Synchronous composition of quasi-complete and quasi-deterministic FSMs

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

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

Nina Yevtushenko
Software engineering deparment
Ivannikov Institute for System Programming
Moscow, Russia
evtushenko@ispras.ru

Tiziano Villa
Computer Science Department
University of Verona
Verona, Italy
tiziano.villa@univr.it

***Abstract*——In this paper, we consider the synchronous composition of Finite State Machines (FSMs) that corresponds to instantaneous communication of hardware components. Such FSMs can be deterministic or nondeterministic, partial or complete. We first extend the existing synchronous composition operator from a pair of components to a collection of components with multiple input and output ports, and provide a procedure to compute the composition working directly on the collection of transition tables of the component FSMs. Then, based on the notion of input-output Moore pair (the output is not sensitive to the input), we prove the following sufficient condition: if the composition has the property that every cycle of ports has a component with a Moore pair, then the class of quasi-complete and quasi-deterministic FSMs is closed under the proposed synchronous composition.**

Keywords—Finite State Machine (FSM), quasi-deterministic FSM, quasi-complete FSM, Moore pair, synchronous composition

I. Introduction

Many complex discrete systems are organized as a collection of interacting components and transition system models are widely used for analyzing and designing such composed systems. There are various composition operators over transition systems [1-14] which are used for components which are reactive (actions are divided into inputs and outputs) and non-reactive (actions are not divided into inputs and outputs) or extended transitions systems such as timed and hybrid automata; the composition can be synchronous and asynchronous, etc. The main issue is to guarantee the closure of a given formal model under a chosen composition operator. In this paper, we consider the so-called synchronous composition of Finite State Machines (FSMs) [12] that corresponds to instantaneous communication of hardware components. An FSM describes the behavior of a system that moves from state to state and produces an output when an input is applied. Such systems, and thus, FSMs which describe their behavior, can be deterministic or nondeterministic, partial or complete. An FSM is complete and deterministic if, for each state and each input, there exists exactly one pair 'next_state, output'. To the best of our knowledge, the synchronous composition of FSMs was first considered in [1] where the authors compose Moore FSMs [15], i.e., deterministic complete FSMs where output values do not depend on input values. The composition is constructed by the use of a successor tree and as each component is a Moore FSM, the composition operator returns a Moore FSM. The condition was later weakened in [12, 16] (see also [11] for related work), where the authors prove that the synchronous compositions of complete deterministic FSMs is complete and deterministic if every composition cycle has a Moore FSM.

Composition of two FSMs was addressed in [11]; later in [16], synchronous composition of many components was reduced to iterating the composition of two components, since synchronous composition over FSMs is associative. A sufficient condition for the composition to be deterministic and complete is defined for an embedded component [11]: internal output values of the embedded component do not depend on its internal input values. In [2, 6], the asynchronous composition of two input/output automata is defined via appropriate composition of component languages; the complete description of such operator via the language composition was presented in [12], and extended in [16] to the synchronous composition of many component FSMs. Most of the previous literature on synchronous composition addressed complete deterministic FSMs. However, in many applications it is important to model partial and nondeterministic behaviors. Therefore, in this paper we study synchronous composition over quasi-complete and quasi-deterministic FSMs.

An FSM is quasi-complete if the set of inputs is partitioned into classes such that at each state exactly one input of the class is a defined input, i.e., for any input of a class the FSM will have the same defined reaction. An FSM is quasi-deterministic if the set of outputs is partitioned into classes and for each defined input any output of a corresponding output class can be produced at once. A deterministic complete FSM is a particular case of an FSM that is quasi-complete and quasi-deterministic. In [17, 18], the composition of $n > 2$ component FSMs is reduced to deriving such a composition of $(n - 1)$ component FSMs. As mentioned above, there exist some sufficient conditions when it is the case. However, the only necessary and sufficient conditions for checking whether a binary composition of complete possibly nondeterministic components is complete [12] are based on solving an equation $A \bullet X = S$ over FSM where X is a free variable and $\bullet$ is the synchronous composition, and is related to checking whether a corresponding equation has a compositionally progressive solution. The problem of deriving the largest compositionally progressive solution is rather hard and in this paper, following [17, 18], we analyze when the class of quasi-complete and quasi-deterministic FSMs is closed under the synchronous composition operator, and prove that it is the case when every cycle over component ports has a Moore-pair, i.e., a pair of <input-port, output_port> of some component FSM such that the signal at the output port does not depend on the signal of the input port. Moreover, we define the composition directly for any number of component FSMs with multiple input and output ports.

The rest of the paper is structured as follows. Section 2 deals with FSMs and the synchronous composition of FSMs. An algorithm for deriving FSM composition without using finite automata representing FSM languages is proposed in Section 3. A novel sufficient condition for the synchronous composition of quasi-deterministic and quasi-complete FSMs to be quasi-deterministic and quasi-complete is proven in Section 4. Section 5 concludes the paper.

II. Finite State Machines

In this section, we remind the main FSM concepts and introduce the notion of an FSM that has several input and several output ports. The latter is important since when combining FSMs where inputs and outputs are Boolean vectors it is exactly the case while most papers consider binary compositions where there are at most two input and two output ports.

A. Finite State Machines with single input and output ports

A *finite state machine (FSM)*, often simply called a *machine*, is a quintuple $A = \langle S_A, X_A, Y_A, h_A, s_{A0}\rangle$, where S_A is a finite nonempty set of states with the initial state s_{A0}, X_A and Y_A are input and output alphabets, and $h_A \subseteq S_A \times X_A \times Y_A \times S_A$ is a transition relation. We say that there is a transition from a state $s \in S_A$ to a state $s' \in S_A$ labeled with an I/O pair x/y (an I/O pair xy) if and only if the 4-tuple (s, x, y, s') is in the transition relation h_A. An FSM A is *deterministic* if for each state $s \in S_A$ and each input $x \in X_A$, there exists at most one pair (y, s') such that $(s, x, y, s') \in h_A$. If A is not deterministic, then it is called *nondeterministic*. An FSM A is *complete*, if for each state $s \in S_A$ and each input $x \in X_A$ there exists at least one pair (y, s'), such that $(s, x, y, s') \in h_A$. If A is not complete, then it is *partial*. For a partial FSM A, given a state s and input x, x is a *defined* input at state s if there exist at least one pair (y, s'), such that $(s, x, y, s') \in h_A$. The set of defined inputs at state s is denoted $D_A(s)$.

As usual, the transition relation h_A of the FSM can be extended to sequences over the alphabet X_A and Y_A. The extended relation is also denoted by h_A and is a subset of $S_A \times X_A^* \times Y_A^* \times S_A$. By definition, for each state $s \in S_A$ the tuple $(s, \varepsilon, \varepsilon, s)$ is in the relation h_A where ε is the empty sequence. Given a tuple $(s, \alpha, \beta, s') \in h_A$, $\alpha \in X_A^*$, $\beta \in Y_A^*$, and an input $x \in X_A$ and an output $y \in Y_A$, the tuple $(s, \alpha x, \beta y, s'') \in h_A$ if $(s', x, y, s'') \in h_A$.

Given state s, an input/output sequence $x_1y_1 \ldots x_ky_k$, $x_1 \ldots x_k \in X_A^*$, $y_1 \ldots y_k \in Y_A^*$, such that $(s, x_1 \ldots x_k, y_1 \ldots y_k, s') \in h_A$, is called a *trace* of FSM A at state s. Hereafter, such a trace can also be written as $x_1 \ldots x_k / y_1 \ldots y_k$ using its *input sequence* $x_1 \ldots x_k$ and the corresponding *output sequence* $y_1 \ldots y_k$ or $(x_1, y_1) \ldots (x_k, y_k)$. The set of all traces at state s is denoted $Tr_A(s)$. We denote Tr_A the set of traces at the initial state s_{A0}, i.e., the set of traces of S_A, for short.

FSM A is *observable* if $\forall s \in S_A\ \forall x \in X_A\ \forall y \in Y_A\ \forall s', s'' \in S_A$ it holds that

$(s, x, y, s') \in h_A\ \&\ (s, x, y, s'') \in h_A \Rightarrow s' = s''$.

In other words, FSM A is *observable* if for each triple $(s, x, y) \in S_A \times X_A \times Y_A$ there exists at most one state $s' \in S_A$ such that $(s, x, y, s') \in h_A$. Given an observable FSM A, state $s \in S_A$ and a trace $x_1y_1 \ldots x_ky_k$, $x_1 \ldots x_k \in X_A^*$, $y_1 \ldots y_k \in Y_A^*$, there is at most one state s' such that $(s, x_1 \ldots x_k, y_1 \ldots y_k, s') \in h_A$. In fact, an FSM is observable if and only if the underlying automaton that describes the set of FSM traces is deterministic and the latter simplifies a lot when solving FSM analysis and synthesis problems. On the other hand, considering only observable FSMs is not a theoretical limitation, since for every nondeterministic FSM there exists an equivalent observable FSM (by subset construction) [19].

Given an observable FSM $A = (S_A, X_A, Y_A, h_A, s_{A0})$, we define the transition and output functions $\delta_A : S_A \times X_A \times Y_A \to S_A \cup \{\varnothing\}$ and $\lambda_A : S_A \times X_A \to 2^{Y_A}$. For each pair $(s, x) \in S_A \times X_A$ and $y \in Y_A$, by definition, $\delta_A(s, x, y) = s'$ if $(s, x, y, s') \in h_A$; otherwise, $\delta_A(s, x, y) = \varnothing$. Given the transition function $\delta_A : S_A \times X_A \times Y_A \to S_A \cup \{\varnothing\}$, we define an output function $\lambda_A : S_A \times X_A \to 2^{Y_A}$ as the set of possible outputs at state s under input x, i.e.,

$\forall s \in S_A\ \forall x \in X_A\ (\lambda_A(s, x) = \{ y \mid \delta_A(s, x, y) \neq \varnothing \})$.

By definition, for each undefined input x at state s, it holds that $\lambda_A(s, x) = \varnothing$. It can easily be shown that the two definitions of an observable FSM which are based on the transition relation and transition and output functions are equivalent.

Consider FSM $A = (S_A, X_A, Y_A, \delta_A, s_{A0})$ where X_A and Y_A are partitioned into partitions π^{in} and π^{out}. As usual, a class of a partition π^{in} (π^{out}) that contains an input x (an output y) is written as $[x]$ ($[y]$). Classes of π^{in} and π^{out} are called *input and output classes*, correspondingly. FSM A is *quasi-complete* (with respect to the partition π^{in}) if for each state $s \in S_A$ and class $b \in \pi^{in}$, there exists a defined input $x \in b$, i.e.,

$\forall s \in S_A\ \forall b \in \pi^{in}\ \exists x \in b\ (\lambda_A(s, x) \neq \varnothing)$.

FSM A is *quasi deterministic* (w.r.t. the partitions π^{in} and π^{out}) if for each state $s \in S_A$ and class $b \in \pi^{in}$, there exists exactly one defined input $x \in b$ for which $\lambda_A(s, x)$ coincides with some class of π^{out}, i.e., $\forall s \in S_A\ \forall b \in \pi^{in}\ \exists!\ x \in b\ (\lambda_A(s, x) \neq \varnothing\ \&\ \lambda_A(s, x) \in \pi^{out})$.

Here we notice that a deterministic complete FSM is an observable quasi-complete and quasi-deterministic FSM with respect to trivial partitions where each class is a singleton.

For an observable quasi-complete and quasi-deterministic FSM $A = (S_A, X_A, Y_A, \delta_A, s_{A0})$ we define three additional functions related to input and output partitions:

- $\gamma_A : S_A \times \pi^{in} \rightarrow X_A$: for each state $s \in S_A$ and input class p, the function γ_A associates an input $x \in p$ that is defined at state s, i.e.,

  $\forall\, s \in S_A\ \forall\, p \in \pi^{in}\ (\ \gamma_A(s, p) = x \Leftrightarrow \lambda_A(s, x) \neq \varnothing\ )$.

The function γ_A is well defined because there is a unique $x \in p$ such that $\lambda_A(s, x) \neq \varnothing$.

- $\lambda_A^{\wedge} : S \times \pi^{in} \rightarrow \pi^{out}$: for each state $s \in S_A$ and input class p, the function $\lambda_A^{\wedge}$ associates an output class, i.e.,

  $\forall\, s \in S\ \forall\, p \in \pi^{in}\ (\lambda_A^{\wedge}(s, p) = \lambda_A(s, \gamma_A(s, p))$.

Since $\lambda_A(s, \gamma_A(s, p)) = \{y\}$ returns a unique output y, where $y \in q$ and $q \in \pi^{out}$, so q is the output class yielded by $\lambda_A^{\wedge}$.

- $\delta_A^{\wedge} : S_A \times \pi^{in} \times Y_A \rightarrow S_A$: for each state $s \in S_A$, input class p, and output y of the set $\lambda_A^{\wedge}(s, p)$, the function $\delta_A^{\wedge}$ associates the next FSM state, i.e.,

  $\forall\, s \in S_A\ \forall\, p \in \pi^{in}\ \forall\, y \in \lambda_A^{\wedge}(s, p)\ \forall\, s' \in S_A\ (\ \delta_A^{\wedge}(s, p, y) = s' \Leftrightarrow s' = \delta_A(s, \gamma_A(s, p), y)\ )$.

The function $\delta_A^{\wedge}$ is well defined because A is observable and thus, there is a unique $s' = \delta_A(s, \gamma_A(s, p), y)$.

By definition, given functions γ_A, $\lambda_A^{\wedge}$, $\delta_A^{\wedge}$ of an observable quasi-complete and quasi-deterministic FSM, the functions λ_A and δ_A can be derived as follows:

If $x = \gamma_A(s, [x])$ then $\lambda_A(s, x) = \lambda_A^{\wedge}(s, [x])$; otherwise, $\lambda_A(s, x) = \varnothing$.

If $y \in \lambda_A^{\wedge}(s, [x])$ then $\delta_A(s, x, y) = \{\delta_A^{\wedge}(s, [x], y)\}$; otherwise, $\delta_A(s, x, y) = \varnothing$.

B. Finite State Machines with multiple input and output ports

When a synchronous composition is derived using component FSMs, component FSMs may have multiple input and output ports which are connected with those of other components. Given sets X_k, $k \in K$, we define the Cartesian product X of these sets in the usual way:

$$X = \Pi\{\, X_k \mid k \in K \,\} = \{\, x : K \rightarrow \cup\{\, X_k \mid k \in K \,\} \mid x(k) \in X_k, k \in K \,\}.$$

If π_k is a partition of the set X_k, $k \in K$, then the Cartesian product $P = \Pi\{\, \pi_k \mid k \in K \,\}$ of π_k, $k \in K$, induces the Cartesian product π of the Cartesian product X of the sets X_k:

$$\pi = \{\ \{\, x \in X \mid x(k) \in p(k), k \in K \,\} \mid p \in P \,\}.$$

The partition π is called a *component-wise partition* of the set X.

Let FSM A have several input and several output ports. The set of input ports is I, the set of output ports is J, and $I \cap J = \varnothing$. For each input (output) port $i \in I$ ($j \in J$), a corresponding alphabet X_i (Y_j) is specified. Correspondingly, the input alphabet X_A is the Cartesian product of input alphabets X_i, i.e., $X_A = \Pi\{\, X_i \mid i \in I \,\}$, and the output alphabet Y_A is the Cartesian product of output alphabets Y_j, i.e., $Y_A = \Pi\{\, Y_j \mid j \in J \,\}$.

Given $x \in X_A$ and an input port $i \in I$, we use $x(i)$ for denoting an input at the port i while $y(j)$ denotes an output at port j for $y \in Y_A$. Given an input port i, two inputs x and x' are *i-adjacent*, written $x \sim_i x'$, if $x(k) = x'(k)$ for any other input port k, $k \in I\ \&\ k \neq i$.

We now extend the notion of Moore pairs [1, 17, 18] to possibly partial and nondeterministic FSMs. Given an observable FSM A with the set I of input ports and the set J of output ports, $I \cap J = \varnothing$, the pair (i, j), $i \in I$ and $j \in J$, is a *Moore-pair* if the set of outputs at port j does not depend on the input at port i, i.e., $\forall\, s \in S\ \forall\, x \in X\ \forall\, x' \in X$

$$x \sim_i x'\ \&\ \lambda(s, x) \neq \varnothing\ \&\ \lambda(s, x') \neq \varnothing \Rightarrow \lambda(s, x)(j) = \lambda(s, x')(j).$$

If (i, j) is a Moore pair then we write $i \text{ m } j$, otherwise, $i \not\text{m}\, j$.

If A is an observable, quasi-complete and quasi-deterministic FSM, π_k is a partition of the set X_k, $k \in K$, and $P = \Pi\{\, \pi_k \mid k \in K \,\}$ of π_k, $k \in K$, is the Cartesian product of π_k then a Moore-pair can be defined also in another way:

$$\forall\, s \in S\ \forall\, p \in P\ \forall\, p' \in P\ (\forall\, k \neq I\ (p\,(k) = p'\,(k) \Rightarrow \lambda^{\wedge}(s, p)(j) = \lambda^{\wedge}(s, p')(j)).$$

Consider now non-empty subsets $B \subseteq I$ and $C \subseteq J$ of input and output ports of A. Then the *limitation* $X_{A\downarrow B}$ of X_A onto the set B (*limitation* $Y_{A\downarrow C}$ of Y_A onto C) is $\Pi\{\, X_i \mid i \in B \,\}$ ($\Pi\{\, Y_j \mid j \in C \,\}$). The *limitation* $A_{\downarrow B\cup C}$ of the FSM A onto B and C is the FSM $(S_A, X_{A\downarrow B}, Y_{A\downarrow C}, h_{A\downarrow B\cup C}, s_{A0})$ if $h_{A\downarrow B\cup C} = \{(s, x_{\downarrow B}, y_{\downarrow C}, s') \mid (s, x, y, s') \in h_A\}$.

Proposition 1. If A is a complete deterministic FSM then every limitation of A is a complete but possibly nondeterministic FSM.

Proposition 2. If A is a quasi-complete and quasi-deterministic FSM then every limitation of A is quasi-complete but not necessarily quasi-deterministic FSM.

III. SYNCHRONOUS COMPOSITION OF FSMS

Consider a system $\mathcal{A}$ of interacting FSMs such that the sets of states as well the sets of ports of component FSMs are pairwise disjoint. Given an FSM $A \in \mathcal{A}$, we write $A = (S_A, X_A, Y_A, h_A, s_{A0})$. Component FSMs interact with each other and the environment. For this reason, an output port of a component FSM can be connected with an input port of possibly another component FSM; without loss of generality, we assume that any input (output) port is connected with at most one output (input) port. Indeed, if it is not the case, additional repeaters between components should be added. A pair <output-port, input_port> if these ports are connected is a *channel*. By default, we assume that the alphabets of the channel ports coincide. Input (output) ports which are not connected with another port are *external* input (output) ports. External ports are connected with the environment; external inputs are applied at external input ports while external outputs are produced at external output ports.

Correspondingly,

$I = \cup\{\, I_A \mid A \in \mathcal{A} \,\}$ is the union of the sets I_A of input ports over all $A \in \mathcal{A}$;

$J = \cup\{\, J_A \mid A \in \mathcal{A} \,\}$ is the union of the sets J_A of output ports over all $A \in \mathcal{A}$;

$I_e \subseteq I$ is the set of external input ports of the system;

$J_e \subseteq J$ is the set of external output ports of the system;

$S = \Pi\{\, S_A \mid A \in \mathcal{A} \,\}$ is the system state;

$X = \Pi\{ X_i \mid i \in I \}$ is the set of inputs of the system while $X \mid I_e$ limited to the set of external input ports is the set of external inputs of the system;
$Y = \Pi\{ Y_j \mid j \in J \}$ is the set of outputs of the system while $Y \mid J_e$ limited to the set of external output ports is the set of external outputs of the system.
Both sets I_e and J_e of external input and output ports are not empty.

We now define the synchronous composition $\bullet(\mathcal{A})$ with the input ports of the set I_e and output ports of the set J_e. For each component FSM A, the lifting $A\uparrow I\cup J$ is constructed where all the ports of the set $I\cup J$ which are not in the set of ports of A, are added to A. When constructing the lifting we force the same value for all ports which are connected by a channel. The intersection of extended FSMs ($A\uparrow I\cup J$) determines all the sequences which can appear at the system channels. Correspondingly, the external behavior is the intersection limitation on the set $I_e \cup J_e$ of external ports. Three steps can be considered when constructing the synchronous composition $\bullet(\mathcal{A})$.

Algorithm 1 for constructing the synchronous composition
Input: a system $\mathcal{A}$ of interacting FSMs
Output: the synchronous composition $\bullet(\mathcal{A})$
Step 1. Constructing the extension $A\uparrow I\cup J$, $A \in \mathcal{A}$.

The set of input (output) ports of FSM $A\uparrow I\cup J$ is the set $I\cup J$ of all input (output) ports of FSMs $A \in \mathcal{A}$. The behavior relation of A is lifted to added ports as follows. Given the sets B (C) of input (output) ports of FSM A, let $(s_j, x\downarrow B, y\downarrow C, s'_j)$ be a transition of FSM A. Then the lifted FSM $A\uparrow I\cup J$ has the set of transitions $\{(s_j, x, y, s'_j)$: $(s_j, x\downarrow B, y\downarrow C, s'_j) \in h_A\}$ where x and y are the input and output at the ports of the sets I and J. If two ports are connected via a channel in the composition then each transition with different values at these ports is deleted from the behavior relation of $A\uparrow I\cup J$.

By definition of the lifting operator, the following statement holds.

Proposition 3. Given input x (output y) at the ports of the set I and J, FSM $A\uparrow I\cup J$, $A \in \mathcal{A}$, has transition (s_j, x, y, s'_j) if and only if the items of x and y at the ports connected with a channel are the same and FSM A has transition $(s_j, x\downarrow B, y\downarrow C, s'_j)$ where B and C are the sets of input and output ports of FSM A.

Step 2. Constructing FSM $\odot(\mathcal{A}) = \cap \{A\uparrow I\cup J \mid A \in \mathcal{A}\}$.
The lifted component FSMs are intersected and the FSM $\odot(\mathcal{A}) = \cap \{A\uparrow I\cup J \mid A \in \mathcal{A}\}$, is called the *global composition FSM*. Given x (y) at the ports of the set I(J), the FSM has a transition $(s_1\ldots s_n, x, y, s'_1\ldots s'_n)$ if and only if each FSM $A\uparrow I\cup J$ has a transition (s_j, x, y, s'_j), $A \in \mathcal{A}$. Moreover, if two ports are connected via a channel in the composition then each transition with different values at these ports is deleted. Thus, the following statement holds.

Proposition 4. Given input x (output y) at the ports of the set I(J), the FSM $\odot(\mathcal{A}) = \cap A\uparrow I\cup J$, $A \in \mathcal{A}$, has a transition $(s_1\ldots s_n, x, y, s'_1\ldots s'_n)$ if and only if each FSM A, $A \in \mathcal{A}$, has a transition $(s_j, x\downarrow B, y\downarrow C, s'_j)$ where B and C are the sets of input and output ports of FSM A, $A \in \mathcal{A}$, and the values of x and y at the ports connected with a channel coincide.

Step 3. Constructing FSM $\bullet(\mathcal{A})$.

The limitation of the FSM $\odot(\mathcal{A})$ onto external input and output ports is derived. The FSM $\bullet(\mathcal{A})$ is *the synchronous composition* of FSMs A, $A \in \mathcal{A}$.

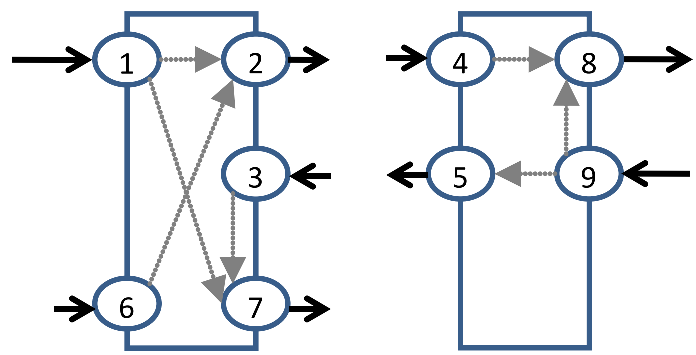
Due to the definition of the limitation operator, the following statement holds.

Proposition 5. The behavior of FSM $\bullet(\mathcal{A})$ at state $s_1\ldots s_n$ is defined under input $x\downarrow I_e$ if and only if the global composition FSM $\odot(\mathcal{A})$ has a transition $(s_1\ldots s_n, x, y, s'_1\ldots s'_n)$.

If the synchronous composition of two component FSMs is considered then the FSM $\bullet(\mathcal{A})$ is equivalent to the FSM which is derived by composing FSM languages first and then coming back to FSMs [12, 16].

Example 1. Consider FSMs S and P in Figs. 1a and 1b with transition relations in Table 1 where port numbers are shown in bold.

The FSM S has three input ports and two output ports. All the ports have the same alphabet {0, 1}. The output signal $2 = 1 \oplus 6$, i.e., the output signal 2 does not depend on the input signal at port 3; therefore, the pair (3, 2) is a Moore pair. The output signal $7 = \neg 1 \vee 3$, i.e., the output signal 7 does not depend on the input signal at port 6; therefore, the pair (6,7) is also a Moore pair. In the FSM P, $8 = \neg 4 \oplus 9$, $5 = 9$, i.e., (4, 5) is a Moore pair in this machine. Dotted lines in Fig. 1 show the dependency between signals at input and output ports.



a) b)

Fig 1. FSM S (a) and (b) FSM P

Table 1 – Transitions relations of FSMs S(a) and P(b)

1	**3**	**6**	**2**	**7**
0	0	0	0	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	1

(a)

4	**9**	**5**	**8**
0	0	0	1
0	1	1	0
1	0	0	0
1	1	1	1

(b)

Consider the system of interacting FSMs in Figure 2 that has external input ports 1 and 9 and one external output port 8 and construct the synchronous composition $S \bullet P$.

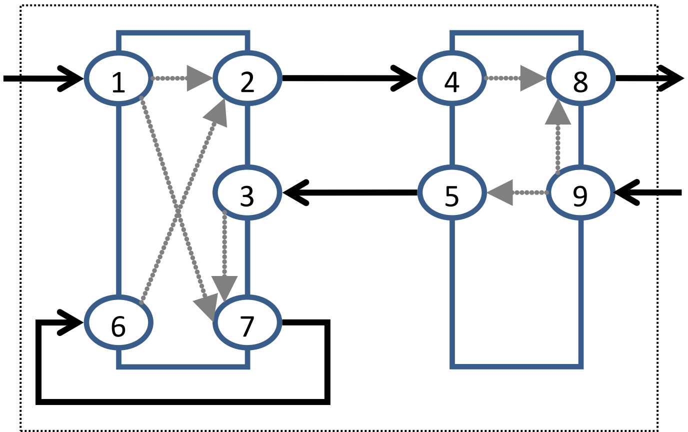


Fig. 2. The system of interacting FSMs *S* and *P* that has external input ports 1 and 9 and one external output port 8

Table 2 – Transitions relations of FSM *S* (a) lifted to ports 4, 5, 8 and 9 and *P*(b) lifted to ports 1, 2, 3, 6, 7

(a)

1	**3**	**6**	**2**	**7**	**4**	**9**	**5**	**8**
0	0	1	1	1	1	**0**	0	**0**
0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	0	0	1
0	0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	0	1	1
0	1	1	1	1	1	**1**	1	**1**
1	0	0	1	0	1	**0**	0	**0**
1	0	0	1	0	1	1	0	0
1	0	0	1	0	1	0	0	1
1	0	0	1	0	1	1	0	1
1	1	1	0	1	0	0	1	0
1	1	1	0	1	0	**1**	1	**0**
1	1	1	0	1	0	0	1	1
1	1	1	0	1	0	1	1	1

(b)

1	**3**	**6**	**2**	**7**	**4**	**9**	**5**	**8**
0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	1	1	0
0	0	0	1	0	1	0	0	0
0	1	0	1	0	1	1	1	1
0	0	1	0	1	0	0	0	1
0	1	1	0	1	0	1	1	0
0	0	1	1	1	1	**0**	0	**0**
0	1	1	1	1	1	**1**	1	**1**
1	0	0	0	0	0	0	0	1
1	1	0	0	0	0	1	1	0
1	0	0	1	0	1	**0**	0	**0**
1	1	0	1	0	1	1	1	1
1	0	1	0	1	0	0	0	1
1	1	1	0	1	0	**1**	1	**0**
1	0	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1

Lift the transition relation of *S* to ports 4, 5, 8 and 9 and leave only vectors which have the same value for ports 2 and 4, 3 and 5, 6 and 7. The transition relations of lifted FSMs are shown in Table 2. Lift the transition relation of *P* to ports 1, 2, 3, 6, 7 and leave only vectors which have the same value for ports 2 and 4, 3 and 5, 6 and 7; in Table 2 those vectors are not greyed. The intersection of lifted FSMs has four vectors shown in bold in Table 2. Its limitation to external input ports 1 and 9 and external output port 8 provides the transition relation {0 0 0, 0 1 1, 1 0 0, 1 1 0}. We notice that the obtained composition of two deterministic FSMs is complete and deterministic since each cycle in the composition has a Moore pair. This feature is proven as the corollary to Theorem 10. Here we notice that if all the ports have the alphabet {0, 1} and states are represented as Boolean vectors, it would be interesting to get more scalable formulas for obtaining the composition transition relation but this needs more research.

IV. Synchronous composition of observable quasi-complete and quasi-deterministic FSMs

In general, the behavior of a system is described by an incomplete FSM, i.e., FSM components of a synchronous composition can be incomplete. In order to escape deadlocks which can occur according to such incompleteness, restrictions on component FSMs are required. In this section, we consider a special class of quasi-complete and quasi-deterministic FSMs and show that the synchronous composition of observable quasi-complete and quasi-deterministic component FSMs is observable quasi-complete and quasi deterministic if every cycle in the global FSM $\odot(\mathcal{A})$ has a Moore pair. As a deterministic complete FSM is a particular case of an observable quasi-complete and quasi deterministic FSM, the synchronous composition of deterministic complete FSMs is a deterministic complete FSM if every cycle in the global FSM $\odot(\mathcal{A})$ has a Moore pair.

A. *The dependency graph*

Consider the synchronous composition of observable quasi-complete and quasi deterministic FSMs *A*, $A \in \mathcal{A}$, and the global composition FSM $S = \odot(\mathcal{A})$ that has the sets *I* and *J* of input and output ports. Every Moore pair of each component FSM *A*, $A \in \mathcal{A}$, is a Moore pair of the FSM $S = \odot(\mathcal{A})$. A sequence of pairs <input_port, output_port> of component FSMs is a *path* in the FSM *S* if starting from the second pair the input port of each pair is connected with the output port of the previous pair. As usual, a path is a *cycle* if the output port of the tail pair is connected with the input port of the head pair.

In this paper, we establish the following sufficient condition: if the composition has the property that every cycle of ports has a component with a Moore pair then the FSM $\bullet(\mathcal{A})$ also is an observable quasi-complete and quasi deterministic FSM. In order to prove this condition, given FSM $\odot(\mathcal{A})$, we construct a special dependency graph G_{dep}. The nodes of the graph are all the ports of the FSM $\odot(\mathcal{A})$ and for each component FSM A_k, $A_k \in \mathcal{A}$, there are two special nodes. The node labeled with s_k corresponds to a current state of the component FSM A_k, while the node labeled with s'_k corresponds to the next state of the component FSM A_k, $A_k \in \mathcal{A}$.

The graph has the following edges.

1) If a pair <output_port, input_port> is connected by a channel, then there is an edge from the output port to the input port in the graph.

2) If a pair <input_port, output_port> belongs to some component FSM A_k, $A_k \in \mathcal{A}$, and is not a Moore pair in *A*, then there is an edge from the input port to the output port in the graph; there also is an edge from the input port to the node s'_k.

3) Given the node s_k that corresponds to the current state of A_k, the graph has an edge from the node s_k to the node s'_k and each output port of A_k, $A_k \in \mathcal{A}$. A node s'_k, $A_k \in \mathcal{A}$, has no outgoing edges.

We now show that if every cycle in the FSM $\odot(\mathcal{A})$ has a Moore pair then the graph G_{dep} is acyclic and this allows us to introduce special composition functions such that the output function for the output class is uniquely defined by the component FSM states and external input values.

Proposition 6. The graph G_{dep} is an acyclic graph if every cycle in the FSM $\odot(\mathcal{A})$ has a Moore pair.

Proof. Assume that the graph is not acyclic while every cycle in the FSM $\odot(\mathcal{A})$ has a Moore pair. By construction, there are no incoming edges in the nodes of G_{dep} which correspond to input ports of the FSM $\odot(\mathcal{A})$ and current states of the component FSMs. On the other hand, there are no outgoing edges in the nodes of G_{dep} which correspond to output ports of the FSM $\odot(\mathcal{A})$ and next states of the component FSMs. Correspondingly, every cycle in the graph does not traverse nodes-states and is an interchanging sequence of pairs <output_port, input_port> connected by a channel and pairs <input_port, output_port > of some component FSM, i.e., is a cycle in the FSM $\odot(\mathcal{A})$. Since by construction graph G_{dep} has no Moore pairs, this contradicts the hypothesis.

□

As usual, the nodes of an acyclic graph can be ranked starting from nodes corresponded to external inputs and nodes s_k corresponded to a current state of A_k, $A_k \in \mathcal{A}$. This ranking is used in the next section.

B. A closure condition on the synchronous composition of observable quasi-complete and quasi-deterministic FSMs

Consider now a system $\mathcal{A}$ of interacting FSMs where each component FSM $A \in \mathcal{A}$ is observable quasi-complete and quasi-deterministic. In order to show that the synchronous composition also possesses these features we first prove three statements. In this section, we use additional notations for the case when all the component FSMs are observable quasi-complete and quasi-deterministic. We denote by $P = \Pi\{ \pi_i \mid i \in I \}$ the set of input classes of the FSM $\odot(\mathcal{A})$ and by $P_e = \{ p \mid I_e \mid p \in P \}$ the set of external input classes. In the same way, $Q = \Pi\{ \pi_j \mid j \in J \}$ is the set of output classes of the FSM $\odot(\mathcal{A})$, and $Q_e = \{ q \mid J_e \mid q \in Q \}$ is the set of external output classes. Given a port $m \in I \cup J \setminus I_e \setminus J_e$, $r(m)$ is the port connected with port m, and $r(M)$, $M \subseteq I \cup J \setminus I_e \setminus J_e$, is the set of such ports $\{ r(m) \mid m \in M \}$, where $M \subseteq I \cup J \setminus I_e \setminus J_e$. We also denote $a(m)$ an FSM $A \in \mathcal{A}$ with the port $m \in I \cup J$.

Given $p_e \in P_e$ and $s \in S$, the following system (1) of equations with respect to variables $p(i)$, $i \in I$, and $q(j)$, $j \in J$, is considered:

$$\begin{cases} p(i) = p_e(i),\ i \in I_e, \\ p(i) = q(r(j)),\ i \in I \setminus I_e, \\ q(j) = \hat{\lambda}_{a(j)}(s(a(j)), p \mid I(a(j)))(j). \end{cases} \quad (1)$$

The second equation specifies that for two ports connected by a channel, the same classes are associated. The third equation specifies the class $q(j) = \hat{\lambda}_{a(j)}(s(a(j)), p \mid I(j))(j)$ that is the result of the function $\hat{\lambda}_{a(j)}$ for the component FSM $a(j)$ at the port j where p is the class value at the input ports of FSM $a(j)$.

In fact, a solution to this system associates with each output port j a class $q(j)$ that is specified by the function $\hat{\lambda}_{a(j)}$ of FSM $a(j)$ at state $s(a(j))$ for input class $p{\downarrow}I(j)$ limited to the set $I(j)$ of input ports of the FSM which are not Moore pairs with port j.

Proposition 7. Given $p_e \in P_e$ and $s \in S$, the system (1) of equations with respect to variables $p(i)$, $i \in I$, and $q(j)$, $j \in J$, is solvable and has a unique solution such that

$q(j) = \hat{\Lambda}_j(s, p_e)$, $j \in J$,
$p(i) = p_e(i)$, $i \in I_e$,
$p(i) = \hat{\Lambda}_{r(i)}(s, p_e)$, $i \in I \setminus I_e$

In other words, for each output port j, $j \in J$, $q(j)$ is computed by a well-defined function $\hat{\Lambda}_j : S \times P_e \to \pi_j$ that depends on the state s of the system and classes p_e which are associated with external input ports.

Proof. As usual, we say that a node of an acyclic graph G_{dep} is ranked by k, if k is the maximum distance from this node to nodes without incoming nodes. For those nodes, $k = 0$. Moreover, if G_{dep} has at least one edge, then the graph has a node of rank 1. It is well known that all the incoming edges of a node of rank $k > 0$ have starting nodes with rank less than k. Due to the graph definition, a node has rank 0 if this node is an external input port or this node corresponds to a current state s_k of a component FSM $A_k \in \mathcal{A}$.

For proving the statement, we use the induction on port ranks of the graph G_{dep}. Given an external input port $i \in I_e$, the class $p(i) = p_e(i)$ is associated with the port, and thus, the *induction base* holds.

Induction assumption. We now assume that the statement holds for all ports of rank at most $n > 0$ and prove the statement for the port of rank $(n + 1)$ (if there is such a port).

If n is odd then each port i of rank $(n + 1)$ is an input port that has a single incoming edge from the port $r(i)$ of smaller rank that is connected with this port by a channel. For each such port we have $p(i) = q(r(i))$.

If n is even then each port j of rank $(n + 1)$ is an output port that has incoming edges from input ports of this component FSM $a(j)$ and all these ports have ranks $\leq n$. We associate to each port j the class $q(j) = \hat{\lambda}_{a(j)}(s(j), p \mid I(j))(j)$. For each input port of the FSM $a(j)$ it holds that $\forall\ i \in I(j) \setminus I_e\ (p(i) = q(r(i))\ )$.

All the ports of the set $r(I(j) \setminus I_e)$ have rank at most n, since there is an incoming edge from such a port to the port j in the graph. Thus, due to the induction assumption, for each $i \in I(j) \setminus I_e$ it holds that $q(r(i)) = \hat{\Lambda}_{r(i)}(s, p_e)$.

Given $\hat{\lambda}_{a(j)}(s(j), p \mid I(j))(j)$, we consider each pair $(i, p(i)) \in p \downarrow I(j))$ and replace $p(i)$ by $\hat{\Lambda}_{r(i)}(s, p_e)$. As a result, we obtain that for each output port j of rank $(n + 1)$ the class $q(j)$ is computed by a well-defined function $\hat{\Lambda}_j(s, p_e)$.

□

Consider the system (2) of equations over $x(i)$, $i \in I$, and $y(j), j \in J \setminus J_e$:

$$\begin{cases} x(i) = \gamma_{a(i)}(s(a(i)), p \mid I_{a(i)})(i)\ ,\ i \in I, \\ y(j) = x(r(j)), j \in J \setminus J_e. \end{cases} \quad (2)$$

The first equation states that $x(i) = \gamma_{a(i)}(s(a(i)), p \mid I_{a(i)})(i)$, $i \in I$, that is we associate to each input port i a function $x(i)$, where x is specified by the function $\gamma_{a(i)}$ of FSM $a(i)$ at state $s(a(i))$ for the input class $p \mid I_{a(i)}$ and $x(i) \in p(i)$. The second equation states that for two ports connected with a channel, the same values are associated.

Proposition 8. System (2) of equations is solvable with respect to $x(i)$, $i \in I$, and $y(j)$, $j \in J \setminus J_e$: it has a unique solution $x(i) = \Gamma_i(s, p_e)$, $i \in I$, where $\Gamma_i : S \times P_e \to X_i$ is a function that depends on the state s of the system and classes p_e associated with external input ports while $y(j) = x(r(j))$.

Proof. We consider the function $x(i)$ over the set I of input ports: $\forall\ i \in I\ (x(i) = \gamma_{a(i)}(s(a(i)), p \mid I_{a(i)})(i))$.

If $i \in I_{a(i)} \cap I_e$, i.e., i is an external port, then by hypothesis, it holds that $p(i) = p_e(i)$. If $i \in I_{a(i)} \setminus I_e$ then $p(i) = q(r(i))$ where according to Proposition 7, $q(r(i)) = \Lambda^{\wedge}_{r(i)}(s, p_e)$.

Given $\gamma_{a(i)}(s(a(i)), p \mid I_{a(i)})(i)$, we consider each pair $(i, p(i)) \in p{\downarrow}I_{a(i)}$ and replace $p(i)$ by $\Lambda^{\wedge}_{r(i)}(s, p_e)$. As a result, we obtain that $x(i)$ is defined by a well-defined function $\Gamma_i(s, p_e)$.

For each non-external output port, according to the statement conditions, it holds that $y(j) = x(r(j))$.

□

Let all the functions associated with ports of the graph G_{dep} be solutions to systems (1) and (2) of equations and $y_e(j) \in q(j)$, $j \in J_e$. Consider the system (3) of equations over $y(j)$, $j \in J_e$, and $s'(A)$, $A \in \mathcal{A}$, where $s'(A)$ is the next state of the component FSM A:

$$\begin{cases} y(j) = y_e(j), y_e(j) \in q(j), j \in J_e, \\ s'(A) = \delta^{\wedge}_A(s(A), p \mid I_A, y \mid J_A). \end{cases} \quad (3)$$

Proposition 9. System (3) of equations is solvable with respect to $y(j)$, $j \in J_e$, and $s'(A)$, $A \in \mathcal{A}$: it has a unique solution $s'(A) = \Delta^{\wedge}_A(s, p_e, y \mid J_e)$, where $\Delta^{\wedge}_A : S \times P_e \times Y_e \to S_A$ is a function that depends on the state s of the system, external input classes p_e and outputs y_e, which are associated with external output ports.

Proof. According to Proposition 8, we associate to each external output port j the class $y(j) = y_e(j)$. For each component FSM $A \in \mathcal{A}$, the next state $s'(A)$ is equal to $\delta^{\wedge}_A(s(A), p \mid I_A, y \mid J_A)$, where $y \downarrow J_A \in \lambda^{\wedge}_A(s(A), p \downarrow I_A)$. According to the system of equations (1), $p(i) = p_e(i)$, $i \in I_A \cap I_e$, and $p(i) = q(r(i))$, $i \in I_A \setminus I_e$. By Proposition 7, $q(r(i)) = \Lambda^{\wedge}_{r(i)}(s, p_e)$.

For an external output port j, $y(j) = y_e(j)$, $j \in J_A \cap J_e$, and $y(j) \in q(j)$. For a non-external output port j, according to Proposition 8, $y(j) = x(r(j)) = \Gamma_{r(j)}(s, p_e)$.

Given $\Lambda^{\wedge}_{r(i)}(s, p_e)$, we consider each pair $(i, p(i)) \in p \mid I_A$ and replace $p(i)$ by $\Lambda^{\wedge}_{r(i)}(s, p_e)$, while for each pair $(j, y(j))$ output $y(j)$ is replaced by $\Gamma_{r(i)}(s, p_e)$.

As a result, we obtain that $s'(A)$ is computed by a well-defined function $\Delta^{\wedge}_A(s, p_e, y_e)$.

□

Propositions 7 – 9 imply the following:

1) Inputs x at external input ports and classes $q \mid J_e$ of external output ports are uniquely defined by state s of the FSM $\odot(\mathcal{A})$ and external input port classes $p \mid I_e$;

2) The next state s' of the system is uniquely defined by state s of the FSM $\odot(\mathcal{A})$, external input port classes $p \downarrow I_e$, and outputs $y \downarrow J_e$ at external output ports of the classes $q \downarrow J_e$.

The latter means as follows:

1) There exists the function $\Gamma : S \times P \downarrow I_e \to X \downarrow I_e$ such that $\forall\ s \in S\ \forall\ p \in P \downarrow I_e$
$\Gamma(s, p) = \{\ (i, \Gamma_i(s, p)) \mid i \in J_e\ \}$;
i.e., the function Γ is complete and unique, along with $\Gamma(s, p)$ being within the corresponding class.

2) There exists the function $\Lambda^{\wedge} : S \times P \mid I_e \to Q \mid I_e$ such that $\forall\ s \in S\ \forall\ p \in P \downarrow I_e$
$\Lambda^{\wedge}(s, p) = \{\ (j, \Lambda^{\wedge}_j(s, p)) \mid j \in J_e\ \}$;
i.e., the function $\Lambda^{\wedge}$ is complete and unique, along with $\Lambda^{\wedge}(s, p)$ being within the corresponding class.

3) There exists the function $\Delta^{\wedge} : S \times P \mid I_e \times Y \mid J_e \to S$ such that $\forall\ s \in S\ \forall\ p \in P \downarrow I_e$
$\forall\ y \in \Lambda^{\wedge}(s, p)_{\downarrow}\ \Delta^{\wedge}(s, p) = \{\ (A, \Delta^{\wedge}_A(s, p)) \mid A \in \mathcal{A}\ \}$,
i.e., the function $\Delta^{\wedge}$ is complete and unique and, for each y, it is within the class $\Lambda^{\wedge}(s, p)$.

As the above functions correspond to classes over external input and output ports, the set of observable quasi-complete and quasi-deterministic component FSMs is closed under the synchronous composition when every cycle in the global FSM $\odot(\mathcal{A})$ has a Moore pair, i.e., the following statement holds.

Theorem 10. The synchronous composition of observable quasi-complete and quasi-deterministic component FSMs is observable, quasi-complete and quasi deterministic if every cycle in the global FSM $\odot(\mathcal{A})$ has a Moore pair.

For deterministic complete FSMs, corresponding partitions are trivial containing only singletons and thus, the following statement holds as a corollary to Theorem 10.

Corollary. The synchronous composition of complete and deterministic component FSMs is complete and deterministic if every cycle in the global FSM $\odot(\mathcal{A})$ has a Moore pair.

The corollary to Theorem 10 generalizes most of the existing results when the synchronous composition of complete and deterministic FSMs is a complete deterministic FSM. However, this requirement is only a sufficient condition, since in [12] there are examples when the condition does not hold but the synchronous composition of complete and deterministic FSMs is a complete deterministic FSM.

V. Conclusions and Future Work

In this paper, we investigate multi component synchronous composition of FSMs; we extended the existing synchronous composition operator from a pair of components to a collection of components with multiple input and output ports, and provided a procedure to compute the composition directly from the transition tables of the component FSMs. More research is needed to evaluate the complexity of the proposed procedure; however, it seems that a scalable FSM representation by logic circuits can be used when each port alphabet is Boolean. Another interesting question is whether Algorithm 1 can be applied also to more expressive models like timed FSMs and other extensions.

Furthermore, in this paper we study the synchronous composition of quasi-complete and quasi-deterministic FSMs, a subclass of partial and nondeterministic FSMs, whereas most previous investigations were restricted to complete and deterministic FSMs. In this paper, we consider multi component compositions where each component FSM is an observable quasi-deterministic and quasi-complete FSM and can have multiple input and output ports; the synchronous composition operator is defined over such FSMs. We then prove that the set of observable quasi-deterministic and quasi-complete FSMs is closed under synchronous composition if every cycle over component ports has an input-output Moore pair (the output is insensitive to the input). Future research includes studying the problem of finding the unknown component over quasi-complete and quasi-deterministic FSMs.

ACKNOWLEDGEMENT

The authors of the Institute for System Programming of RAS appreciate the support of the Ministry of Science and Higher Education of the Russian Federation; Tiziano Villa was partially supported by MIUR, Project "Italian Outstanding Departments, 2018-2022" and by INDAM, GNCS 2019, "Formal Methods for Mixed Verification Techniques".

REFERENCES

[1] Hartmanis, J., Stearns, R.E. The algebraic structure theory of sequential machines. Prentice-Hall (1966)

[2] P. Merlin and G. v. Bochmann. On the construction of submodule specifications and communication protocols. ACM Transactions on Programming Languages and Systems, 5(1): 1–25 (1983)

[3] A. Arnold. Finite transition systems : semantics of communicating systems. Prentice-Hall (1994)

[4] C. C. Cassandras and S. Lafortune. Introduction to discrete event systems. Kluwer Academic Publishers (1999)

[5] J. Peng and J. Hammer. Input/output control of asynchronous sequential machines with races. International Journal of Control, 83(1): 125–144 (2010)

[6] H. Qin and P. Lewis. Factorization of finite state machines under strong and observational equivalences. Formal Aspects of Computing, 3: 284–307 (1991)

[7] P. Ramadge and W.Wonham. The control of discrete event systems. Proceedings of the IEEE, 77(1): 81–98 (1989)

[8] Elizabeth Wolf. Hierarchical Models of Synchronous Circuits for Formal Verification and Substitution. PhD thesis, Stanford University (1995)

[9] G. T. Heineman and W. T. Councill, Eds., Component-based Software Engineering: Putting the Pieces Together. Addison-Wesley Longman Publishing Co. (2001)

[10] R. Kurshan, M. Merritt, A. Orda, and S. Sachs. Modelling asynchrony with a synchronous model. Formal Methods in System Design, 15 (3): 175–199 (1999)

[11] Kam, T., Villa, T., Brayton, K. R., Sangiovanni-Vincentelli, A. Synthesis of FSMs: Functional Optimization. Springer (1997)

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

[13] P. Bouyer and A. Petit. Decomposition and composition of timed automata. International Colloquium on Automata, Languages, and Programming, ICALP: 210-219 (1999)

[14] Thomas A. Henzinger. The Theory of Hybrid Automata. 11th Annual IEEE Symposium on Logic in Computer Science: 278-292 (1996)

[15] Gill, A. Introduction to the Theory of Finite-State Machines. NY (1964)

[16] N. Yevtushenko, T. Villa, and S. Zharikova. Solving language equations over synchronous and parallel composition operators. The 1st International Workshop on Theory and Applications of Language Equations: 14-32 (2007)

[17] Burdonov I., Kossatchev A. Testirovanie sistemy avtomatov. Vestnik Tomskogo universiteta. Upravlenie, vychislitelnaia tekhnika i informatika: 63-71 (2017)

[18] I. Burdonov, A. Kossatchev. Obobshchennaia model sistemy avtomatov. Vestnik Tomskogo universiteta. Upravlenie, vychislitelnaia tekhnika i informatika: 89-97 (2017)

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