_k$  to state  $s_i$ ;

1. Replace in  $\mathbf{R}_{12\dots(k-1)}$  the transition  $(r, i, o, r_a)$  with the transition  $(r, i, o, D_{ia})$ .
2. Append the  $\mathbf{R}_{12\dots(k-1)}$  with  $\mathbf{R}_{ia}$  at state  $D_{ia}$  always checking that the set of states of a test case under construction is different from the set of each  $\mathbf{R}_{jk}$  except of the designated deadlock state  $D$ ; if that is not the case, without loss of generality, rename the corresponding states.

**EndFor**

$k++$

**EndWhile**

**Step 3** Delete each state  $r_a$  that is not reachable from the initial state. Minimize FSM  $\mathbf{R}_{12\dots n}$  in a usual way (if necessary).

**Return** the test case  $\mathbf{R}_{12\dots n}$ .  $\square$

Note, that for a weakly initialized FSM the conditions of Proposition 3 become only sufficient. The reason is that when a synchronizing test case  $\mathbf{P}$

exists for a weakly initialized FSM  $\mathbf{S}$ , there exists a state  $s'$  such that for each initial state  $s \in S_m$  the test case  $\mathbf{P}$  is a d-transfer test case  $\mathbf{P}(s, s')$ . However, the latter does not necessarily hold for each state of FSM  $\mathbf{S}$ .

Checking the existence of a homing test case  $\mathbf{R}$  for a complete observable FSM  $\mathbf{S} = (S, I, O, h_S, S)$  can be performed in polynomial time (Kushik and Yevtushenko, 2015). On the other hand, the complexity of checking the existence of a d-transfer test case  $\mathbf{P}(s, s')$  for a state pair  $\{s, s'\}$  is polynomial as well. Indeed, this complexity is ‘hidden’ in the maximal number of iterations at Step 2 (Procedure 1). In the worst case, at each iteration, only one state is added to the set  $P$ . Therefore, after at most  $(n - 1)$  iterations, either a message “the state  $s'$  is not definitely reachable from  $s$ ” will be produced or a test case  $\mathbf{P}(s, s')$  will be returned. As the number of state pairs is polynomial, the problem of checking the existence of a state  $s' \in S$  such that for each state  $s \in S$ , state  $s'$  is definitely reachable from  $s$ , can be solved in polynomial time. In other words, the following proposition holds.

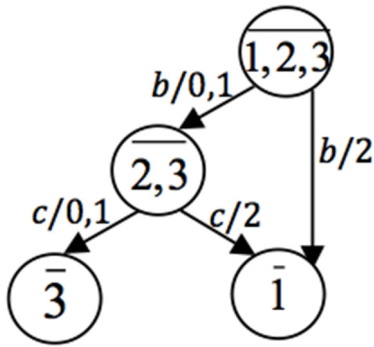
**Proposition 4.** For complete observable non-initialized FSMs checking the existence of a synchronizing test case can be performed in polynomial time.

When an adaptive synchronizing test case exists it can be derived with the use of Procedure 2 (for a homing test case) and application of Proposition 3. The corresponding algorithm is presented as Procedure 3.

**Proposition 5.** For each synchronizing FSM  $\mathbf{S} = (S, I, O, h_S, S)$ ,  $S_m = S$ , there exists a synchronizing test case with the length of the order  $O(n^3)$ .

**Proof.** In fact, the length of the homing test  $\mathbf{R}_{12\dots n}$  (if it exists) does not exceed  $(n - 1) \binom{n}{2}$ , where  $\binom{n}{2}$  is the number of different state pairs of the FSM  $\mathbf{S}$ . The length of a d-transfer test case  $\mathbf{P}(s_k, r_a)$  does not exceed  $(n - 1)$  if it exists. Therefore, the overall length of a synchronizing test case is at most  $((n - 1)^2 n)/2 + (n - 1)$ .  $\square$

In the running example, we first derive a homing test case  $\mathbf{R}_{1,2,3}$  for the set  $S_m = \{1, 2, 3\}$ . A homing test case that can be returned by Procedure 2 is illustrated in Fig. 2.


 Figure 2: A homing test case  $R_{1,2,3}$  for an FSM  $S$ .

### Procedure 3 for deriving a synchronizing test case for a complete observable FSM

**Input:** A non-initialized complete observable FSM  $S = (S, I, O, h_S, S)$

**Output:** A synchronizing test case  $P$  for the FSM  $S$  or a message “FSM  $S$  is not synchronizing”

**Step 1** Derive a homing test case  $R_{12\dots n}$  for the FSM  $S$  by calling Procedure 2.

**If** Procedure 2 returns a message “FSM  $S$  is not homing”

**Then Return** the message “FSM  $S$  is not synchronizing”

**Step 2**  $k := 1$

**While**  $k \leq n$

**For** each terminal state  $r_a$  of the homing test case  $R_{12\dots n}$ , such that  $r_a \neq s_k$

Call Procedure 1 to derive a d-transfer test case  $P(r_a, s_k)$

**EndFor**

**If**  $P(r_a, s_k)$  is derived for each terminal state  $r_a$  of the test case  $R_{12\dots n}$ , i.e. each state  $r_a$  is adaptively transferred to state  $s_k$ ,

**Then**

**For** each trace  $\gamma = \alpha(io)$  that takes  $R_{12\dots n}$  from the initial state to state  $r_a$ ,  $a \in \{1, \dots, n\}$

Replace in  $R_{12\dots n}$  the transition  $(r, i, o, r_a)$  with the transition  $(r, i, o, D_a)$ .

Append the  $R_{12\dots n}$  with  $P(r_a, s_k)$  at state  $D_a$ .

**EndFor**

Minimize the obtained FSM in a usual way (if necessary) and **Return** the reduced FSM  $R_{synch}$ .

$k++$

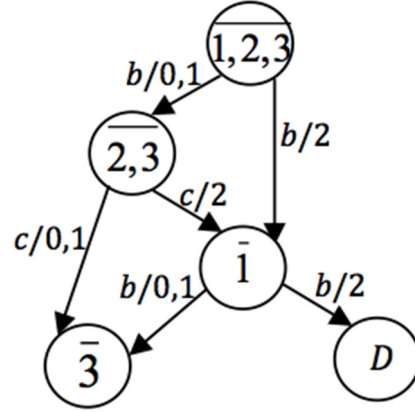
**EndWhile**

**Step 3**

**Return** the message “FSM  $S$  is not synchronizing”.

□

As state  $\bar{1}$  can be transferred to state  $\bar{3}$  by an application of a single input  $b$ , the resulting synchronizing test case still has length 3 (Fig. 3).


 Figure 3: A synchronizing test case  $R_{synch}$  for an FSM  $S$  (Method 2).

## 5 CONCLUSIONS

In this paper, the notion of an adaptive synchronizing test case for nondeterministic finite state machines has been proposed. A method for checking the existence of such test cases and a procedure for their derivation are also proposed whenever it is possible. As the height of the corresponding test case in general can be exponential, we discussed a possibility of reducing the complexity. In particular, we have proposed the necessary and sufficient conditions for non-initialized FSMs for checking the existence of a synchronizing test case in polynomial time. The conditions are only sufficient when checking the existence of a synchronizing test case for weakly initialized FSMs. As a future work, we plan to perform experimental evaluation in order to investigate how ‘realistic’ is the FSM class with the reduced complexity of adaptive synchronizing test cases. We also would like to investigate more FSM classes for which the complexity of related problems is polynomial, as well as to study adaptive synchronizing test cases for partial machines, i.e. the problem of careful adaptive FSM synchronization.

## ACKNOWLEDGEMENTS

This work is partially supported by RFBR Grant# 15-58-46013 CT\_a and TÜBİTAK Grant# 114E921.

## REFERENCES

- Cern'y, H., 1964. Pozn'amka k homog'ennym eksperimentom s konecn'ymi avtomatami. *Mat.-Fyz. Cas. Slovensk. Akad. Vied (14)*, pp. 208-216 (in Slovak).
- Chow, T.S., 1978. Testing software Design Modelled by Finite State Machines. *IEEE Trans. Software Eng.*, vol. 4, pp. 178-187.
- Gill, A., 1961. State-identification experiments in finite automata. *Information and Control*, pp. 132-154.
- Hibbard, T. N., 1961. Least upper bounds on minimal terminal state experiments of two classes of sequential machines. *Journal of the ACM*, 8(4), pp. 601-612.
- Hierons, R. M., 1998. Adaptive testing of a deterministic implementation against a nondeterministic finite state machine. *The Computer Journal*, 41(5), pp. 349-355.
- Ito, M., Shikishima-Tsuji, K., 2004. Some results on directable automata. *Lecture Notes in Computer Science*, vol. 3113, pp. 125-133.
- Klyachko, A. A., Rystsov, I. K., Spivak, M. A., 1987. In extremal combinatorial problem associated with the bound on the length of a synchronizing word in an automaton. *Cybernetics* 23, pp. 165-171.
- Kushik, N., El-Fakih, K., Yevtushenko, N., Cavalli, A., 2014. On adaptive experiments for nondeterministic finite state machines. *International Journal on Software Tools for Technology Transfer (in press)*.
- Kushik, N., Yenigun, H., 2015. Heuristics for Deriving Adaptive Homing and Distinguishing Sequences for Nondeterministic Finite State Machines. In *Proc. of the International Conference on Testing Software and Systems*, pp. 243-248.
- Kushik, N., Yevtushenko, N., 2012. Deriving adaptive synchronizing experiments for nondeterministic FSMs. *Russian Physics Journal*, T. 55, № 9/2 pp. 315-316.
- Kushik, N., Yevtushenko, N., 2013. On the length of homing sequences for nondeterministic finite state machines. In *Proc. of the 18th International Conference on Implementation and Application of Automata*, pp. 220-231.
- Kushik, N., Yevtushenko, N., 2015. Adaptive Homing is in P. In *Proc. of MBT'2015*, pp. 73-78.
- Petrenko, A., Yevtushenko, N., 2005. Conformance tests as checking experiments for partial nondeterministic FSM, In *Proc. of the FATES'2005*, pp. 118-133.
- Petrenko, A., Yevtushenko, N., 2011. Adaptive testing of deterministic implementations specified by nondeterministic FSMs. In *Proc. of the International Conference on Testing Software and Systems, Lecture Notes in Computer Science 7019*, pp. 162-178.
- Petrenko, A., Yevtushenko, N., Bochmann, G. v., 1996. Testing Deterministic Implementations from their Nondeterministic Specifications. In *Proc. of the IFIP Ninth International Workshop on Testing of Communicating Systems*, pp. 125-140.