Deriving a Module of a Multi Agent System via Finite State Machine Equation Solving

Ekaterina Darusenkova, Nina Yevtushenko
Tomsk State University
Tomsk, Russia
darusenkova@gmail.com, yevtushenko@sibmail.com

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

***Abstract*— In multiagent systems, there is a problem of constructing an agent that can work in different contexts satisfying different specifications. One of ways is to solve a system of corresponding automata equations. Since in general, the complexity of solving such equations is exponential w.r.t. to the number of states of the context and specification, the question arises whether a system of equations can be reduced to a single equation. In this paper, we consider two special cases when a system of equations under the parallel composition over Finite State Machines can be reduced to a single equation. For each case, it is shown how a corresponding single equation can be derived.**

***Keywords*— multiagent systems; Finite State Machine (FSM); Parallel Composition of FSMs; FSM equations**

I. Introduction

When constructing a multiagent system there is a problem of designing an agent that can work in different contexts facing different specifications [1]. The problem about the permissible changes in the component (agent) behavior arises in many other applications, such as optimization of digital circuits, cryptography, logic synthesis, logic games, stabilizers for the synthesis of asynchronous systems, etc. The problem is known as the equation solving problem [2], submodule construction [3] or the unknown component problem [4]. If the behavior of all systems is described by sets of sequences of actions then the problem can be solved using systems of appropriate equations. In [1, 4], an approach is proposed for solving a system of equations over Finite State Machines (FSMs), i.e., over finite automata where actions are divided into inputs and outputs and each input is followed by an output. The proposed approach [1, 4] is to solve equations separately and then derive the product of the largest solutions. However, the complexity of solving a single equation over FSMs is known to have the exponential complexity with respect to the size of the context and specification and for this reason, it would be nice to check whether the problem of solving a system of FSM equations can be reduced to solving a single equation.

Usually the interaction between the components is described by using the operator of parallel or synchronous composition, and thus we have a system of parallel or synchronous equations [2]. In the case of the synchronous composition that is used in hardware, all components are active at each time instance; the composition produces an external output and a matched pair of internal signals. The parallel composition is based on the dialogue between components, only one component is active at each time instance and an external output is produced only after finishing the internal dialogue between components. Such compositions are widely used in telecommunication systems. For multiagent systems, the parallel composition also seems to be more appropriate.

In this paper, we consider a system of finite number of equations under the parallel composition over FSMs. We show that there exist two special cases when we can reduce the system of FSM equations to a single FSM equation. Namely, it happens when all contexts or all specifications coincide.

The rest of the paper is structured as follows. In Section 2, we define basic notions and definitions for solving parallel FSM equations. Section 3 contains the problem statement and a proposed approach for reducing a system of equations to a single equation. The section also contains a short discussion about a general case. Section 4 concludes the paper.

II. Preliminaries

A. *Finite State Machine*

Finite State Machine (FSM) [5] is a quintuple $\mathbb{A} = (A, I, O, T_{\mathbb{A}}, a_0)$, where A is a finite nonempty set of states with the initial state a_0, I and O are input and output alphabets, and $T_{\mathbb{A}} \subseteq I \times A \times A \times O$ is a transition relation. An FSM $\mathbb{A}$ is *observable*, if for each triple $(a, i, o) \in A \times I \times O$ there exists at most one state $a' \in A$ such that $(a, i, o, a') \in T_{\mathbb{A}}$. An FSM $\mathbb{A}$ is *complete*, if $\forall a \in A$ and $\forall i \in I$ there exist $o \in O$ and $a' \in A$, such that $(a, i, o, a') \in T_{\mathbb{A}}$. If $\mathbb{A}$ is not complete, then it is called *partial*. An FSM $\mathbb{A}$ is *deterministic*, if $\forall a \in A$ and $\forall i \in I$ there exist at most one pair of output o and state a', such that $(a, i, o, a') \in T_{\mathbb{A}}$; an FSM $\mathbb{A}$ is *nondeterministic*, if there exist $a \in A$ and $i \in I$ such that there are two different transitions (a, i, o, a'), $(a, i, o', a'') \in T_{\mathbb{A}}$. When solving FSM equations the notion of a nondeterministic FSM allows to describe a general solution to a solvable equation. An FSM $\mathbb{B} = (B, I, O, T_{\mathbb{B}}, b_0)$ is a *sub-machine* of $\mathbb{A}$ if $B \subseteq A$ and $T_{\mathbb{B}} \subseteq T_{\mathbb{A}}$. The largest complete submachine of FSM $\mathbb{A}$ can be obtained by iterative deleting states where the behavior of the FSM is not defined at

978-1-4799-7103-9/15/$31.00 ©2015 IEEE

least for a single input. If the initial state is deleted then FSM A has no complete machine. Otherwise, the remained complete FSM is the largest submachine of A. Each complete submachine of A is a submachine of the largest complete submachine of A (if it exists). As usual, the transition relation T_{A} of FSM A can be extended to sequences over the alphabets I and O. A trace of the FSM is a sequence of pairs (i, o) which correspond to consecutive transitions. A trace defines an output response of the FSM to its input projection and thus, an FSM can be considered as a sequential function that maps the infinite set of input sequences to the sets of output sequences. If each state of an FSM is reachable from the initial state via some trace, then an FSM is called *initially connected* or simply *connected*. FSMs are widely used in order to describe the behavior of reactive systems which get some messages (inputs) and produce a corresponding response (output) [2]. States of the FSM correspond to the memory about previously applied inputs and produced outputs. For example, in Fig. 2 an FSM with the initial state a is presented. After applying an input i_1 at the initial state a the FSM produces output u_2 and moves to the next state b; thus, when the next input i_1 is applied the FSM produces output o_2 and remains at state b.

Given an FSM A, the set of all traces at state a of A is called the *language* of A *at state a*, written $L_A(a)$. The language of the FSM A at the initial state is called the language of the FSM A and is denoted by L_{A}, for short. The FSM $(\{t_0\}, I, O, T, t_0)$ where $T = \{t_0\} \times I \times O \times \{t_0\}$, written MAX($I$, O), is called the *maximum* FSM over the input alphabet I and the output alphabet O. The machine MAX(I, O) has the language $(IO)^*$. An FSM B is a *reduction* of FSM A, written A $\leq$ B, if $L_{\mathrm{B}} \subseteq L_{\mathrm{A}}$. If $L_{\mathrm{B}} = L_{\mathrm{A}}$ then FSMs A and B are *equivalent*. For complete deterministic FSMs the reduction and the equivalence relations coincide.

The common behavior of two FSMs can be described by the intersection of these machines. The *intersection (or a product)* A $\cap$ B of FSMs A and B is the largest connected submachine of the FSM $(A \times B, I, O, T_{\mathrm{A} \cap \mathrm{B}}, a_0b_0)$. Formally, $T_{\mathrm{A} \cap \mathrm{B}} = \{(ab, i, o, a'b') \mid (a, i, o, a') \in T_{\mathrm{A}} \wedge (b, i, o, b') \in T_{\mathrm{B}}\}$. The language of A $\cap$ B is the intersection $L_{\mathrm{A}} \cap L_{\mathrm{B}}$. The intersection of two observable FSMs is an observable FSM; however, the intersection of complete FSMs can be partial.

B. *Finite Automaton*

When combining FSMs, all the operators are defined over finite automata. Finite automata are very close to FSMs: there is the non-empty finite set of states with the designated initial state and the designated subset of final states which correspond to finishing some job and the non-empty finite set of actions that are not divided into inputs and outputs. Given an FSM A, the corresponding finite automaton *Aut*(A) is derived by unfolding FSM transitions. Final states are states that have an incoming transition labeled with an output and the initial state. In Fig. 3, an automaton for the FSM in Fig. 2 is shown.

Given a sequence $\alpha \in V^*$ and an alphabet W, a *W-restriction* of α, written $\alpha_{\downarrow W}$, is obtained by deleting from α all symbols that belong to the set $V \setminus W$. Given a sequence $\alpha \in V^*$ and an alphabet W, a *W-expansion* of α, written $\alpha_{\uparrow W}$, is a set that contains each sequence over alphabet $(V \cup W)$ with the V–restriction α. All the operators are extended to operators over finite automata. Let P be an automaton over alphabet V with the language L. *Restriction* ($\downarrow$): Given a non-empty subset U of V, the automaton $\mathrm{P}_{\downarrow U}$ that accepts the language $L_{\downarrow U}$ over U is obtained by replacing each transition (s, a, s'), $a \in V \setminus U$, in P by the transition (s, ε, s'). *Expansion* ($\uparrow$): Given alphabet U, the automaton $\mathrm{P}_{\uparrow U}$ with the language $L_{\uparrow U}$ over $U \cup V$ is obtained by adding a transition (s, a, s) for each $a \in U \setminus V$ and each state s of P.

C. *Parallel Composition of FSMs*

Let C = $(C, I \cup V, O \cup U, T_{\mathrm{C}}, c_0)$ and X = (X, U, V, T_X, x_0) be two complete communicating FSMs where alphabets I, V, O, U are pair-wise disjoint. The alphabet I represents the set of external inputs of the composition, while the alphabet O represents the set of external outputs of the composition. The embedded component X corresponds to an agent to be designed.

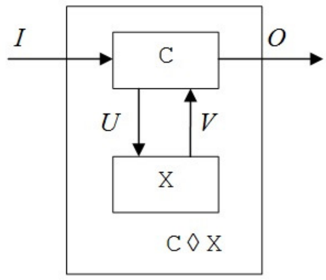

Fig. 1. The parallel composition of FSMs C and X

The two FSMs communicate under a single message in transit, i.e., the next external input is submitted to the system only after producing an external output to the previous input (so-called "slow" environment). Under these conditions, the collective behavior of the two communicating FSMs can be described by an FSM. When applying an external input to the FSM component C, the FSM produces an internal or an external output. If the C produces an internal signal, components start their dialogue that should be finished by producing an external output by the component C. If the dialogue becomes infinite then the composition is said to *fall into a livelock*. If an external output is produced then the composition waits for the next external input.

The *parallel composition of FSMs* C and X [1], denoted C ◊ X, can be formally described using automata for FSM components: First, for FSMs C and X, the corresponding automata *Aut*(C) and *Aut*(X) are derived and the automaton *Aut*(X) is expanded to external inputs and outputs. At the next step, the intersection $(Aut(\mathrm{C}) \cap Aut(\mathrm{X})_{\uparrow I \cup O})_{\downarrow I \cup O}$ is restricted to the set of external inputs and outputs and the restriction is intersected with the automaton *Aut*(MAX(I, O)) that models the slow environment. The obtained automaton is converted into an FSM by combining inputs with following outputs. It is known that the parallel composition of two complete FSMs can be partial if the communicating FSMs fall into a live-lock. In this case, the I-restriction of $(Aut(\mathtt{C}) \cap Aut(X)_{\uparrow I \cup O})_{\downarrow I \cup O} \cap Aut(\mathtt{MAX}(I, O))$ to may not coincide with I^*.

D. FSM Equations

Solving the problem of designing an agent of the multiagent system, the problem of solving the system of parallel FSM equations arises. Let $\mathtt{C} = (C, I \cup V, O \cup U, T_{\mathtt{C}}, c_0)$ and $\mathtt{S} = (S, I, O, T_{\mathtt{S}}, s_0)$ be two complete FSMs. An expression "$\mathtt{C} \lozenge \mathtt{X} \cong \mathtt{S}$" is called an *FSM equation* w.r.t. the unknown FSM $\mathtt{X}$ over the input alphabet U, and the output alphabet V. The FSM $\mathtt{C}$ is called the *context*, and the FSM $\mathtt{S}$ is called the *specification*. An FSM equation can have no solution. For a solvable equation, there exists a largest solution that is the FSM with the language $\overline{Aut(\mathtt{C}) \lozenge \overline{Aut(\mathtt{S})}} = \overline{(Aut(\mathtt{C}) \cap \overline{Aut(\mathtt{S})}_{\uparrow U \cup V})_{\downarrow U \cup V}}$ [2]. Undefined input sequences of the largest solution correspond to input sequences which violate the specification and for this reason, when designing an agent we are interested in complete solutions to the FSM equation. If a solvable FSM equation has a complete solution then it is known to have the largest complete solution. Each complete solution of an equation is a reduction of the largest complete solution [2].

E. System of FSM Equations

If an agent has to work in different contexts and it is necessary to provide a certain level of service in each context then when designing an agent, the problem arises of solving systems of equations.

Let $\mathtt{C}_i$ and $\mathtt{S}_i$ be collections of complete FSMs where all $\mathtt{C}_i$ (and all $\mathtt{S}_i$) are defined over the same input and output alphabets. A collection of equations $\mathtt{C}_i \lozenge \mathtt{X} \cong \mathtt{S}_i$, $i = 1, 2, \ldots, k$, is called a *system of FSM equations*. An FSM $\mathtt{A}$ defined over the alphabets of the unknown component $\mathtt{X}$ is a solution to the system if $\mathtt{A}$ is a solution to each equation of the system. The detailed method for solving FSM equations is described in [4] where it is also shown that a solvable FSM equation has the largest solution. A solvable system of parallel FSM equations also has the largest solution that is the intersection over largest solutions to all equations.

III. PROBLEM STATEMENT AND A SOLUTION PROPOSED

When solving an FSM equation $\mathtt{C} \lozenge \mathtt{X} \cong \mathtt{S}$, a corresponding FSM inequality $\mathtt{C} \lozenge \mathtt{X} \le \mathtt{S}$ is solved first. Given a system of FSM inequalities $\mathtt{C}_i \lozenge X \le \mathtt{S}_i$, $i = 1, 2$, the question is whether we could reduce the system of FSM inequalities to a single FSM inequality. In other words, whether there exists a single equation such that the sets of solutions of the system and this equation coincide.

In this paper, we consider two special cases.

A. Case 1

Let $\mathtt{S}_1 = \mathtt{S}_2 = \mathtt{S}$, be the specification, however, this service has to be provided into two different contexts $\mathtt{C}_1$ and $\mathtt{C}_2$.

Theorem 1. Given a system of FSM inequalities $\mathtt{C}_i \lozenge \mathtt{X} \le \mathtt{S}$, $i = 1, 2$, where $\mathtt{C}_1$ and $\mathtt{C}_2$ are specified over the same alphabets, the largest solution to the system and the largest solution to the inequality $(\mathtt{C}_1 \cup \mathtt{C}_2) \lozenge \mathtt{X} \le \mathtt{S}$ coincide.

The sketch of the proof. For each sequence β of the context $\mathtt{C}_1$ (and also $\mathtt{C}_2$) and each sequence α of the largest solution $\mathtt{M}$ of the system $\mathtt{C}_i \lozenge \mathtt{X} \le \mathtt{S}$, $(i = 1, 2)$ the composition $\beta \lozenge \alpha$ is in the specification $\mathtt{S}$. Since the union of the contexts $\mathtt{C}_1$ and $\mathtt{C}_2$ has only sequences of $\mathtt{C}_1$ and $\mathtt{C}_2$, for each sequence β of the context $\mathtt{C}_1 \cup \mathtt{C}_2$ and each sequence α of the largest solution $\mathtt{M}$, the composition $\beta \lozenge \alpha$ is in the specification $\mathtt{S}$.

On the other hand, for each sequence α of the largest solution $\mathtt{M}_i$ to a single equation $\mathtt{C}_i \lozenge \mathtt{X} \le \mathtt{S}$ it holds that $\beta \lozenge \alpha$ is in $\mathtt{S}$.

The following statement can be proven by induction.

Corollary 1. Given a system of FSM inequalities $\mathtt{C}_i \lozenge \mathtt{X} \le \mathtt{S}$, $i = 1, 2, \ldots, k$, where $\mathtt{C}_1, \ldots, \mathtt{C}_k$ are specified over the same alphabets, the largest solution to the system and the largest solution to the inequality $(\mathtt{C}_1 \cup \ldots \cup \mathtt{C}_k) \lozenge \mathtt{X} \le \mathtt{S}$ coincide.

Corollary 2. Given a system of FSM equations $\mathtt{C}_i \lozenge \mathtt{X} \cong \mathtt{S}$, $i = 1, 2, \ldots, k$, where $\mathtt{C}_1, \ldots, \mathtt{C}_k$ are specified over the same alphabets and FSM $\mathtt{S}$ is deterministic, the largest solutions to the system and to the equation $(\mathtt{C}_1 \cup \ldots \cup \mathtt{C}_k) \lozenge \mathtt{X} \cong \mathtt{S}$ coincide.

We now illustrate Theorem 1 by the following examples.

Example 1. Consider parallel composition in Fig. 1, contexts $\mathtt{C}_1$ and $\mathtt{C}_2$ in Figs. 2 and 4 and the specification $\mathtt{S}$ with the set of transitions $(1, i_1, o_1, 2)$, $(2, i_1, o_2, 2)$, $(2, i_2, o_1, 2)$.

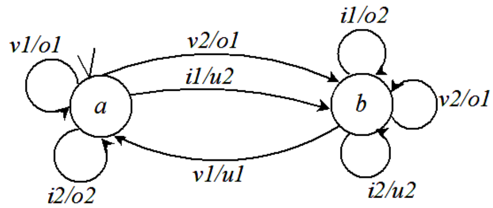

Fig. 2. The context $\mathtt{C}_1$

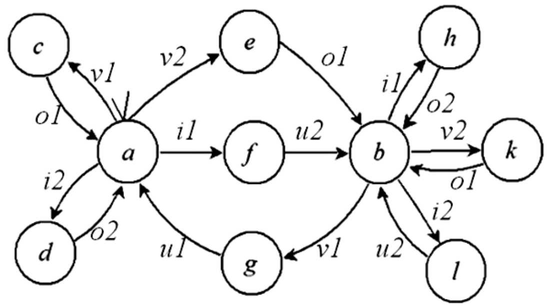

Fig. 3. An automaton $Aut(\mathtt{C}_1)$

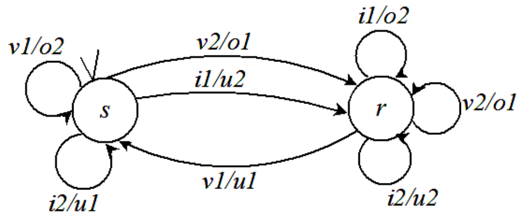

Fig. 4. The context C_2

The specification S is defined over the set of external inputs $I = \{i_1, i_2\}$ and the set of external outputs $O = \{o_1, o_2\}$. Contexts C_1 and C_2 are defined over the set of external inputs $I = \{i_1, i_2\}$, the set of external outputs $O = \{o_1, o_2\}$, the set of internal inputs $V = \{v_1, v_2\}$ and the set of internal outputs $U = \{u_1, u_2\}$ and are shown in Figs. 2 and 4. For the context C_2, there is no solution to the inequality $C_2 \Diamond X \le S$ and thus, there is no solution for the system of inequalities. By direct inspection, a reader can assure that for the union of C_1 and C_2, there also is no solution of the inequality $(C_1 \cup C_2) \Diamond X \le S$.

Example 2. Consider the specification FSM S with the set of transitions $(1, i_1, o_1, 2)$ $(1, i_2, o_2, 1)$, $(2, i_1, o_2, 2)$, $(2, i_2, o_1, 2)$. The largest solution L to the system over the set of inputs $U = \{u_1, u_2\}$ and the set of outputs $V = \{v_1, v_2\}$ derived as the intersection of largest solutions to the inequalities $C_1 \Diamond X \le S$ and $C_2 \Diamond X \le S$ is shown in Fig. 5. Consider now the inequality $(C_1 \cup C_2) \Diamond X \le S$. By direct inspection one can assure that the largest complete solution L' to the inequality $(C_1 \cup C_2) \Diamond X \le S$ is equivalent to the FSM in Fig. 5.

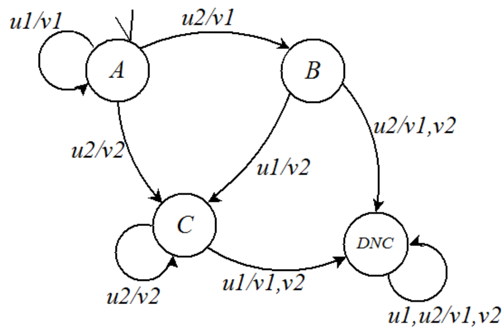

Fig. 5. The largest complete solution L to the system

B. Case 2

Let $C_1 = C_2 = C$ be the context; however, different services have to be provided in this context.

Theorem 2. Given a system of FSM inequalities $C \Diamond X \le S_i$, $i = 1, 2$, where S_1 and S_2 are specified over the same alphabets, the largest solution to the system and the largest solution to the inequality $C \Diamond X \le (S_1 \cap S_2)$ coincide.

The sketch of the proof. For each sequence β of the context C and each sequence α of the largest solution M of the system $C \Diamond X \le S_i$, $(i = 1, 2)$, the composition $\beta \Diamond \alpha$ is in the specification S_1 (and also is in S_2). Since the intersection of specifications S_1 and S_2 has only sequences which are in both S_1 and S_2, for each sequence β of the context C and each sequence α of the largest solution M, the composition $\beta \Diamond \alpha$ is in $S_1 \cap S_2$.

On the other hand, for each sequence α of the largest solution M_i to a single equation it holds that $\beta \Diamond \alpha$ is in $S_1 \cap S_2$.

The following statement can be proven by induction.

Corollary 1. Given a system of FSM inequalities $C \Diamond X \le S_i$, $i = 1, 2, \ldots, k$, where $S_1, \ldots, S_k$ are specified over the same alphabets, the largest solution to the system and the largest solution to the inequality $C \Diamond X \le (S_1 \cap \ldots \cap S_k)$ coincide.

Corollary 2. Given a system of FSM equations $C \Diamond X \cong S_i$, $i = 1, 2, \ldots, k$, where $S_1, \ldots, S_k$ are deterministic, the largest solution to the system and the largest solution to the equation $C \Diamond X \cong (S_1 \cap \ldots \cap S_k)$ coincide.

IV. Conclusion

In this paper, we have studied the problem of reducing the system of parallel FSM equations to a single equation that arises in many applications, such as the optimization of digital circuits, cryptography, logic synthesis, logic games, stabilizers for the synthesis of asynchronous systems, etc. In particular, we have considered two special cases when such reduction can be performed. For each case, it is shown how to derive a corresponding single equation. A general case, when contexts and specifications of different equations do not coincide, needs more research and this is our future work. The same approach can be applied for the general case of parallel composition when both, the context and the unknown have external input and outputs. We also mention that the system of equations over the synchronous composition can be treated in the same way; the synchronous composition operator corresponds to the case when all components are active at each time instance and is used for describing the behavior of hardware modular systems.

Acknowledgment

This work is partly supported by Project 739 (goszadanie minobrnayki RF).

References

[1] K. El-Fakih and N. Yevtushenko, “Progressive solutions to FSM equations,” CIAA, pp. 274-282, 2008.

[2] N.V. Yevtushenko, M.V. Rekun and S.V. Tihomirova, “Nondeterministic Finite State Machines: analysis and synthesis. Part 2. Solving FSM Equations,” Tomsk, Tomsk State University, p. 111, 2009. (In Russian).

[3] G. v. Bochmann, “Using logic to solve the submodule construction problem,” Discrete Event Dynamic Systems, pp. 27-59, 2013.

[4] T. Villa, N. Yevtushenko, R.K. Brayton, A. Mishchenko, A. Petrenko and A. Sangiovanni-Vincentelli, “The Unknown Component Problem,” Theory and Applications, p. 312, 2012.

[5] J.E. Hopcroft, R. Motwani, and J.D. Ullman, “Introduction to Automata Theory, Languages, and Computation,” Addison-Wesley Publishing Company, 2001.