Equisolvability of Series vs. Controller's Topology in Synchronous Language Equations

Nina Yevtushenko[¶] Tiziano Villa[§] Robert K. Brayton[†] Alex Petrenko[‡]

Alberto L. Sangiovanni-Vincentelli[†]

[¶]Dept. of EECS, Tomsk State University, Tomsk, 634050, Russia

[§]PARADES, Via di S.Pantaleo, 66, 00186 Roma, Italy

[†]Dept. of EECS, Univ. of California, Berkeley, CA 94720

[‡]CRIM, 550 Sherbrooke West, Montreal, H3A 1B9, Can

May 15, 2002

Abstract

Given a plant M_A and a specification M_C, the largest solution of the FSM equation $M_X \bullet M_A \preceq M_C$ contains all possible discrete controllers M_X. Often we are interested in computing the complete solutions whose composition with the plant is exactly equivalent to the specification. Not every solution contained in the largest one satisfies such property, that holds instead for the complete solutions of the series topology. We study the relation between the solvability of an equation for the series topology and of the corresponding equation for the controller's topology. We establish that, if M_A is a deterministic FSM, then the FSM equation $M_X \bullet M_A \preceq M_C$ is solvable for the series topology with an unknown head component iff it is solvable for the controller's topology. Our proof is constructive, i.e., for a given solution M_B of the series topology it shows how to build a solution M_D of the controller's topology and viceversa.

1 Introduction

An important step in the design of complex systems is the decomposition of the system into a number of separate components which interact in some well-defined way. In this context, a typical question is how to design a component that combined with a known part of the system, called the context, conforms or satisfies or matches a given overall specification. This question arises in several applications ranging from logic synthesis to the design of discrete controllers.

In [4] we proposed a general frame based on defining equations over languages associated to the components of a given system. We introduced two composition operators for abstract languages: synchronous composition, $\bullet$, and parallel composition, $\diamond$, and we studied the most general solutions of the language equations $A \bullet X \subseteq C$ and $A \diamond X \subseteq C$ ($\subseteq$ denotes language containment), defining the language operators needed to express them. In particular we studied the solutions of the equations defined over finite state machines (FSMs) of the type $M_A \bullet M_X \subseteq M_C$ and $M_A \diamond M_X \subseteq M_C$, where M_A models the context, M_C models the specification and M_X is unknown. Basic definitions about equations over languages are provided in Sec. 2. We refer to [5] for a report on FSM equations of the type $M_A \diamond M_X \subseteq M_C$ and to [3], Chap. 6, for a survey of previous work.

There are various FSM composition topologies of interest. For instance, the problem of designing a discrete controller that controls a given discrete plant in order to match a specification (represented by an

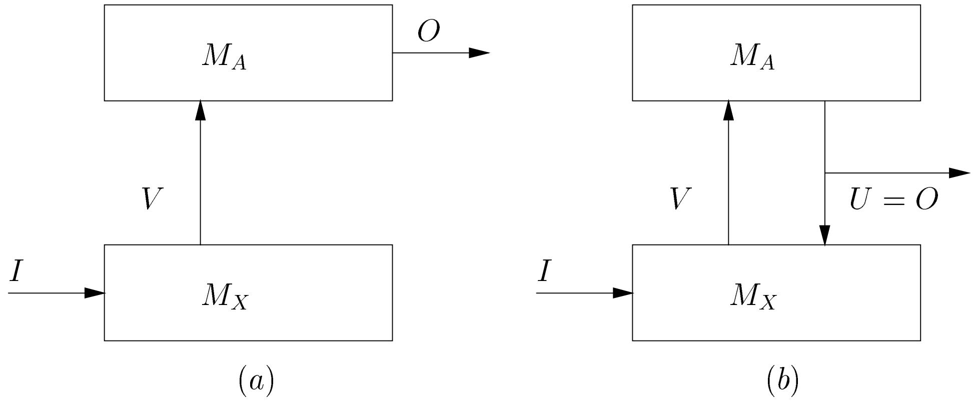

Figure 1: (a) 1-way cascade (or 1-way series) topology; (b) controller's (or supervisory control) topology.

FSM too) yields the so-called controller's (or supervisory control) topology (see [1] for an introduction to the discrete model matching problem). A more straightforward interconnection is the series topology where signals flow uni-directionally from a head FSM to a tail FSM. Figure 1 shows a series topology and a controller's topology. In the series topology M_X is the unknown head component, and M_A is the given tail component; in the controller's topology M_X is the unknown controller component, and M_A is the given plant component.

Given a plant M_A and a specification M_C, the largest solution of the FSM equation $M_X \bullet M_A \preceq M_C$ contains all possible discrete controllers M_X. Consider the constrained problem of practical interest $M_X \bullet M_A \cong M_C$, where both M_A and M_C are complete deterministic FSMs (DFSMs), and the objective is to find all complete DFSMs whose composition with M_A is equivalent to M_C ($\cong$ stands for equivalent). If M_A is not a Moore DFSM, the composition of a complete solution (when it is not a Moore DFSM either) with M_A may fail to produce a complete DFSM, i.e., the largest solution may contain complete DFSMs whose composition with M_A is not a complete DFSM (and so not be equivalent to M_C). Therefore if the goal is to select an 'optimal' controller, e.g., one with a minimum number of states, it will not be sufficient to find an 'optimal' reduction (contained FSM) of the largest solution, because this reduction may not be a solution of the original equation.

Instead the equation for the series topology has the property that every complete reduction of the largest solution is a DFSM whose composition with M_A yields a complete DFSM. Can this fact be of help when solving (or modelling) the controller's topology ? To answer the question we should study the relation between the solvability of an equation for the series topology and of the corresponding equation for the controller's topology.

In this note we establish that, if M_A is a deterministic FSM, then the FSM equation $M_X \bullet M_A \preceq M_C$ is solvable for the series topology with an unknown head component iff it is solvable for the controller's topology. The non-trivial direction of the proof is going from the controller's topology to the series topology. The proof is based on the following fact: let M_D be a solution for the controller's topology (M_D has inputs I and O and output V), then M_C composed with M_D is a solution for the series topology ($M_C \bullet M_D$ has input I and output V). Indeed, when an input i is applied to such composition $M_C \bullet M_D$ M_C produces the reference output o; when this pair (i, o) is applied to M_D, by construction M_D produces the internal signal v under which M_A produces the reference output o generated by M_C under input i. The other direction relies on the fact that a solution for the series topology can be augmented with an inessential input o such that its next state and output functions do not depend on o.

Notice that the theorem of equisolvability can be proved for the general equation $M_X \bullet M_A \preceq M_C$, with no restriction on M_C (M_C is a NDFSM), whereas M_A should be a DFSM (or an appropriate restriction).

2 Equations over Languages

2.1 Languages

We remind the notions of substitution and homomorphism of languages [2]. A **substitution** f is a mapping of an alphabet Σ onto subsets of $\Delta^\star$ for some alphabet Δ. The substitution f is extended to strings by setting $f(\epsilon) = \epsilon$ and $f(xa) = f(x)f(a)$. An **homomorphism** h is a substitution such that $h(a)$ is a single string for each symbol a in the alphabet Σ.

1. Given a language L over alphabet $X \times V$, consider the homomorphism h defined as $h((x,v)) = x,\ \forall x \in X, \forall v \in V$, then the language $L_{\downarrow X} = \{h(\alpha) \mid \alpha \in L\}$ over alphabet X is the **projection** of language L to alphabet X, or X-projection of L. By definition of substitution $h(\epsilon) = \epsilon$.

2. Given a language L over alphabet X and an alphabet V, consider the substitution f defined as $f(x) = (x,v),\ \forall x \in X, \forall v \in V$, then the language $L_{\uparrow V} = \{f(\alpha) \mid \alpha \in L\}$ over alphabet $X \times V$ is the **lifting** of language L over alphabet V, or V-lifting of L. By definition of substitution $f(\epsilon) = \epsilon$.

2.2 Synchronous Composition of Languages

Consider two systems A and B with associated languages $L(A)$ and $L(B)$. The systems communicate with each other by a channel U and with the environment by channels I and O. We introduce a composition operator that describes the external behaviour of the composition of $L(A)$ and $L(B)$.

Definition 2.1 *Given alphabets I, U, O, language L_1 over $I \times U$ and language L_2 over $U \times O$, the* **synchronous composition** *of languages L_1 and L_2 is the language* [1] $[(L_1)_{\uparrow O} \cap (L_2)_{\uparrow I}]_{\downarrow I \times O}$*, denoted by $L_1 \bullet L_2$, defined over $I \times O$.*

2.3 Solution of Language Equations under Synchronous Composition

Given alphabets I, U, O, a language A over alphabet $I \times U$ and a language C over alphabet $I \times O$, let us consider the language equation

$$X \bullet A \subseteq C. \tag{1}$$

Definition 2.2 *Given alphabets I, U, O, a language A over alphabet $I \times U$ and a language C over alphabet $I \times O$, language B over alphabet $U \times O$ is called a* **solution** *of the equation $X \bullet A \subseteq C$ iff $B \neq \emptyset$ and $B \bullet A \subseteq C$. The* **largest solution** *is the solution that contains any other solution.*

Theorem 2.1 *The largest solution of the equation $A \bullet X \subseteq C$ is the language $S = \overline{A \bullet \overline{C}}$, if $S \neq \emptyset$. A language $B \neq \emptyset$ over alphabet $U \times O$ is a solution of $X \bullet A \subseteq C$ iff $B \subseteq \overline{A \bullet \overline{C}}$.*
If $\overline{A \bullet \overline{C}} = \emptyset$, then the language equation $X \bullet A \subseteq C$ has no solution.

3 Equisolvability of Series vs. Controller's Topology

Definition 3.1 *A* **non-deterministic FSM** *(NDFSM), or simply an* **FSM** *or a* **machine***, is defined as a 5-tuple $M = \langle S, I, O, T, r\rangle$ where S represents the finite state space, I represents the finite input space, O represents the finite output space and $T \subseteq I \times S \times S \times O$ is the transition relation. On input i, the NDFSM at present state p may transit to next state n and output o if and only if $(i, p, n, o) \in T$. State $r \in S$*

[1] Use the same order $I \times U \times O$ in the languages $(L_1)_{\uparrow O}$ and $(L_2)_{\uparrow I}$.

represents the initial or reset state. We denote the restriction of relation T to $I \times S \times S$ (next state relation) by $T_n \subseteq I \times S \times S$, i.e., $(i, s, s') \in T_n \Leftrightarrow \exists o\ (i, s, s', o) \in T$; similarly, we denote the restriction of relation T to $I \times S \times O$ (output relation) by $T_o \subseteq I \times S \times O$, i.e., $(i, s, o) \in T_o \Leftrightarrow \exists s'\ (i, s, s', o) \in T$. We may use δ instead of T_n and λ instead of T_o. If at least a transition is specified for each present state and input pair, the FSM is said to be **complete**. *If no transition is specified for at least a present state and input pair, the FSM is said to be* **partial**. *An FSM is said to be* **trivial** *when $T = \emptyset$.*

The following theorem argues that a synchronous FSM equation is solvable for the controller's topology iff it is solvable for the series topology. The proof shows how the two sets of solutions relate.

Theorem 3.1 *Given a plant $M_A = (S_A, V, O, T_A, r_A)$ and a reference model $M_C = (S_C, I, O, T_C, r_C)$, if M_A is a DFSM then the FSM equation $M_X \bullet M_A \preceq M_C$ is solvable for the series topology with an unknown head component iff it is solvable for the controller's topology.*

Proof.
Only if Let $M_B = (S, I, V, T_B, r)$ be an FSM solution of the equation for the series topology. Construct the FSM $M_D = (S, I \times O, V, T_D, r)$, where T_D is defined by

$$(s, io, v, s') \in T_D \Leftrightarrow (s, i, v, s') \in T_B.$$

We show that M_D is a solution of the equation for the controller's topology.
Given states $(s, s_A), (s', s'_A) \in S \times S_A$, $i \in I$ and $o \in O$, by definition of $T_{B \bullet A}$,

$$((s, s_A), i, o, (s', s'_A)) \in T_{B \bullet A} \Leftrightarrow \exists v \in V [(s, i, v, s') \in T_B \wedge (s_A, v, o, s'_A) \in T_A].$$

Furthermore, by definition of T_D,

$$\exists v \in V [(s, i, v, s') \in T_B \wedge (s_A, v, o, s'_A) \in T_A] \Leftrightarrow \exists v \in V [(s, io, v, s') \in T_D \wedge (s_A, v, o, s'_A) \in T_A],$$

and, by definition of $T_{D \bullet A}$,

$$\exists v \in V [(s, io, v, s') \in T_D \wedge (s_A, v, o, s'_A) \in T_A] \Leftrightarrow ((s, s_A), i, o, (s', s'_A)) \in T_{D \bullet A}.$$

Summing it up

$$((s, s_A), i, o, (s', s'_A)) \in T_{B \bullet A} \Leftrightarrow ((s, s_A), i, o, (s', s'_A)) \in T_{D \bullet A}.$$

Therefore $M_B \bullet M_A$ and $M_D \bullet M_A$ are equivalent and M_D is a solution of the equation for the controller's topology.
If For notational convenience in this proof we represent the transition relation T of a DFSM by the next state function δ and by the output function λ.

Let $M_D = (S, I \times O, V, \delta_D, \lambda_D, r)$ be a DFSM that is a solution for the controller's topology of the FSM equation $M_X \bullet M_A \preceq M_C$ (if the equation is solvable there always exists a deterministic solution); moreover the composition $M_D \bullet M_A$ is a DFSM, since both M_D and M_A are DFSMs.

Construct an FSM $M_B = (S \times S \times S_A, I, V, \delta_B, \lambda_B, (r, r, r_A))$ such that for all $(s, s, s_A) \in S \times S \times S_A$ and $i \in I$ it holds that

$$\begin{aligned}
\delta_B((s, s, s_A), i) &= (\delta_D(s, (i, \lambda_{D \bullet A}((s, s_A), i))), \delta_{D \bullet A}((s, s_A), i)), \\
\lambda_B((s, s, s_A), i) &= \lambda_D(s, (i, \lambda_{D \bullet A}((s, s_A), i))).
\end{aligned}$$

Notice that states of the form (s', s'', s_A), with $s', s'' \in S$ and $s' \neq s''$, need not be considered because they are unreachable. We show that $M_B \bullet M_A = M_D \bullet M_A$, implying that M_B is a solution for the series topology of the equation $M_X \bullet M_A \preceq M_C$. To argue that $M_B \bullet M_A = M_D \bullet M_A$, we need to prove that their respective initial states $((r, r, r_A), r_A) \in ((S, S, S_A), S_A)$ and $(r, r_A) \in (S, S_A)$ are equivalent, meaning that they agree on the outputs and that their respective next states are equivalent.

Define the matching states $((s, s, s_A), s_A) \in ((S, S, S_A), S_A)$ and $(s, s_A) \in (S, S_A)$ as *similar states*. We show that the next states of similar states are similar states. Precisely, for each pair of similar states $((s, s, s_A), s_A) \in ((S, S, S_A), S_A)$ of the series composition $M_B \bullet M_A$ and $(s, s_A) \in (S, S_A)$ of the controller's composition $M_D \bullet M_A$, and for each input $i \in I$

$$\delta_{B\bullet A}(((s, s, s_A), s_A), i) = ((s', s', s'_A), s'_A)$$

where

$$(s', s'_A) = \delta_{D\bullet A}((s, s_A), i).$$

Indeed, if $\delta_{D\bullet A}((s, s_A), i) = (s', s'_A)$, then $\delta_D(s, (i, \lambda_{D\bullet A}((s, s_A), i))) = s'$ and, by definition of δ_B, it is $\delta_B((s, s, s_A), i) = (s', s', s'_A)$. Finally, from $\delta_A(s_A, \lambda_D(s, (i, \lambda_{D\bullet A}((s, s_A), i)))) = s'_A$, it follows $\delta_{B\bullet A}(((s, s, s_A), s_A), i) = ((s', s', s'_A), s'_A)$.

This fact allows to prove by induction that states reached at the any finite step from the initial state are similar. The basis of the induction holds because the initial states are similar by construction. If by induction hypothesis the states reached at the $(i-1)$-th state are similar, then by the previous result the states reached at the i-th step are similar too.

Next we show that for each pair of similar states $((s, s, s_A), s_A) \in ((S, S, S_A), S_A)$ of the series composition $M_B \bullet M_A$ and $(s, s_A) \in (S, S_A)$ of the controller's composition $M_D \bullet M_A$, and for each input $i \in I$

$$\lambda_{B\bullet A}(((s, s, s_A), s_A), i) = \lambda_{D\bullet A}((s, s_A), i),$$

meaning that similar states agree on the outputs. For each state $((s, s, s_A), s_A) \in ((S, S, S_A), S_A)$ of the series composition $M_B \bullet M_A$ and for each input $i \in I$, by definition of series composition, it holds

$$\lambda_{B\bullet A}(((s, s, s_A), s_A), i) = \lambda_A(s_A, \lambda_B((s, s, s_A), i))$$

and, by definition of λ_B,

$$\lambda_A(s_A, \lambda_B((s, s, s_A), i)) = \lambda_A(s_A, \lambda_D(s, (i, \lambda_{D\bullet A}((s, s_A), i)))).$$

Furthermore, by definition of controller's composition,

$$\lambda_{D\bullet A}((s, s_A), i) \subseteq \lambda_A(s_A, \lambda_D(s, (i, \lambda_{D\bullet A}((s, s_A), i)))),$$

where the containment relation can be replaced by strict equality under the hypothesis that A is a DFSM[2], yielding

$$\lambda_{D\bullet A}((s, s_A), i) = \lambda_A(s_A, \lambda_D(s, (i, \lambda_{D\bullet A}((s, s_A), i)))).$$

Putting it all together

$$\lambda_{B\bullet A}(((s, s, s_A), s_A), i) = \lambda_A(s_A, \lambda_D(s, (i, \lambda_{D\bullet A}((s, s_A), i)))) = \lambda_{D\bullet A}((s, s_A), i).$$

Summing up, we showed first that states reached at any finite step are similar and then that similar states agree on the outputs, implying that states reached from the initial states at any finite step agree on the outputs, and so that the initial states are equivalent. Therefore $M_B \bullet M_A$ and $M_D \bullet M_A$ are equivalent. □

[2]By definition of series composition it is always $\lambda_{B\bullet A}(((s, s, s_A), s_A), i) = \lambda_A(s_A, \lambda_B((s, s, s_A), i))$, whereas for controller's composition $\lambda_{D\bullet A}((s, s_A), i) \subseteq \lambda_A(s_A, \lambda_D(s, (i, \lambda_{D\bullet A}((s, s_A), i))))$, because an output $\tilde{o}$, issued at state s_A of M_A under input v, may be "blocked", meaning that from state s of M_D there is no transition with label $i\tilde{o}/v$. The hypothesis that M_A is a DFSM is sufficient (but not necessary) to guarantee that the previous situation cannot happen and so the containment relation becomes a strict equality.

Corollary 3.1 *Given a plant $M_A = (S_A, V, O, T_A, r_A)$ and a reference model $M_C = (S_C, I, O, T_C, r_C)$, if M_A is a DFSM then the FSM equation $M_X \bullet M_A \preceq M_C$ has a complete solution for the series topology with an unknown head component iff it has a complete solution for the controller's topology.*

Say that a solution M_D is compositionally complete if the composition $M_D \bullet M_A$ is a complete FSM. Under the same hypothesis, the FSM equation $M_X \bullet M_A \preceq M_C$ has a compositionally complete solution for the series topology with an unknown head component iff it has a compositionally complete solution for the controller's topology.

The construction in the proof of Th. 3.1 establishes immediately both results.

4 Conclusions

In this note we established that, if M_A is a deterministic FSM, then the FSM equation $M_X \bullet M_A \preceq M_C$ is solvable for the series topology with an unknown head component iff it is solvable for the controller's topology. Our proof is constructive, i.e., for a given solution M_B of the series topology it shows how to build a solution M_D of the controller's topology and viceversa.

A practical implication might be a procedure to compute discrete controllers by solving first a companion series topology equation and then transforming its solutions to solutions for the controller's topology. Notice that the largest solution for the series topology may have more states than the one for the controller's topology, however the latter has more inputs.

5 Acknowledgments

The first author was partly supported by the Russian Fund of Basic Research. Both the first and the third author gratefully acknowledge the support of a NATO travel grant (NATO Linkage Grant No. 971217). The second author gratefully acknowledges the support of the MADESSII Project (Italian National Research Council). The fourth author gratefully acknowledges the support of NSERC (Grant OGP0194381).

References

[1] M. Di Benedetto, A. Sangiovanni-Vincentelli, and T. Villa. Model Matching for Finite State Machines. *IEEE Transactions on Automatic Control*, 46(11):1726–1743, December 2001.

[2] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company, 1979.

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

[4] N. Yevtushenko, T. Villa, R. Brayton, A. Petrenko, and A. Sangiovanni-Vincentelli. Sequential synthesis by language equation solving. *The Proceedings of the International Workshop on Logic Synthesis*, June 2000.

[5] N. Yevtushenko, T. Villa, R. Brayton, A. Petrenko, and A. Sangiovanni-Vincentelli. Solution of parallel language equations for logic synthesis. In *The Proceedings of the International Conference on Computer-Aided Design*, pages 103–110, November 2001.