# Probabilistic approach for minimizing checking sequences for non-deterministic FSMs

Natalia Kushik, Nina Yevtushenko, Jorge López

# Probabilistic approach for minimizing checking sequences for non-deterministic FSMs

Natalia Kushik[1], Nina Yevtushenko[2], and Jorge López[3]

[1] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau, France
[2] Ivannikov Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia
[3] Airbus, Issy-Les-Moulineaux, France
natalia.kushik@telecom-sudparis.eu, evtushenko@ispras.ru, jorge.lopez-c@airbus.com

**Abstract.** The paper is devoted to model based testing against probabilistic FSMs. Differently from our prior work in 2021, we consider checking sequences and possibilities of test suite minimization through reducing the length of the resulting checking sequence. Given a level of certainty $P$, we define a $P$-probably checking sequence under a white box testing assumption and discuss how a suffix of an input sequence can be omitted, such that the resulting sub-sequence is $P$-probably checking. The specification and possible implementations are non-initialized, i.e., the assumption of 'no reset' is supported.

**Keywords:** Model Based Testing · Non-deterministic Finite State Machines · Checking sequence · Probabilistic Approach.

## 1 Introduction

Model based test generation strategies, and in particular, Finite State Machine (FSM) test generation strategies are known to have guaranteed fault coverage under certain assumptions. When an implementation under test (IUT) is non-initialized, i.e., each implementation state can be initial, checking sequences are often considered. A checking sequence represents therefore a test suite and often consists of a combination of synchronizing / transfer sequences with the proper distinguishing sequences for a specification and related fault domain.

In this work, we focus on non-deterministic FSMs as related specifications; the specification and implementations are non-initialized, possibly non-deterministic machines, i.e., the 'no strict reset' assumption is supported. A fault domain consists of the FSM implementations that are explicitly enumerated, i.e., similar to [3, 7], we consider a test derivation strategy under the white box testing assumption. In our previous publication [4], we studied the possibility of test suite minimization through the introduction of the probabilities to the specification machine. Together with that, we introduced a new $P$-probably separability relation to be able to distinguish each faulty implementation from the specification

with a given level of certainty, $P$. We now extend this work by taking away a number of assumptions. First of all, we allow all the machines to be non-initialized and thus, we consider a test suite represented by a single (checking) sequence. Secondly, as no reset can be applied during testing, we do not minimize the test suite cardinality, instead we shorten the overall length of the checking sequence, whenever possible. The latter is based on the introduction of $P$-probably separating sequences, a $P$-probably checking sequence and a proper use of related transfer sequences.

The structure of the paper is as follows. Section 2 contains preliminaries. Non-initialized probabilistic machines are introduced in Section 3, while the checking sequence minimization strategy is presented in Section 4. Section 5 concludes the paper.

## 2 Preliminaries

An *FSM* is a 4-tuple $\mathcal{S} = \langle S, I, O, h_S \rangle$ where $S$ is a finite nonempty set of states, $I$ and $O$ are finite input and output alphabets, and $h_S \subseteq S \times I \times O \times S$ is a *transition relation*. The FSM $\mathcal{S}$ is *non-deterministic* if for some pair $(s, i) \in S \times I$, there exist several pairs $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$; otherwise, the FSM is *deterministic*. The FSM $\mathcal{S}$ is *observable* if for every two transitions $(s, i, o, s_1)$, $(s, i, o, s_2) \in h_S$ it holds that $s_1 = s_2$; otherwise, the FSM is *non-observable*. The FSM $\mathcal{S}$ is *complete* if for every pair $(s, i) \in S \times I$, there exists a transition $(s, i, o, s') \in h_S$; otherwise, the FSM is *partial* (partially specified). We hereafter consider complete observable FSMs, if not stated otherwise.

A non-deterministic FSM $\mathcal{S} = \langle S, I, O, h_S, pr \rangle$ is *probabilistic*, when for each non-deterministic transition $(s, i, o, s') \in h_S$, the function $pr$ defines the probability for the output $o$ to be produced at state $s$ under input $i$, $pr : S \times I \times O \longrightarrow [0, 1]$. For a non-deterministic FSM, the function $pr$ is defined in such a way that $\forall s \in S \ \forall i \in I \sum_{o \in O} pr(s, i, o) = 1$, and it is extended over input/output sequences from $(IO)^*$. Given an input/output sequence $\alpha/\beta = (\alpha'/\beta').(i/o)$ and a state $s_0$, $pr(s_0, \alpha, \beta) = pr(s_0, \alpha', \beta') * pr(s, i, o)$, where $s$ is the $\alpha'/\beta'$-successor of the state $s_0$ of the specification FSM $\mathcal{S}$; if the trace $\alpha'/\beta'$ is not defined at state $s_0$ then $pr(s_0, \alpha', \beta') = 0$; $pr(s, \varepsilon, \varepsilon) = 1$.

In this paper, similar to our previous work [4], for test minimization, we consider the following fault model $\langle \mathcal{S}, \cong, FD \rangle$, where $\mathcal{S}$ is complete possibly non-deterministic observable FSM, $\cong$ is the non-separability relation, and all the implementations from $FD$ are explicitly enumerated, $FD = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k\}$. FSMs $\mathcal{I}_j$ and $\mathcal{S}$ are *separable*, (written $\mathcal{I}_j \ncong \mathcal{S}$), if there exists a *separating* sequence $\alpha \in I^*$ such that the sets of output reactions of $\mathcal{I}_j$ and $\mathcal{S}$ to $\alpha$ do not intersect, i.e., $out(\mathcal{I}_j, \alpha) \cap out(\mathcal{S}, \alpha) = \emptyset$. We are interested in *exhaustive* test suites, such that each $\mathcal{I}_j \in FD$ that is separable with $\mathcal{S}$ can be detected by the test suite. Moreover, we are interested in a test suite containing a single sequence which is referred to as a *checking* sequence with respect to a corresponding fault model. Therefore such checking sequence $\alpha$ should be able to detect all non-

conforming implementations, i.e., all the implementations of the fault domain which are separable with the specification.

The main difference (with our previous work) and the main contribution of this work is that we take away the assumption of having a designated initial state, be that in the specification or in an implementation. Previously a *P-probably separating* sequence was defined as follows: $\alpha \in I^*$ is a $P$-probably separating sequence for $\mathcal{I}_j$ and $\mathcal{S}$, if $\sum_{\beta \in out(\mathcal{I}_j,\alpha) \cap out(\mathcal{S},\alpha)} pr(s_0, \alpha, \beta) \leq 1 - P$. In the latter, $pr(s_0, \alpha, \beta)$ was the probability to observe $\beta$ when $\alpha$ is applied at the initial state of the specification machine $\mathcal{S}$. We further adapt this notion to non-initialized FSMs $\mathcal{S}$ and $\mathcal{I}_j$ and explain how a checking sequence $\alpha$ can be shortened for a given level $P$.

## 3 Non-initialized probabilistic FSMs

In this section, we define the probability of an output sequence to appear as a reaction to a given input sequence, when the machine can start at any initial state. We avoid going through the determinization procedure for that matter, i.e., obtaining an initialized equivalent, not to encounter potential state explosion.

Given a non-initialized probabilistic specification FSM $\mathcal{S} = \langle S, I, O, h_S, pr \rangle$, $S = \{s_1, s_2, \ldots, s_n\}$, and an input/output pair $i/o$, $pr(\mathcal{S}, i, o) = \frac{1}{n} \sum_{s \in S} pr(s, i, o)$. The latter assumes that the probability $p$ for $\mathcal{S}$ to start in state $s_i$ is the same as in any other state $s_j \in S$. In other words, pressing a 'reset' button does not bring any certainty concerning the initial state of the machine. Assume now that for a state $s_j$, $j \in \{1, \ldots, n\}$ a probability $p_j$ is given, for the machine $\mathcal{S}$ to start in this state ($s_j$), in this case $pr(\mathcal{S}, i, o) = \sum_{j=1}^{n} p_j * pr(s_j, i, o)$ for an input/output pair $i/o$. For a given input $i$, it holds that $\sum_{o \in O} \sum_{j=1}^{n} p_j * pr(s_j, i, o) = 1$.

As an example of a non-initialized probabilistic FSM, consider the machine in Figure 1 (similar to that one in [4]). Consider an input/output pair $i_1/o_1$, for $p_1 = 0.8$, and $p_2 = p_3 = 0.1$, it holds that $pr(\mathcal{S}, i_1, o_1) = 0.74$.

As usual, we extend the behavior of the probabilistic machine over input/output sequences from $(IO)^*$. Given an input/output sequence $\alpha/\beta = (\alpha'/\beta').(i/o)$, the probability of the non-initialized $\mathcal{S}$ to produce $\beta$ on $\alpha$ is $pr(\mathcal{S}, \alpha, \beta) = \sum_{j=1}^{n} p_j * pr(s_j, \alpha, \beta)$. For example, for the FSM in Figure 1, $pr(\mathcal{S}, i_1 i_1, o_1 o_2) = 0.096$.

We are interested in a checking sequence $\alpha$ that delivers a *P-probably exhaustive* test suite for a given specification $\mathcal{S}$ and a set of its potential implementations $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k\}$. The *P-probably separability* is therefore adjusted for non-initialized machines. Input sequence $\alpha \in I^*$ is a *P-probably separating* sequence for $\mathcal{I}_j$ and $\mathcal{S}$, if $\sum_{\beta \in out(\mathcal{I}_j,\alpha) \cap out(\mathcal{S},\alpha)} pr(\mathcal{S}, \alpha, \beta) \leq 1 - P$. Note that $out(\mathcal{I}_j, \alpha)$ ($out(\mathcal{S}, \alpha)$) is the union of all output reactions $\beta$ on the sequence $\alpha$ that can be obtained at any initial state of $\mathcal{I}_j$ ($\mathcal{S}$).

An interesting question arises about the probability distribution for initial states of implementation FSMs. In this paper, we assume that all the states can be initial with the same probability. If this assumption is not supported then the formula for defining a $P$-probably separating sequence for the specification and such implementation should be modified, accordingly.

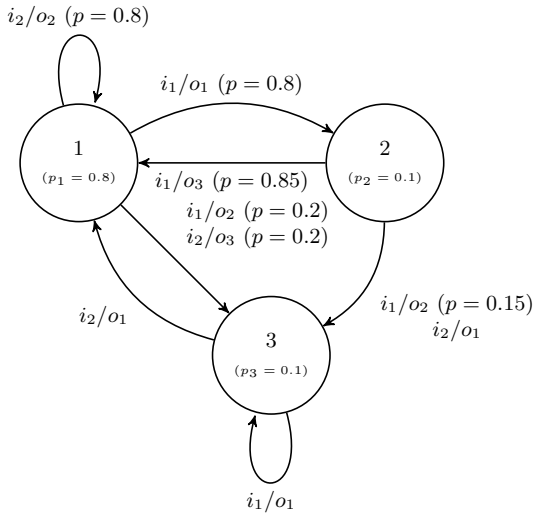


**Fig. 1.** An example probabilistic FSM $\mathcal{S}$

Coming back to the same example of $\mathcal{S}$, let us consider an implementation $\mathcal{I}_1 \in FD$ in Figure 2. According to the above definition the sequence $\alpha = i_2 i_2 i_1 i_1 i_1$ is a 0.9-probably separating sequence for $\mathcal{I}_1$ and $\mathcal{S}$ (Figure 1).

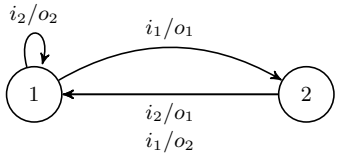


**Fig. 2.** An implementation FSM $\mathcal{I}_1 \in FD$

A sequence $\alpha$ is *P-probably checking* for the fault model $\langle \mathcal{S}, \cong, FD = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k\}\rangle$, if this sequence $P$-probably separates each implementation $\mathcal{I}_j$, $j \in \{1, \ldots, k\}$, from the specification $\mathcal{S}$.

## 4 Minimizing a checking sequence with a level of *P*-exhaustiveness

Assume that a sequence $\alpha$ is a checking sequence for the fault model $FM = \langle \mathcal{S}, \cong, FD = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k\}\rangle$. Given a level $P$ of certainty, the question arises: can we shorten $\alpha$ in such a way that the resulting sequence would be $P$-probably checking for $\langle \mathcal{S}, \cong, FD = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k\}\rangle$ ? We conjecture the following: non-initialized implementations can be hard to test as in the checking sequence a

*transfer* to a known state of the implementation, is implicitly used or even explicitly included during its derivation (see some related works on the checking sequence derivation, for example in [1,2,5,6]). We therefore propose the following: before the application of the sequence $\alpha$ or its shorter preamble $\alpha'$, one can apply a synchronizing sequence $SS$ with further verification that the sequence $SS.\alpha'$ is $P$-probably checking for $FM$[1]. Note however, that the sequence $SS$ should be synchronizing for all the implementations $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$, and this sequence can be derived for a single FSM which is the direct sum of $\mathcal{I}_j, j \in \{1, \ldots, k\}$. The latter contains all the transitions of each implementation $\mathcal{I}_j$ and thus, its synchronizing sequence is also one for each $\mathcal{I}_j, j \in \{1, \ldots, k\}$. Note that if implementations are non-initialized but deterministic then such a sequence can be efficiently computed [8] (in polynomial time, if the number of mutants $k$ is polynomial too w.r.t. $n$, for the corresponding automaton where the outputs are omitted).

As an example of the proposed strategy, consider again the FSM $\mathcal{S}$ in Figure 1, and the $FD = \{\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3\}$. $\mathcal{I}_1$ is shown in Figure 2, while $\mathcal{I}_2$ and $\mathcal{I}_3$ in Figure 3 and Figure 4, respectively. Note that the sequence $\alpha = i_1 i_1 i_2 i_2 i_1 i_1 i_2 i_1 i_2 i_2$ is a checking sequence for $\langle \mathcal{S}, \cong, FD = \{\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3\}\rangle$. The direct sum for the three implementations possesses a synchronizing sequence $SS = i_2 i_2$; indeed, each of the implementations has the same $SS$, which can be checked by direct inspection. We append this $SS$ as a prefix to $\alpha$ and start cutting its suffix. For $P = 0.8$, one can cut 8 inputs in the resulting sequence, i.e., $SS.\alpha' = i_2 i_2 i_1 i_1$ is 0.8-probably checking sequence and it is six inputs shorter than the initial $\alpha$. This approach can be therefore applied iteratively, until the level $P$ of exhaustiveness is respected.

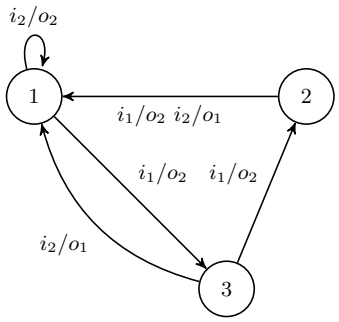


**Fig. 3.** An implementation FSM $\mathcal{I}_2 \in FD$

The following suggestions for deriving a shorter checking sequence can be made based on the considered example. First of all, the use of proper final state identification sequences such as homing and synchronizing sequences, can help

[1] Such checking is needed to assure that $SS.\alpha'$ is $P$-probably separating for each implementation $\mathcal{I}_j, j \in \{1, \ldots, k\}$ and $\mathcal{S}$.

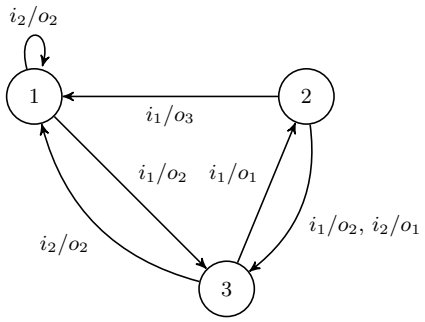


**Fig. 4.** An implementation FSM $\mathcal{I}_3 \in FD$

to derive a shorter checking sequence with the given level of certainty. Secondly, when deriving a checking sequence it seems to be worth considering a deterministic projection of the specification where transitions and the initial state have the highest probability. In a deterministic projection only one transition is left at each state for every input. Finally, if all the implementations are deterministic it is worth introducing another conformance probabilistic relation such as for example, the $P$-probably reduction when the behavior of an implementation is $P$-probably included in that of the specification, for a given level of certainty $P$. All these challenging issues should be studied in details in the future.

## 5 Conclusion

In this paper, we presented a probabilistic approach for minimizing the length of a checking sequence, probably keeping a given level of its exhaustiveness. It is a continuation of the paper [4], where only initialized machines (specification and its implementations) were considered. An interesting direction would be to combine both approaches, when some of the test sequences could be regrouped together, i.e., building a $P$-probably checking sequence for a subset of implementations.

There are many possibilities for future work, we state some of these future directions. We did not discuss any $P$-probably checking sequence derivation scenarios in detail, assuming that a starting sequence to be shortened is given. At the same time, we did not consider any other testing assumptions nor other conformance relations. Finally, experimental evaluations need to be performed to see how often the approach brings good practical results.

## References

1. Hennie, F.C.: Fault detecting experiments for sequential circuits. In: Proceedings of Symposium on Switching Circuit Theory and Logical Design. pp. 95–110 (1964)

2. Jourdan, G., Ural, H., Yenigün, H.: Reduced checking sequences using unreliable reset. Inf. Process. Lett. **115**(5), 532–535 (2015). https://doi.org/10.1016/j.ipl.2015.01.002
3. Kushik, N., Yevtushenko, N., Cavalli, A.R.: On testing against partial non-observable specifications. In: 9th International Conference on the Quality of Information and Communications Technology, QUATIC 2014, Guimaraes, Portugal, September 23-26, 2014. pp. 230–233. IEEE Computer Society (2014). https://doi.org/10.1109/QUATIC.2014.38
4. Kushik, N., Yevtushenko, N., López, J.: Testing against non-deterministic fsms: A probabilistic approach for test suite minimization. In: Clark, D., Menéndez, H.D., Cavalli, A.R. (eds.) Testing Software and Systems - 33rd IFIP WG 6.1 International Conference, ICTSS 2021, London, UK, November 10-12, 2021, Proceedings. Lecture Notes in Computer Science, vol. 13045, pp. 55–61. Springer (2021). https://doi.org/10.1007/978-3-031-04673-5_4
5. Nguena-Timo, O., Petrenko, A., Ramesh, S.: Checking sequence generation for symbolic input/output fsms by constraint solving. In: Fischer, B., Uustalu, T. (eds.) Theoretical Aspects of Computing - ICTAC 2018 - 15th International Colloquium, Stellenbosch, South Africa, October 16-19, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11187, pp. 354–375. Springer (2018). https://doi.org/10.1007/978-3-030-02508-3_19
6. Petrenko, A., Yevtushenko, N.: Conformance tests as checking experiments for partial nondeterministic FSM. In: Formal Approaches to Software Testing, 5th International Workshop, FATES 2005, Edinburgh, UK, July 11, 2005, Revised Selected Papers. pp. 118–133 (2005). https://doi.org/10.1007/11759744_9
7. Poage, J.F., McCluskey, E.J.: Derivation of optimum test sequencies for sequential machines. In: 1964 Proceedings of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design. pp. 121–132 (1964). https://doi.org/10.1109/SWCT.1964.7
8. Volkov, M.V.: Synchronizing automata and the cerny conjecture. In: Martín-Vide, C., Otto, F., Fernau, H. (eds.) Language and Automata Theory and Applications, Second International Conference, LATA 2008, Tarragona, Spain, March 13-19, 2008. Revised Papers. Lecture Notes in Computer Science, vol. 5196, pp. 11–27. Springer (2008). https://doi.org/10.1007/978-3-540-88282-4_4