Proc. of the Third Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-077-4 doi: 10.15224/ 978-1-63248-077-4-97

On Testing Safety Properties of synchronous FSM composition

Maxim Gromov, Nina Yevtushenko

***Abstract*—This paper studies the safety issues of the synchronous composition of Finite State Machines (FSMs). Synchronous FSM composition is widely used when designing and analyzing various aspects in hardware implementation including so-called Trojan subcircuits. When an external input sequence is applied and component FSMs cannot agree on matched internal actions or there are different matching options, the sequence can destruct or block a corresponding device and thus, should be clearly avoided. In this paper, we propose a formal approach for analyzing which external input sequence can induce such situations (if any) and then to derive a finite automaton that represents the set of all possible safe external input sequences.**

***Keywords*— Finite State Machines (FSMs), synchronous FSM composition, safety properties.**

I. Introduction

The problem of designing safe digital circuits is well-known: given a multi-component digital circuit, to avoid the situations when for some external input sequences, component FSMs cannot agree on matched internal actions or there are different such options [1, 2, 3]. Such situations if occurred can destruct a corresponding device and thus, have to be detected at the verification phase.

In the paper we consider this problem for sequential logic represented by a synchronous composition of Finite State Machines (FSMs), i.e., a network of synchronized interacting components each modeled by an FSM. The problem can be formalized by using different approaches; one of which is based on the synchronous regular language composition represented by finite automata [3]. In [3], the global automaton that represents all possible traces of the FSM composition is derived. Based on this global automaton, for each current state it can be determined whether the composition is not progressive, i.e., whether component FSMs cannot agree on matched internal actions at some current states. It also can be determined when there are different matched options. If a current composition is not safe then an unsafe component can be replaced with another safe component preserving the external composition behavior. A safe component can be selected based on a non-deterministic FSM that represents all possible solutions to an appropriate FSM [3].

Maxim Gromov, Nina Yevtushenko
Tomsk State University
Russia
gromov

The rest of the paper is organized as follows. Section II introduces the preliminaries. Section III presents the formal approach for deriving synchronous FSM compositions. The safety properties of the composition of partial FSMs are discussed in Section IV where an approach for checking these properties is also proposed as well as a technique for deriving the set of all safe external input sequences. Section V concludes the paper and in this section, appropriate directions for the future work are discussed.

II. Preliminaries

A *Finite State Machine* (*FSM*) or simply a *machine* throughout this paper is a 5-tuple $S = (S, I, O, h_S, s_0)$, where S is a finite non-empty set of states with the initial state s_0, I and O are the input and output alphabets, and $h_S \subseteq S \times I \times O \times S$ is a *transition* relation. The 4-tuple $(p, i, o, n) \in h_S$ is a *transition* from the present state p under input symbol i to the next state n with output symbol o, denoted as $p \rightarrow n$. Since each FSM has an equivalent observable FSM [1], in this paper, an FSM is assumed to be *deterministic* and *complete*, i.e. for each pair $(p, i) \in P \times I$ there exists exactly one state $n \in S$ and one output $o \in O$ such that $(p, i, o, n) \in h_S$. The FSM is *connected* if each state is reachable from the initial state. FSM $B = (B, I, O, h_B, b_0)$ is a *submachine* of S if $B \subseteq S$ and $h_B \subseteq h_S$.

In a deterministic FSM, the transition relation is usually replaced by two functions, the transition function δ and the output function λ. Therefore, a complete deterministic FSM is a 6-tuple $S = (S, I, O, \delta, \lambda, s_0)$ where $\delta(p, i) = n$ is a *transition* function and $\lambda(p, i) = o$ is an output function. A complete deterministic FSM is of a *Moore-type* [4] if the output function does not significantly depend on an input, i.e., given a state $p \in S$ and any two input symbols i_1 and i_2, it holds that $\lambda(p, i_1) = \lambda(p, i_2)$.

FSMs usually describe the behavior of digital circuits that transform input sequences into output sequences. In the usual way, the relation h_S is extended to sequences over alphabets I and O. Let I^* denotes the set of all finite strings over an alphabet I including the empty sequence ε. Given states $p,n \in S$, $i_1...i_k \in I^*$ and $o_1...o_k \in O^*$, the 4-tuple $(i_1...i_k, p, n, o_1...o_k) \in h_S$ if and only if there exist states $s^1,\ldots,s^{k+1}$ such that $s^1 = p,\ldots, s^{k+1} = n$ and for each $j = 1,\ldots,k$, $(i_j, s^j, s^{j+1}, o_j) \in h_S$. The sequence $(i_1o_1)...(i_ko_k)$ is called an *I/O sequence* of FSM S at state p or a *trace* at state p. The set of all traces at state s, denoted $L_S(s)$, is a subset of $(I \times O)^*$ and is called the *behavior* or the *language* of the FSM S at state s. The set $L_S(s_0)$ of traces of FSM S at initial state s_0 is denoted L_S, for short. A complete FSM is *reduced* if for each two states the languages at these states do not coincide. It is known that for each complete FSM there exists a reduced FSM with the same behavior [5].

Proc. of the Third Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-077-4 doi: 10.15224/ 978-1-63248-077-4-97

A multi component sequential circuit usually is represented as a composition of black boxes with *ports*. When the behavior of each black box is described by a deterministic complete FSM the synchronous composition of FSMs [1, 6] can model the synchronous behavior of a multi component sequential circuit.

Consider the collection of n component FSMs $S_1 = (S_1, I_1, O_1, h_1, s_{10}), \ldots, S_n = (S_n, I_n, O_n, h_n, s_{n0})$. If the component FSM S_i has k input ports with alphabets $I_{i1}, \ldots, I_{ik}$ and m output ports with alphabets $O_{i1}, \ldots, O_{im}$ then the input alphabet I_i of the component FSM A_i is the Cartesian product of input alphabets $I_{i1}, \ldots, I_{ik}$, i.e. $I_i = I_{i1} \times \ldots \times I_{ik}$, while the output alphabet O_i is the Cartesian product of output alphabets $O_{i1}, \ldots, O_{im}$, i.e. $O_i = O_{i1} \times \ldots \times O_{im}$.

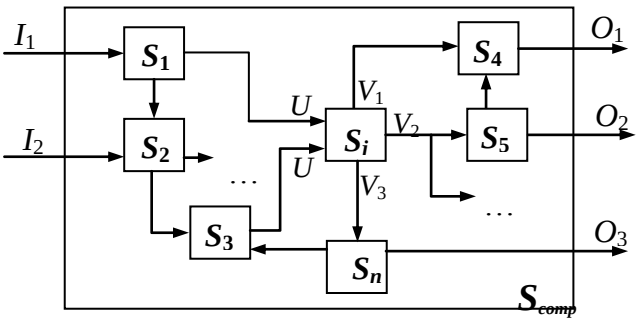


Fig. 1 – A system of interacting FSMs

For allowing the different modules to communicate with each other, their ports must be connected. The structure of the composition is defined by the partition π over the set of all ports. Each block of π has all the ports that are connected and thus, at each moment, we are required to have the same signal at all these ports. By this reason, we assume that each block has at most one output port. For the sake of simplicity, alphabets related to connected ports, i.e. alphabets of each block of the partition π, are further assumed to be equal and are denoted with the same capital letter. External inputs of the composed system are associated with blocks that do not have output ports. An external output can be associated with any block that has an output port. Let $\{I_1, \ldots, I_m\}$ and $\{O_1, \ldots, O_k\}$ be the sets of alphabets associated with external inputs and external outputs of the composed system. Thus, each external input of the overall system is the vector $\mathbf{i} = (i_1, \ldots, i_m)$, while each external output of the system is the vector $\mathbf{o} = (o_1, \ldots, o_k)$. A system of interacting FSMs is shown in Fig. 1.

For communication between several modules, the interactions between the different modules within the system are assumed to be synchronized by a global clock, and there is an interaction at each port during each clock period. Such composition is called "synchronous composition". The joint behavior of the composition of FSMs $S_1, \ldots, S_n$ can be described by the composed FSM S_{comp}.

The input alphabet of the FSM S_{comp} is the Cartesian product $I_1 \times \ldots \times I_m$, the output alphabet of the FSM S_{comp} is the Cartesian product $O_1 \times \ldots \times O_k$. Let $V_1, \ldots, V_r, U_1, \ldots, U_f$ be the internal alphabets of the composition. The initial state of the FSM S_{comp} is $s_0 = s_{10} \ldots s_{n0}$, where $s_{10}, \ldots, s_{n0}$ are initial states of the component FSMs $S_1, \ldots, S_n$. Other states of the FSM can be derived performing the reachability analysis, i.e., by use of the reachability tree.

The root of the tree is the initial state $s_0 = s_{10} \ldots s_{n0}$. Given the node $s_1 \ldots s_n$ of the tree, an input vector $\mathbf{i} = (i_1, \ldots, i_m)$ and an output vector $\mathbf{o} = (o_1, \ldots, o_m)$, there is a transition from state $s_1 \ldots s_n$ to state $s_1' \ldots s_n'$ labeled with the I/O pair $\mathbf{i}/\mathbf{o}$ if and only if there exist internal matching signals $v_1, \ldots, v_r$ and $u_1, \ldots, u_p$ such that corresponding 4-tuples are in the transition relations of component FSMs, i.e. $(\mathbf{i_1}, s_1, s_1', \mathbf{o_1}) \in h_1, \ldots, (\mathbf{i_n}, s_n, s_n', \mathbf{o_n}) \in h_n$, where $\mathbf{i_j}$, $\mathbf{o_j}$ are corresponding input and output vectors of S_j, $j = 1, \ldots, n$, and therefore, some components of $\mathbf{i_j}$ and $\mathbf{o_j}$ can be taken from the internal alphabets $V_1, \ldots, V_r$ and $U_1, \ldots, U_f$. The process is terminated when the successors of each node are in the tree. The reduced form of the machine derived from the reachability tree is called the *synchronous composition* $S_1 \bullet \ldots \bullet S_n$ of component FSMs $S_1, \ldots, S_n$. The composition is *safe* if for each input at each node there is exactly one successor; otherwise, the composition is *unsafe*. It is well known that given complete and deterministic component FSMs, in general case, the composition is not safe, i.e., the FSM S_{comp} obtained from the reachability tree can be partial and non-deterministic [3].

A truncated successor can be represented in a more compact way as an appropriate product of component FSMs. The set of safe external input sequences can be then derived as the projection of the product onto external composition alphabets.

III. Product based approach for checking the safety properties of the synchronous FSM composition

We illustrate our approach for binary FSM composition shown in Fig. 2.

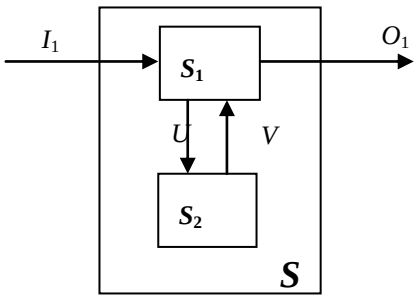


Fig. 2 – Composition of two FSMs

A. *Global machine*

In order to describe the composition of two FSMs by a product machine each FSM should be represented as a component of such composition. Since the FSM composition is based on the interchange of sequences at the ports, we use the notion of finite automata [7] when representing such a composition. A *finite automaton* is a quintuple $\mathcal{S} = \langle S, A, \delta_S, s_0, F_S \rangle$, where S is a finite nonempty set of states with the initial state s_0 and a subset F_S of *final* (or *accepting*)

Proc. of the Third Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-077-4 doi: 10.15224/ 978-1-63248-077-4-97

states, A is an alphabet of actions, and $\delta_S \subseteq A \times S \times S$ is a transition relation. We say that there is a transition from a state s to a state s' labeled with an action a, if and only if the triple (a, s, s') is in the transition relation δ_S. The automaton S is called *deterministic*, if for each state $s \in S$ and any action $a \in A$ there exists at most one state s', such that $(a,s,s') \in \delta_S$. If S is not deterministic, then it is called *nondeterministic*.

Well-known results state that each regular language can be represented by a deterministic finite automaton and that regular languages are closed under the union, intersection and complementation [7]. Regular languages are also closed under projection, lifting, restriction and expansion. Below we sketch the constructions for the less known operations of projection, lifting, restriction and expansion.

Projection ($\downarrow$) Given FA F that accepts language L over $I \times U$, FA $F_{\downarrow I}$ that accepts language $L_{\downarrow I}$ over I is obtained by replacing each edge $((i,u),s,s')$ in F by the edge (i,s,s').[1]

Lifting ($\uparrow$) Given FA F that accepts language L over I, FA $F_{\uparrow\{I,U\}}$ that accepts language $L_{\uparrow\{I,U\}}$ over $I \times U$ is obtained by replacing each edge (i,s,s') in F by the set of edges $\{((i,u),s,s'): u \in U\}$.

Procedure 1 Deriving the global automaton for the composition of two FSMs.

Input. FSMs S_1 and S_2 (Fig 1)

Output. The FSM that describes all possible global traces of the joint work of FSMs S_1 and S_2

1. Derive finite automata F_1 and F_2 which accept respectively languages of FSMs S_1 and S_2.
2. Obtain the automaton $F_{2\uparrow\{I,\ O\}}$ by replacing each label (u,v) with all 4-tuples (i, u, v, o), $i \in I$, $o \in O$.
5. Build the intersection $F_1 \cap (F_{2\uparrow\{I,\ O\}})$ keeping the same order of alphabets I, V, U, O in both automata. The states of the obtained automaton are pairs of states of F_1 and $F_{2\uparrow\{I,\ O\}}$; the initial state is the pair of initial states, and a state of the intersection is accepting if both states of the pair are accepting.
6. Project $F_1 \cap F_{2\uparrow\{I,\ U,\ O\}}$ onto alphabets I (inputs) and $V \times U \times O$ (outputs)[1]. Call the obtained FSM a global FSM G for the composition of FSMs S_1 and S_2. □

The global FSM G is constructed keeping in mind the same rules as for the truncated successor tree. For this reason, it can be used for checking the safety properties of the composition.

B. *Detecting safety synchronous composition properties*

Given deterministic and complete component FSMs S_1 and S_2, let G be the global FSM returned by Procedure 1. The following statement holds.

Proposition 1. The composition is safe if and only if the global machine G is complete and deterministic.

[1] Apply the subset construction to obtain an equivalent deterministic FA.

Indeed, if there is no transition at some current state of G that means that there are no matching internal actions at component current states. If at some current state, there are two transitions the latter means that nondeterministic behavior can occur as there are at least two pairs of matching internal actions at current states of component FSMs.

Consider FSMs in Fig. 3; the global machine is in Fig 4. By direct inspection, one can assure that at states a_1 and b_1 of component FSMs there are no matching internal signals for the input i_2.

In order to derive safe input sequence unsafe states should be iteratively delete from the global FSM. An input is *unsafe* at state *sp* if there is no transition or there are at least two transitions in the global FSM.at this state under this input. A state is *unsafe* if there are no transitions at this state or all the inputs are unsafe at this state.

Procedure 2 Deriving safe input sequences for the composition of two FSMs.

Input. Global FSM for the composition of FSMs S_1 and S_2 (Fig 1)

Output. The automaton representing the set of all safe input sequences

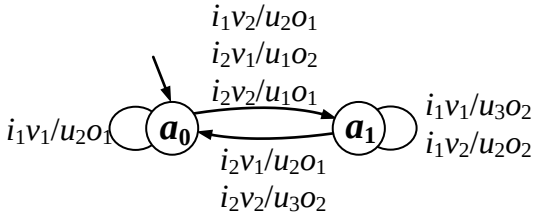
Iteratively delete unsafe inputs at each state, all unsafe states and transitions to these states.

If the initial state is deleted then there are no safe input sequences in the composition.

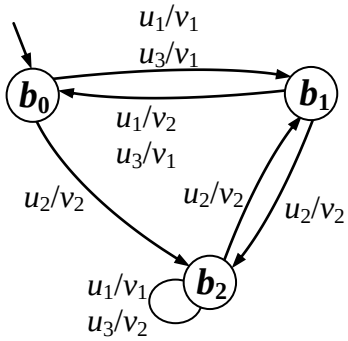
If the initial state is not deleted and the machine has only safe states then take the projection onto external alphabet. □

Proposition 2. The automaton returned by Procedure 2 represents the set of all safe input sequences. □

Consider the FSM composition in Fig. 2 with component FSMs in Fig. 3.



a)



b)

Fig. 3 – FSMs S_1 (a) and S_2 (b)

Proc. of the Third Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-077-4 doi: 10.15224/ 978-1-63248-077-4-97

The global FSM G is shown in Fig. 4. The state set of G equals $\{a_0b_0, a_1b_1, a_0b_2, a_1b_2\}$. There is no transition from state a_1b_1 under input i_2. Correspondingly, the automaton representing safe input sequences for this composition is shown in Fig. 5.

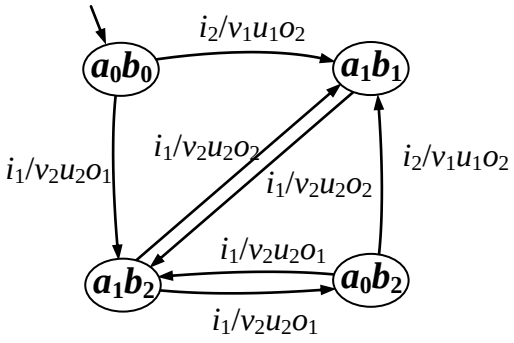


Fig. 4 – FSM G

The only case when the composition is guaranteed to be safe is given in Proposition 3 when each loop has a Moore FSM. Unfortunately this condition is too strong and does not hold for real compositions.

Proposition 3 [3]. Given a composition of complete and deterministic FSMs $S_1, \ldots, S_n$, let each loop have a machine of a Moore-type. Then the FSM $S_1 \bullet \ldots \bullet S_n$ is complete and deterministic.

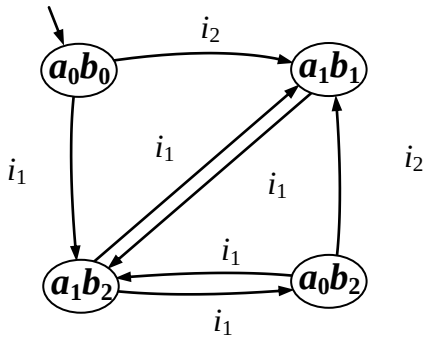


Fig. 5 – The automaton representing safe external input sequence

IV. Concluding remarks

This paper proposes a formal model for checking whether the composition is safe, i.e., has no input sequences which lead to states where deadlocks and concurrency is possible, When composition possesses these futures it opens a door for inserting malicious subcircuits which influence the safety of corresponding hardware. The future research is devoted for more rigorous analysis of inserting such unsafe circuits into FSM compositions.

References

[1] T. Kam, T. Villa, R. Brayton, A. Sangiovanni-Vincentelli, *Synthesis of FSMs: functional optimization*, Kluwer Academic Publishers, 1997.

[2] Yevtushenko, N., Villa, T., Brayton, K.R., Petrenko, A., Sangiovanni-Vincentelli. (2008) Compositionally progressive solutions of synchronous FSM equations. Discrete Event Dynamic Systems: Theory and Applications, 18 (1), 51-89.

[3] Villa, T., Yevtushenko, N., Brayton, K.R., Mishchenko, A., Petrenko, A., Sangiovanni-Vincentelli, A. (2012) The Unknown Component Problem: Theory and Applications. Springer.

[4] Moore E. F. Gedanken-experiments on Sequential Machines. Automata Studies, Annals of Mathematical Studies, 34, 129–153. Princeton University Press, Princeton, N.J.(1956).

[5] P.H. Starke Abstract automata, American Elsevier Pub. Co (1972).

[6] J.Hartmanis and R.E.Stearns. The algebraic structure theory of sequential machines. Prentice-Hall, N.Y., 1966, 210 p.

[7] A. V. Aho, J. E. Hopcroft, J. D. Ullman, Data Structures and Algorithms. Addison-Wesley, 1983.

About Author (s):

Maxim Gromov received his diploma degree in Computer Science from Tomsk State University in 2009. From 2009 till now he is Associated Proffesor at the department of Computer Science in Discrete Event Systems. His research interests include automata theory, complexity of computations and software testing.

Nina Yevtushenko received her diploma degree in Electrical Engineering from Tomsk State University. Until 1991 she worked as a researcher with the Siberian Scientific Institute of Physics and Technology. From 1991 she joined Tomsk State University as a professor and presently she leads a research team working on the synthesis and analysis of discrete event systems. She published four books and many research papers. Her research interests include formal methods, automata theory, distributed systems, protocol and software testing.