Automatica 87 (2018) 290–300

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Symbolic reachability analysis and maximally permissive entrance control for globally synchronized templates ☆

Liyong Lin [a], Alin Stefanescu [b], Weilin Wang [c], Rong Su [d], W. Murray Wonham [a]

[a] Department of Electrical and Computer Engineering, University of Toronto, 27 King's College Circle, Toronto, ON M5S 1A1, Canada
[b] Department of Computer Science, University of Bucharest, Str. Academiei nr. 14, 010014 Bucharest, Romania
[c] Institute of Transport Studies, Department of Civil Engineering, Monash University, Victoria 3168, Australia
[d] Center for System Intelligence and Efficiency, School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 19 November 2016
Received in revised form 23 August 2017
Accepted 30 August 2017

Keywords:
Symbolic reachability analysis
Parameterized discrete-event systems
Supervisor synthesis
Deadlock
Blocking

ABSTRACT

This paper studies the symbolic reachability relations of a class of parameterized systems in the framework of regular model checking. The modules of each system are instantiated from a globally synchronized template, and each globally synchronized template is represented by a finite state automaton whose event set consists of global events and local events. It is shown that the symbolic reachability relations of these systems are effectively iteration-closed star languages. And for any iteration-closed star language, there exists a template with only global events that realizes it. Application of the symbolic reachability analysis to computing the entrance control functions that enforce deadlock freeness and blocking freeness is then illustrated for systems with idle modules. In particular, we show that the maximally permissive entrance control functions can be encoded using finite state automata.

1. Introduction

Parameterized systems, where an indefinite number of instantiations of the same template usefully interact, are pervasive and critical in our modern society, e.g., telecommunication protocols, sensor networks, etc. However, in general, the correctness of these systems is difficult to analyze, due to the algorithmic undecidability and high computational complexity (Aminof, Kotek, Rubin, Spegni, & Veith, 2014; Apt & Kozen, 1986; Esparza, 2014). Regular model checking has been proposed as a generic symbolic framework for the algorithmic verification of infinite state systems based on automata theory, where sets of states are represented by regular languages and transition relations by regular relations (Abdulla, Jonsson, Nilsson, & Saksena, 2004; Bouajjani, Legay, & Wolper, 2005). The most fundamental problem considered in this framework is symbolic state space exploration, which relies on computing the reflexive and transitive closure of the transition relation, i.e., computing the symbolic reachability relation. However, the (symbolic) reachability relations of most parameterized systems are not effectively computable. Partial algorithms for computing (regular approximations of) reachability relations have been well investigated, using the techniques of widening (Bouajjani, Jonsson, Nilsson, & Touili, 2000; Touili, 2001), inference of regular languages (Habermehl & Vojnar, 2005), abstraction-refinement of automata (Bouajjani, Habermehl, & Vojnara, 2004), and others (Boigelot, Legay, & Wolper, 2003; Dams, Lakhnech, & Steffen, 2001; Jonsson & Nilsson, 2000).

In this work, we identify a class of parameterized systems whose symbolic reachability relations are regular and effectively computable. The modules of each system are instantiated from a globally synchronized template, and each globally synchronized template is a finite state automaton whose event set consists of global events and local events. All modules in the system must synchronize on common global events while each local event is allowed to be executed independently. Motivated by their potential applications to modeling aspects of the behaviors of communication systems and sensor networks, Rohloff & Lafortune (2006) studies several verification and control problems for these systems. It is shown in Rohloff and Lafortune (2006) that state reachability test, blocking freeness verification and deadlock freeness verification problems are all decidable. Later in Wang, Su, and Lin (2013), algorithms that avoid explicit synchronization of modules are developed to verify the deadlock freeness and blocking freeness properties, in an attempt to reduce the verification complexity. Given any n fixed modules, a necessary and sufficient condition that can be used to verify, for an arbitrary global specification over

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Joerg Raisch under the direction of Editor Christos G. Cassandras.

E-mail addresses: liyong.lin@utoronto.ca (L. Lin), alin@stefanescu.eu (A. Stefanescu), wlwang@usst.edu.cn (W. Wang), rsu@ntu.edu.sg (R. Su), wonham@control.toronto.edu (W. Murray Wonham).

https://doi.org/10.1016/j.automatica.2017.10.015
0005-1098/

the n modules, the existence of n isomorphic supervisors is also provided in Rohloff and Lafortune (2006). If the language of the closed-loop system is required to be a non-empty subset of the global specification, then the decision problem is undecidable for any fixed $n \geq 2$ (Lin, Stefanescu, & Su, 2016; Lin, Stefanescu, Su, Wang, & Shehabinia, 2014). A more scalable alternative is to synthesize a parameterized supervisor that solves the control problem in a uniform manner, regardless of the number of modules in the system (Bherer, Desharnais, & St-Denis, 2009). In general, necessary restrictions must be imposed to ensure decidability, since parameterized systems are essentially infinite state systems. Symbolic frameworks based on predicates and predicate transformers have been proposed and studied for several different models of infinite state systems (see, for example, Refs. Bherer et al. 2009; Gall, Jeannet, & Marchand, 2005; Kumar & Garg, 1993, 2005). Although the deadlock avoidance property has been considered, the problem of ensuring blocking freeness has not been addressed.

The contributions of this work are the following: we obtain a characterization of the expressive power[1] of these parameterized systems in terms of their reachability relations, and show that they correspond exactly to the sets of iteration-closed star languages over $S \times S$, where S is the finite state set of each template. Indeed, the reachability relations of these systems are effectively iteration-closed star languages over $S\times S$, i.e., languages of the form $\bigcup_{i=1}^{k} D_i^*$, where $\varnothing \neq D_i \subseteq S \times S$, that are closed under reflexive and transitive closure; and for any iteration-closed star language over $S \times S$, there exists a template with only global events that realizes it, that is, the reachability relation of the system is equal to the given language over $S \times S$. To the best of our knowledge, this is the first known result in characterizing the expressive power of classes of parameterized systems using the reachability relations. The reachability relation based characterization is of interest since reachability relations are closely related to models of computation (Engelfriet, 2007; Terlutte & Simplot, 2000). For each template, regular languages representing the sets of deadlocked states and blocked states are shown. We then also explain how to use the symbolic reachability analysis to finitely compute and encode the maximally permissive entrance control functions enforcing deadlock freeness and blocking freeness.

The paper is organized as follows. In Section 2, we review the mathematical preliminaries. In Section 3, we formally define the globally synchronized templates, and present a characterization of their expressive power using the symbolic reachability relations. The applications of the symbolic reachability analysis to computing deadlocked states and blocked states are shown in Section 4. We consider the entrance control problem in Section 5. Finally, discussions and conclusions are given in the last section, Section 6.

2. Preliminaries

We assume the reader to be familiar with the basic theories of formal languages, finite automata (Hopcroft & Ullman, 1979) and supervisory control (Harrison, 1978). In this section, we shall fix the notations and terminologies used in this work, and review some additional mathematical preliminaries.

Let $\mathbb{N}$ denote the set of non-negative integers. The difference of two sets A and B is denoted by $A - B$. The Cartesian product of A and B is denoted by $A \times B$. Given any relation R on set U, its inverse is denoted by R^{-1}. Given any two relations R_1, R_2 on U, their relational composition is denoted by $R_1 \circ R_2$. The ith iteration power $R^{i(\circ)}$ of R is defined inductively by $R^{0(\circ)} := \{(u, u) \in U \times U \mid u \in U\}$, and $R^{(i+1)(\circ)} := R^{i(\circ)} \circ R$ for $i \geq 0$. $R^{0(\circ)}$ is the identity relation on U, denoted by id_U, and is the identity for relational composition. $R^{*(\circ)}$ stands for the reflexive and transitive closure, or the iteration closure, of R, that is, $R^{*(\circ)} := \bigcup_{i=0}^{\infty} R^{i(\circ)}$. R is said to be iteration-closed if $R^{*(\circ)} = R$. An automaton G over Σ is a 5-tuple $(S, \Sigma, \delta, S_i, S_m)$, where S is the set of states, Σ the finite event set or the alphabet, $\delta \subseteq S \times \Sigma \times S$ the transition relation, $S_i \subseteq S$ the set of initial states and $S_m \subseteq S$ the set of marked states. The components S_i, S_m are sometimes dropped in the discussion of an automaton if they are not important in the given context. If S is finite, then we say G is a finite state automaton. For each $\sigma \in \Sigma$, we use $E_\sigma := \{(s, s') \in S \times S \mid (s, \sigma, s') \in \delta\}$ to denote the transition relation (of G) associated with σ. We often use the 5-tuple $(S, \Sigma, (E_\sigma)_{\sigma\in\Sigma}, S_i, S_m)$ to denote G. We call $E_G := \bigcup_{\sigma\in\Sigma} E_\sigma$ the transition relation of G and $E_G^{*(\circ)}$ the reachability relation of G. Let $G^T := (S, \Sigma, (E_\sigma^{-1})_{\sigma\in\Sigma}, S_m, S_i)$ denote the transposed automaton of G, where S_m is the set of initial states and S_i the set of marked states of G^T. The transition relation of G^T is $E_{G^T} = \bigcup_{\sigma\in\Sigma} E_\sigma^{-1} = E_G^{-1}$. The reachability relation $(E_G^{-1})^{*(\circ)}$ of G^T is the co-reachability relation of G. Let $S(\sigma) := \{s \in S \mid \exists s' \in S, (s, s') \in E_\sigma\}$ denote the subset of states (of G) out of which there exists a transition labeled by event σ. The algebraic structure $\mathcal{S} = (2^{U\times U}, \cup, \circ, ^{*(\circ)}, \varnothing, id_U)$ is a Kleene algebra (Kozen, 1991). Thus we have the following result.

Lemma 1. *Let each R_i be a relation on U. It holds that* $(\bigcup_{i\in\{1,2,\ldots,k\}} R_i)^{*(\circ)} = (R_k^{*(\circ)} \circ R_{k-1}^{*(\circ)} \circ \ldots \circ R_1^{*(\circ)})^{*(\circ)}$.

The synchronous product of two finite automata $G_1 = (S_1, \Sigma_1, \delta_1, S_{1,i}, S_{1,m})$ and $G_2 = (S_2, \Sigma_2, \delta_2, S_{2,i}, S_{2,m})$ is the automaton $G_1 \parallel G_2 = (S_1\times S_2, \Sigma_1\cup\Sigma_2, \delta_1 \parallel \delta_2, S_{1,i}\times S_{2,i}, S_{1,m}\times S_{2,m})$, where $\delta_1 \parallel \delta_2 : S_1\times S_2\times(\Sigma_1\cup\Sigma_2) \mapsto S_1\times S_2$ is the relation defined such that

(1) if $\sigma \in \Sigma_1 \cap \Sigma_2$, then $((s_1, s_2), \sigma, (s_1', s_2')) \in \delta_1 \parallel \delta_2$ iff $(s_1, \sigma, s_1') \in \delta_1$ and $(s_2, \sigma, s_2') \in \delta_2$;
(2) if $\sigma \in \Sigma_1 - \Sigma_2$, then $((s_1, s_2), \sigma, (s_1', s_2')) \in \delta_1 \parallel \delta_2$ iff $(s_1, \sigma, s_1') \in \delta_1$ and $s_2 = s_2'$;
(3) if $\sigma \in \Sigma_2 - \Sigma_1$, then $((s_1, s_2), \sigma, (s_1', s_2')) \in \delta_1 \parallel \delta_2$ iff $s_1 = s_1'$ and $(s_2, \sigma, s_2') \in \delta_2$;

The graph disjoint union of G_1 and G_2 is the automaton $G_1\sqcup G_2 = (S_1\cup S_2, \Sigma_1\cup\Sigma_2, \delta_1\cup\delta_2, S_{1,i}\cup S_{2,i}, S_{1,m}\cup S_{2,m})$, where the sets S_1 and S_2 are assumed without loss of generality to be disjoint (renaming states when necessary).

A monoid is a triple $(M, \cdot_M, 1_M)$ formed from a set M, an associative binary operation $\cdot_M$ and the identity element 1_M. It is a usual practice to identify the monoid with the underlying set M. A homomorphism between two monoids M and N is a function $h : M \mapsto N$ such that $h(1_M) = 1_N$ and $\forall x, y \in M, h(x\cdot_M y) = h(x)\cdot_N h(y)$. We say a homomorphism $h : M \mapsto N$ is an isomorphism if h is bijective, and we say two monoids M and N are isomorphic if there exists an isomorphism between them. A submonoid N of a monoid M is a subset of M that is also a monoid under the same operation, containing the same identity element as that of M. The Cartesian product $M\times N$ of monoids M and N, where $(m_1, n_1)\cdot_{M\times N}(m_2, n_2) := (m_1\cdot_M m_2, n_1\cdot_N n_2)$ and $1_{M\times N} := (1_M, 1_N)$, is again a monoid.

Let Σ^* denote the set of finite strings over Σ. The length of string w (over Σ) is denoted by $|w|$. The concatenation $w_1 \cdot w_2$ of any two strings w_1, w_2 is often denoted by w_1w_2. Let $w_1 \sqcup\!\sqcup w_2 := \{v_1u_1v_2u_2\ldots v_ku_k \mid w_1 = v_1v_2\ldots v_k, w_2 = u_1u_2\ldots u_k, \text{where} v_i, u_i \in \Sigma^* \text{for each } i \in \{1, 2, \ldots, k\}\}$ denote the shuffle of any two strings w_1, w_2 over Σ. Let $L_1 \sqcup\!\sqcup L_2 := \bigcup_{s_1\in L_1, s_2\in L_2} s_1 \sqcup\!\sqcup s_2$ denote the shuffle of any two languages $L_1, L_2 \subseteq \Sigma^*$. For example, the shuffle of strings $w_1 = a_1b_1$ and $w_2 = a_1$ is $w_1 \sqcup\!\sqcup w_2 = \{a_1b_1a_1, a_1a_1b_1\}$. The commutative closure $[w]_{\sqcup\!\sqcup}$ of a string $w = \sigma_1\sigma_2\ldots\sigma_n$, where $\sigma_i \in \Sigma$, is defined to be the set $\{\sigma_1\} \sqcup\!\sqcup \{\sigma_2\} \ldots \sqcup\!\sqcup \{\sigma_n\}$. And the commutative closure of a language $L \subseteq \Sigma^*$ is defined to be $[L]_{\sqcup\!\sqcup} = \bigcup_{w\in L} [w]_{\sqcup\!\sqcup}$. L is said to be commutative if $[L]_{\sqcup\!\sqcup} = L$. The concatenation $L_1 \cdot L_2$ of $L_1, L_2 \subseteq \Sigma^*$

[1] The expressive power of a class of systems means the classes of languages that is represented by the class of systems.

is often denoted by L_1L_2. Given any finite number of non-empty subsets $D_1, D_2, \ldots, D_k \subseteq \Sigma$, where $k \geq 1$, $\bigcup_{i\in\{1,2,\ldots,k\}} D_i^*$ is said to be a star language over Σ. Let $Star(\Sigma)$ denote the (finite) collection of all the star languages over Σ. In the rest of this work, we will mainly deal with the cases when $\Sigma = S$ and when $\Sigma = S \times S$. We identify the monoid $(S \times S)^*$ with the submonoid $\{(w_1, w_2) \in S^* \times S^* \mid |w_1| = |w_2|\}$ of $S^* \times S^*$, since they are isomorphic. Indeed, the homomorphism $h : (S \times S)^* \mapsto \{(w_1, w_2) \in S^* \times S^* \mid |w_1| = |w_2|\}$, where $h((s_1, s_1')(s_2, s_2')\ldots(s_k, s_k')) := (s_1s_2\ldots s_k, s_1's_2'\ldots s_k')$ for each $k \geq 0$ and each $s_i, s_i' \in S$, is an isomorphism between the two monoids. For example, consider $S = \{s_1, s_2\}$. We identify string $(s_1, s_2)(s_2, s_2) \in (S \times S)^*$ with the tuple $(s_1s_2, s_2s_2) \in S^* \times S^*$ and identify language $\{(s_1, s_2)(s_2, s_1), (s_1, s_2)\} \subseteq (S \times S)^*$ with the relation $\{(s_1s_2, s_2s_1), (s_1, s_2)\} \subseteq S^* \times S^*$. For relations on S^*, the operations of relational composition and language concatenation can be performed. To emphasize the difference between the operations of relational composition and language concatenation for relations on S^*, we use the following illustrative example. Let $R_1 = \{(s_1, s_2), (s_1s_2, s_2s_1)\}$ and $R_2 = \{(s_2, s_1)\}$ be two relations on $\{s_1, s_2\}^*$. Their relational composition is $R_1 \circ R_2 := \{(s_1, s_1)\}$ and their concatenation is $R_1R_2 = \{(s_1s_2, s_2s_1), (s_1s_2s_2, s_2s_1s_1)\}$. A relation R on S^* is said to be length-preserving if whenever $(w_1, w_2) \in R \subseteq S^* \times S^*$, we have $|w_1| = |w_2|$, i.e., $R \subseteq (S \times S)^*$. In the rest of this work, we always mean length-preserving relations, whenever we talk about relations. A relation $R \subseteq (S \times S)^*$ is said to be regular iff R is a regular language over the alphabet $S \times S$. The class of regular (length-preserving) relations is closed under relational composition. Every regular relation $R \subseteq (S \times S)^*$ is recognized by a finite state automaton over $S \times S$, often called a finite state transducer. For $L \subseteq S^*$, the image of L under the transduction of $R \subseteq (S \times S)^*$ is $R[L] = \{w \in S^* \mid \exists w' \in L, (w', w) \in R\}$. A language (respectively, relation) is said to be effectively regular, whenever it is regular and a finite state automaton (respectively, transducer) that recognizes it can be computed. Given any regular relation $R \subseteq (S \times S)^*$ and any regular language L over S, $R[L]$ is effectively regular (Harrison, 1978). Let $R_1, R_2 \subseteq (S \times S)^*$ be regular relations that are recognized by finite state transducers $G_1 = (Q_1, S \times S, \delta_1, Q_{1,i}, Q_{1,m})$ and $G_2 = (Q_2, S \times S, \delta_2, Q_{2,i}, Q_{2,m})$, respectively. Then, their relational composition $R_1 \circ R_2$ is recognized by the finite state transducer $G_1 \circ G_2 := (Q_1 \times Q_2, S \times S, \delta_1 \circ \delta_2, Q_{1,i} \times Q_{2,i}, Q_{1,m} \times Q_{2,m})$, where $\delta_1 \circ \delta_2$ is the relation defined such that: $((q_1, q_2), (s, s''), (q_1', q_2')) \in \delta_1 \circ \delta_2$ iff there exists an $s' \in S$, such that $(q_1, (s, s'), q_1') \in \delta_1$ and $(q_2, (s', s''), q_2') \in \delta_2$. Let $R \subseteq (S \times S)^*$ be a regular relation that is recognized by finite state transducer $G = (Q, S\times S, \delta, Q_i, Q_m)$ and let $L \subseteq S^*$ be a regular language that is recognized by finite state automaton $G' = (Q', S, \delta', Q_i', Q_m')$. Then, the transduction image $R[L]$ is recognized by finite state automaton $G[G'] := (Q' \times Q, S, \delta[\delta'], Q_i' \times Q_i, Q_m' \times Q_m)$, where $\delta[\delta']$ is the relation defined such that $((q', q), s, (q_1', q_1)) \in \delta[\delta']$ iff there exists an $s' \in S$, such that $(q', s', q_1') \in \delta'$ and $(q, (s', s), q_1) \in \delta$.

3. Expressiveness characterization

In the next subsection, we define the parameterized systems studied in this work and present the symbolic setup.

3.1. System setup

Let $G = (S, \Sigma, (E_\sigma)_{\sigma\in\Sigma}, S_i, S_m)$ be a finite state automaton, where Σ is partitioned into the global event set Σ_g and the local event set Σ_l. Let G_i be an isomorphic module of G, obtained through relabeling the local event set Σ_l to $\Sigma_{i,l} := \Sigma_l \times \{i\}$ using index i (Rohloff & Lafortune, 2006; Wang et al., 2013). The parameterized system generated by G is the infinite state automaton $G^\infty = f(\bigsqcup_{n\geq 1}(\|_{i=1}^n G_i))$ over Σ, where f denotes the map that eliminates the indexes of local events from the infinite state automaton $\bigsqcup_{n\geq 1}(\|_{i=1}^n G_i)$. The synchronous product of any finite number of modules of G is contained in $\bigsqcup_{n\geq 1}(\|_{i=1}^n G_i)$, using the graph disjoint union operation $\bigsqcup$. The map f ensures that the set of events labeling the transitions of G^∞ is equal to Σ, which is finite. Note that essentially we put an infinite number of automata $f(\|_{i=1}^n G_i)$, $n \geq 1$, inside the same automaton G^∞. We call G a *globally synchronized template*, as the only synchronization between different modules is through the global events.

The states of G^∞ are encoded using strings over S, where the state $(s_1, s_2, \ldots, s_n)$ is encoded symbolically by the string $w = s_1s_2\ldots s_n$. For technical convenience, we use S^* instead of S^+ to encode the state space of G^∞. Then G^∞ is represented by the tuple $(S^*, \Sigma, (R_\sigma)_{\sigma\in\Sigma}, S_i^*, S_m^*)$, where $R_\sigma \subseteq S^* \times S^*$ encodes the (length-preserving) transition relation (on S^*) of G^∞ associated with σ. Our next result establishes the relationship between R_σ and E_σ.

Lemma 2. *For $\sigma \in \Sigma_l$, $R_\sigma = id_{S^*}E_\sigma id_{S^*}$. For $\sigma \in \Sigma_g$, $R_\sigma = E_\sigma^*$.*

Proof. For any local event σ, $(s_1s_2\ldots s_n, s_1's_2'\ldots s_m') \in R_\sigma$ *if and only if* $m = n$ and there exists an index i such that $(s_i, s_i') \in E_\sigma$ and $s_j = s_j'$ for $j \neq i$ *if and only if* $(s_1s_2\ldots s_n, s_1's_2'\ldots s_m') \in id_{S^*}E_\sigma id_{S^*}$.

For any global event σ, $(s_1s_2\ldots s_n, s_1's_2'\ldots s_m') \in R_\sigma$ *if and only if* $m = n$ and $(s_i, s_i') \in E_\sigma$ for each i *if and only if* $(s_1s_2\ldots s_n, s_1's_2'\ldots s_m') \in E_\sigma^*$. ■

Example. As an illustrative example, consider the template G_o shown in Fig. 1, where the global event set is $\Sigma_g = \{g_1, g_2, g_3\}$ and the local event set is $\Sigma_l = \{a, b, c\}$. We have, by Lemma 2,

$$E_a = \{(s_2, s_3)\}, R_a = id_{S^*}\{(s_2, s_3)\}id_{S^*};$$
$$E_b = \{(s_2, s_4)\}, R_b = id_{S^*}\{(s_2, s_4)\}id_{S^*};$$
$$E_c = \{(s_3, s_3)\}, R_c = id_{S^*}\{(s_3, s_3)\}id_{S^*};$$
$$E_{g_1} = \{(s_1, s_2)\}, R_{g_1} = \{(s_1, s_2)\}^*;$$
$$E_{g_2} = \{(s_1, s_3)\}, R_{g_2} = \{(s_1, s_3)\}^*;$$
$$E_{g_3} = \{(s_2, s_1), (s_4, s_1)\}, R_{g_3} = \{(s_2, s_1), (s_4, s_1)\}^*$$

Transition relation R_σ contains all the transitions labeled by event σ in finite state automaton $f(\|_{i=1}^n G_i)$, for any $n \geq 0$. For example, $(s_2s_2s_4, s_2s_3s_4) \in R_a$ captures the transition labeled by event a from state (s_2, s_2, s_4) to state (s_2, s_3, s_4) in $f(\|_{i=1}^3 G_i)$, and $(s_2s_4, s_3s_4) \in R_a$ captures the transition labeled by event a from state (s_2, s_4) to state (s_3, s_4) in $f(\|_{i=1}^2 G_i)$.

Each R_σ, where $\sigma \in \Sigma$, and thus the transition relation $R_{G^\infty} = \bigcup_{\sigma\in\Sigma} R_\sigma$ of G^∞, is a regular relation. One important question is whether the reachability relation $R_{G^\infty}^{*(\circ)}$ is also regular, for each globally synchronized template G. We provide an affirmative answer and study the expressive power of the reachability relations of these systems in the next subsection. Note that we do not require G to be deterministic. In fact, G^∞ is in general a non-deterministic infinite state automaton even if G is required to be deterministic. Systems with multiple templates could also be dealt with in the symbolic framework in a similar way.

3.2. Characterization result

To obtain an expressiveness characterization, we first show that for any globally synchronized template G, the reachability relation $R_{G^\infty}^{*(\circ)}$ of G^∞ is an iteration-closed star language over $S \times S$, i.e., $R_{G^\infty}^{*(\circ)} \in Star(S \times S)$ (Proposition 1). And then we show that for each iteration-closed relation T in $Star(S \times S)$, there exists a globally synchronized template G that realizes it, i.e., $R_{G^\infty}^{*(\circ)} = T$ (Proposition 2).

The following lemmas are needed in order to establish the first proposition. Lemma 3 states that for subsets of $S\times S$, Kleene closure distributes over relational composition. Lemma 4 states that $R_\sigma^{*(\circ)} \in Star(S \times S)$ for each $\sigma \in \Sigma$.

Lemma 3. *Let* $D_1, D_2, \ldots, D_n \subseteq S \times S$, *then* $D_n^* \circ \ldots \circ D_2^* \circ D_1^* = (D_n \circ \ldots \circ D_2 \circ D_1)^*$.

Proof. We prove the statement by induction on n. The statement holds trivially when $n = 1$. Suppose it holds for some $k \geq 1$ that $D_k^* \circ \ldots \circ D_2^* \circ D_1^* = (D_k \circ \ldots \circ D_2 \circ D_1)^*$. Consider the case $n = k+1$. $D_{k+1}^* \circ D_k^* \circ \ldots \circ D_2^* \circ D_1^* = D_{k+1}^* \circ (D_k \circ \ldots \circ D_2 \circ D_1)^*$ by the induction hypothesis. We only need to show that for any $D, D' \subseteq S \times S$, $D^* \circ D'^* = (D \circ D')^*$. It then follows that $D_{k+1}^* \circ (D_k \circ \ldots \circ D_2 \circ D_1)^* = (D_{k+1} \circ D_k \circ \ldots \circ D_2 \circ D_1)^*$ and the proof is done.

Let $(\alpha, \beta) \in (D \circ D')^*$. By definition, there exists some $k \in \mathbb{N}$ such that $(\alpha, \beta) = (s_1, s_1')(s_2, s_2')\ldots(s_k, s_k') \in (D \circ D')^k$. For each $(s_i, s_i') \in D \circ D'$, where $1 \leq i \leq k$, there exists some $s_i'' \in S$ such that $(s_i, s_i'') \in D$ and $(s_i'', s_i') \in D'$. Thus, $(s_1 \ldots s_{k-1}s_k, s_1'' \ldots s_{k-1}''s_k'') \in D^*$ and $(s_1'' \ldots s_{k-1}''s_k'', s_1' \ldots s_{k-1}'s_k') \in D'^*$. This implies that $(s_1 \ldots s_{k-1}s_k, s_1' \ldots s_{k-1}'s_k') \in D^* \circ D'^*$. It follows that $(\alpha, \beta) \in D^* \circ D'^*$.

Let $(\alpha, \beta) \in D^* \circ D'^*$. Then $|\alpha| = |\beta|$, since both D^* and D'^* are length-preserving. There exists some $k \in N$ such that $\alpha = s_1 s_2 \ldots s_k$ and $\beta = s_1' s_2' \ldots s_k'$. Also, there exist $s_1'', s_2'', \ldots, s_k'' \in S$ such that $(s_1 s_2 \ldots s_k, s_1'' s_2'' \ldots s_k'') \in D^*$ and $(s_1'' s_2'' \ldots s_k'', s_1' s_2' \ldots s_k') \in D'^*$. Thus $(s_i, s_i'') \in D, (s_i'', s_i') \in D'$ and then $(s_i, s_i') \in D \circ D'$. We then have $(\alpha, \beta) = (s_1 s_2 \ldots s_k, s_1' s_2' \ldots s_k') \in (D \circ D')^*$. ■

Lemma 4. *For each* $\sigma \in \Sigma$, $R_\sigma^{*(\circ)} \in Star(S \times S)$.

Proof. For $\sigma \in \Sigma_l$, $R_\sigma^{*(\circ)} = (id_{S^*} E_\sigma id_{S^*})^{*(\circ)} = (E_\sigma^{*(\circ)})^* \in Star(S \times S)$, where $\varnothing \neq id_S \subseteq E_\sigma^{*(\circ)} \subseteq S \times S$. In fact, $E_\sigma^{*(\circ)}$ is the reachability relation of G along σ-paths, i.e., the set of state tuples (s, s') of G such that s' can be reached from s by executing a finite sequence (including empty sequence) of σ's. Then $(E_\sigma^{*(\circ)})^*$ is the reachability relation of G^∞ along σ-paths, i.e., $R_\sigma^{*(\circ)} = (E_\sigma^{*(\circ)})^*$, which captures the effect of executing sequences of σ's to all the modules. For $\sigma \in \Sigma_g$, $R_\sigma^{*(\circ)} = \bigcup_{i=0}^{\infty} R_\sigma^{i(\circ)}$ corresponds to the supremum of the ascending chain $C_0 \subseteq C_1 \subseteq C_2 \subseteq \ldots$, where $C_n := \bigcup_{i=0}^{n} R_\sigma^{i(\circ)}$ for each $n \geq 0$. By Lemmas 2 and 3, we have $R_\sigma^{i(\circ)} = (E_\sigma^*)^{i(\circ)} = (E_\sigma^{i(\circ)})^*$ and clearly, $E_\sigma^{i(\circ)} \subseteq S \times S$. In fact, $E_\sigma^{i(\circ)}$ is the reachability relation of G along σ-paths of length i, i.e., the set of state tuples (s, s') of G such that s' can be reached from s by executing a finite sequence of σ's of length i. $R_\sigma^{i(\circ)} = (E_\sigma^{i(\circ)})^*$ then captures the effect that all the modules synchronously execute a finite sequence of σ's of length i. Note that $R_\sigma^{0(\circ)} = id_{S^*} = id_S^* \subseteq (S \times S)^*$ is a star language over $S \times S$, since $\varnothing \neq id_S \subseteq S \times S$. Thus, each $C_n \in Star(S \times S)$. It follows that the chain $C_0 \subseteq C_1 \subseteq C_2 \subseteq \ldots$ stabilizes, i.e., $\exists k \in \mathbb{N}, (\forall m \geq k, C_m = C_k)$, since $Star(S \times S)$ is by definition a finite set. Then $R_\sigma^{*(\circ)} = C_k \in Star(S \times S)$. Thus, for any $\sigma \in \Sigma$, $R_\sigma^{*(\circ)} \in Star(S \times S)$. ■

Example. Consider again the template G_o shown in Fig. 1. From the proof of Lemma 4, we immediately have

$$R_a^{*(\circ)} = (E_a^{*(\circ)})^* = (id_S \cup \{(s_2, s_3)\})^*;$$
$$R_b^{*(\circ)} = (E_b^{*(\circ)})^* = (id_S \cup \{(s_2, s_4)\})^*;$$
$$R_c^{*(\circ)} = (E_c^{*(\circ)})^* = id_{S^*} = id_S^*;$$
$$R_{g_1}^{*(\circ)} = (E_{g_1}^{1(\circ)})^* \cup (E_{g_1}^{0(\circ)})^* = \{(s_1, s_2)\}^* \cup id_S^*;$$
$$R_{g_2}^{*(\circ)} = (E_{g_2}^{1(\circ)})^* \cup (E_{g_2}^{0(\circ)})^* = \{(s_1, s_3)\}^* \cup id_S^*;$$
$$R_{g_3}^{*(\circ)} = (E_{g_3}^{1(\circ)})^* \cup (E_{g_3}^{0(\circ)})^* = \{(s_2, s_1) \cup (s_4, s_1)\}^* \cup id_S^*$$

Now we are ready to show Proposition 1.

Proposition 1. *For any globally synchronized template* G, $R_{G^\infty}^{*(\circ)} \in Star(S \times S)$.

Proof. Let $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_{|\Sigma|}\}$. For each i, there exists some finite $k(i)$ such that $R_{\sigma_i}^{*(\circ)} = \bigcup_{j=1}^{k(i)} D_{i,j}^*$, where $\varnothing \neq D_{i,j} \subseteq S \times S$, by Lemma 4. Using Lemma 1, $R_{G^\infty}^{*(\circ)} = (\bigcup_{\sigma \in \Sigma} R_\sigma)^{*(\circ)} = (R_{\sigma_{|\Sigma|}}^{*(\circ)} \circ R_{\sigma_{|\Sigma|-1}}^{*(\circ)} \circ \ldots \circ R_{\sigma_1}^{*(\circ)})^{*(\circ)}$ corresponds to the supremum of the ascending chain $c_0 \subseteq c_1 \subseteq c_2 \subseteq \ldots$, where $c_n := \bigcup_{i=0}^{n} (R_{\sigma_{|\Sigma|}}^{*(\circ)} \circ R_{\sigma_{|\Sigma|-1}}^{*(\circ)} \circ \ldots \circ R_{\sigma_1}^{*(\circ)})^{i(\circ)}$ for each $n \geq 0$. We observe that $id_S^* \subseteq (R_{\sigma_{|\Sigma|}}^{*(\circ)} \circ R_{\sigma_{|\Sigma|-1}}^{*(\circ)} \circ \ldots \circ R_{\sigma_1}^{*(\circ)})^{i(\circ)} \in Star(S \times S)$ for each i, using the facts that $R_{\sigma_i}^{*(\circ)} = \bigcup_{j=1}^{k(i)} D_{i,j}^*$ and relational composition distributes over union operation and using Lemma 3. Thus $c_n \in Star(S \times S)$. It follows from the fact that $Star(S \times S)$ is finite that the chain $c_0 \subseteq c_1 \subseteq c_2 \subseteq \ldots$ stabilizes, i.e., $\exists k \in N, (\forall m \geq k, c_k = c_m)$. Thus, $R_{G^\infty}^{*(\circ)} = c_k \in Star(S \times S)$. ■

The same conclusion also holds for the co-reachability relation $(R_{G^\infty}^{-1})^{*(\circ)}$ of G^∞, which is the reachability relation of $(G^T)^\infty = (G^\infty)^T$. In fact, we have $(R_{G^\infty}^{-1})^{*(\circ)} = (R_{G^\infty}^{*(\circ)})^{-1}$, since $(R_1 \circ R_2)^{-1} = R_2^{-1} \circ R_1^{-1}$ for any two relations R_1, R_2 and the inverse operation distributes over the union operation for relations.

Remark. We notice that the above proof is constructive. In general, the length $k+1$ of the stabilizing chain $c_0 \subseteq c_1 \subseteq c_2 \subseteq \ldots$ depends on the order upon which the reachability relations of G^∞ along σ-paths are composed. The optimization of this ordering is an interesting problem that can help reduce the complexity of the symbolic computation. However, this problem will not be addressed in this work due to the space limitation.

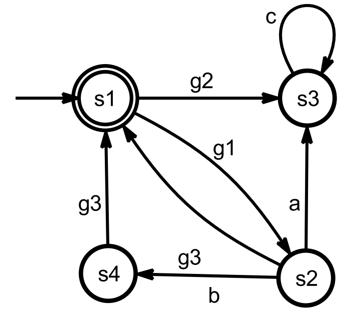


Fig. 1. Globally synchronized template G_o.

Example. We shall illustrate the use of Proposition 1 by computing the reachability relation of G_o^∞, where G_o is the globally synchronized template shown in Fig. 1.

For G_o, if we compose the reachability relations of G_o^∞ along σ-paths in the order of a, b, c, g_1, g_2, g_3, then the ascending chain in Proposition 1 turns out to stabilize at c_3. Thus, the reachability relation $R_{G_o^\infty}^{*(\circ)}$ is equal to $\bigcup_{i=0}^{3} (R_a^{*(\circ)} \circ R_b^{*(\circ)} \circ R_c^{*(\circ)} \circ R_{g_1}^{*(\circ)} \circ R_{g_2}^{*(\circ)} \circ R_{g_3}^{*(\circ)})^{i(\circ)}$, which is equal to

$$(id_S \cup \{(s_2, s_3), (s_2, s_4)\})^* \cup \{(s_2, s_1), (s_4, s_1)\}^* \cup$$
$$\{(s_1, s_2), (s_1, s_3), (s_1, s_4)\}^* \cup$$
$$\{(s_2, s_2), (s_2, s_3), (s_2, s_4), (s_4, s_2), (s_4, s_3), (s_4, s_4)\}^*$$

Thus, the co-reachability relation $(R_{G_o^\infty}^{-1})^{*(\circ)}$ is equal to

$$(id_S \cup \{(s_3, s_2), (s_4, s_2)\})^* \cup \{(s_1, s_2), (s_1, s_4)\}^* \cup$$
$$\{(s_2, s_1), (s_3, s_1), (s_4, s_1)\}^* \cup$$
$$\{(s_2, s_2), (s_3, s_2), (s_4, s_2), (s_2, s_4), (s_3, s_4), (s_4, s_4)\}^*$$

Remark. Lemma 3 can be used to facilitate the computation of $(R_a^{*(\circ)} \circ R_b^{*(\circ)} \circ R_c^{*(\circ)} \circ R_{g_1}^{*(\circ)} \circ R_{g_2}^{*(\circ)} \circ R_{g_3}^{*(\circ)})^{i(\circ)}$ for each $i \in \{1, 2, 3\}$, as in the proof of Proposition 1, and the computation is then reduced to that of relational composition of relations on S. For example,

$$R_a^{*(\circ)} \circ R_b^{*(\circ)} = (id_S \cup \{(s_2, s_3)\})^* \circ (id_S \cup \{(s_2, s_4)\})^* =$$
$$((id_S \cup \{(s_2, s_3)\}) \circ (id_S \cup \{(s_2, s_4)\}))^* =$$
$$(id_S \cup \{(s_2, s_4), (s_2, s_3)\})^*$$

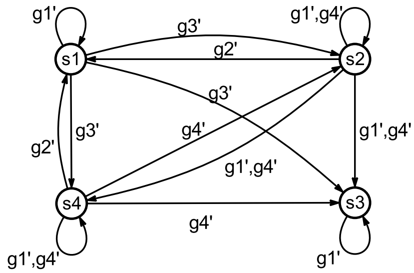


Fig. 2. Construction of G_o' using Proposition 2.

The reachability relation $R^{*(\circ)}_{G^\infty}$ of any globally synchronized template G is iteration-closed by definition and belongs to $Star(S \times S)$ by Proposition 1. In the following, we show that the reverse also holds.

Proposition 2. *For each iteration-closed $T \in Star(S \times S)$, there exists a globally synchronized template G' with only global events such that $R^{*(\circ)}_{G'^\infty} = T$.*

Proof. Suppose $T \in Star(S \times S)$ is iteration-closed. Then $T = \bigcup_{i \in I} D_i^*$ for some (non-empty) finite index set I, where $\varnothing \neq D_i \subseteq S \times S$ for each i. For each i, we create a global event g_i and its corresponding transition relation $E_{g_i} := D_i$. Let $G' = (S, \{g_i \mid i \in I\}, (E_{g_i})_{i \in I})$ be the constructed globally synchronized template, which only has global events. By Lemma 2, we have $R_{g_i} = E^*_{g_i} = D_i^*$ and thus $R^{*(\circ)}_{G'^\infty} = (\bigcup_{i \in I} R_{g_i})^{*(\circ)} = (\bigcup_{i \in I} D_i^*)^{*(\circ)} = T^{*(\circ)} = T$, since T is iteration-closed. Thus, T is realized by template G', i.e., $R^{*(\circ)}_{G'^\infty} = T$. ■

Since, for any globally synchronized template G, $R^{*(\circ)}_{G^\infty}$ is an iteration-closed star language over $S \times S$, there exists a template G' with only global events such that $R^{*(\circ)}_{G'^\infty} = R^{*(\circ)}_{G^\infty}$. This is stated in Corollary 1.

Corollary 1. *For any globally synchronized template G, there exists a globally synchronized template G' such that G' has only global events and $R^{*(\circ)}_{G^\infty} = R^{*(\circ)}_{G'^\infty}$.*

Example. For template G_o, the equivalent template G_o', according to the construction in Proposition 2, is shown in Fig. 2. From the proof of Proposition 2, we create a g_i' for each D_i in $R^{*(\circ)}_{G_o^\infty}$ so that $E'_{g_i} = D_i$. Thus, we have,

$$E_{g_1'} = id_S \cup \{(s_2, s_3), (s_2, s_4)\},$$
$$E_{g_2'} = \{(s_2, s_1), (s_4, s_1)\},$$
$$E_{g_3'} = \{(s_1, s_2), (s_1, s_3), (s_1, s_4)\},$$
$$E_{g_4'} = \{(s_2, s_2), (s_2, s_3), (s_2, s_4), (s_4, s_2), (s_4, s_3), (s_4, s_4)\}$$

We have ignored the initial state and the marked state of G_o', since it is not important in the computation of the reachability relation.

Essentially, the above corollary states that the class of globally synchronized templates with only global events is as expressive as the class of globally synchronized templates with both global events and local events.[2] Finally, to complete the expressiveness characterization, we need the following lemma whose proof is straightforward.

[2] We here remark that the class of templates with only local events is strictly less expressive.

Lemma 5. *For any finite alphabet S and any relation $T \subseteq (S \times S)^*$, $id_{S^*} \subseteq T$ and $T \circ T = T$ iff $T^{*(\circ)} = T$.*

Proof. Suppose $T^{*(\circ)} = T$. Then $id_{S^*} \subseteq T^{*(\circ)} = T$ and $T \circ T = T^{*(\circ)} \circ T^{*(\circ)} = T^{*(\circ)} = T$.

Suppose $id_{S^*} \subseteq T$ and $T \circ T = T$. Then $T^{i(\circ)} = T$ for any $i \geq 1$. Thus, $T^{*(\circ)} = \bigcup_{i=0}^{\infty} T^{i(\circ)} = id_{S^*} \cup T = T$, since $id_{S^*} \subseteq T$. ■

Combining the above results, we have our main characterization theorem below, which is quite straightforward.

Theorem 1. *For any relation $T \subseteq (S \times S)^*$, the following four statements are equivalent:*
(1) T is realizable by a globally synchronized template G with only global events.
(2) T is realizable by a globally synchronized template G.
(3) $T \in Star(S \times S)$ and $T^{*(\circ)} = T$.
(4) $T \in Star(S \times S)$, $id_{S^*} \subseteq T$ and $T \circ T = T$.

For an arbitrary regular relation $T \subseteq (S \times S)^*$, it is decidable whether $T \in Star(S \times S)$. By Theorem 1, we immediately have the following corollary.

Corollary 2. *It is decidable whether an arbitrary regular relation $T \subseteq (S \times S)^*$ is realizable by a globally synchronized template.*

4. Deadlocked states and blocked states

In this section, we shall explain how to compute the sets of deadlocked states and blocked states, as an application of the symbolic reachability analysis. These results will be needed in the entrance control problem, to be discussed in Section 5. We remark that these sets are commutative languages due to the property of symmetry, i.e., the relative positions of modules in the string encoding do not matter.

4.1. Computing deadlocked states

A state $w \in S^*$ of G^∞ is said to be deadlocked if there is no event that can be executed at state w, i.e., $R_\sigma[\{w\}] = \varnothing$ for any $\sigma \in \Sigma$. The set of deadlocked states is denoted by $L_{G^\infty,D}$. It is not difficult to see that the set $L_{G^\infty,ND} = L^c_{G^\infty,D}$ of non-deadlocked states is a regular language over S. Indeed,

$$L_{G^\infty,ND} := S^*(\bigcup_{\sigma \in \Sigma_l} S(\sigma))S^* \cup (\bigcup_{\sigma \in \Sigma_g} S(\sigma)^*)$$

Intuitively, each state in $S^*(\bigcup_{\sigma \in \Sigma_l} S(\sigma))S^*$ can execute at least one local event and each state in $\bigcup_{\sigma \in \Sigma_g} S(\sigma)^*$ is able to execute at least one global event.

Example. For G_o, the set of deadlocked states of G_o^∞ is $L_{G_o^\infty,D} = L^c_{G_o^\infty,ND} = (S^*\{s_2, s_3\}S^* \cup \{s_1\}^* \cup \{s_2, s_4\}^*)^c =$

$\{s_1, s_4\}^*\{s_1\}\{s_1, s_4\}^*\{s_4\}\{s_1, s_4\}^* \cup$
$\{s_1, s_4\}^*\{s_4\}\{s_1, s_4\}^*\{s_1\}\{s_1, s_4\}^*$.

4.2. Computing blocked states

A state w of G^∞ is said to be blocked if $R^{*(\circ)}_{G^\infty}[\{w\}] \cap S^*_m = \varnothing$. The set of blocked states is denoted by $L_{G^\infty,B}$. Since the set of co-reachable, or equivalently, the set of non-blocked states of G^∞ is $L_{G^\infty,NB} := L^c_{G^\infty,B} = (R^{-1}_{G^\infty})^{*(\circ)}[S^*_m]$, the set of blocked states of G^∞ is $L_{G^\infty,B} = ((R^{-1}_{G^\infty})^{*(\circ)}[S^*_m])^c$.

Example. For G_o, the set of co-reachable states of G_o^∞ is $L_{G^\infty,NB} = (R^{*(\circ)}_{G_o^\infty})^{-1}[S_m^*] =$

$$\begin{aligned}&(id_S \cup \{(s_3, s_2), (s_4, s_2)\})^*[\{s_1\}^*] \cup \\ &\{(s_1, s_2), (s_1, s_4)\}^*[\{s_1\}^*] \cup \\ &\{(s_2, s_1), (s_3, s_1), (s_4, s_1)\}^*[\{s_1\}^*] \cup \\ &\{(s_2, s_2), (s_3, s_2), (s_4, s_2), (s_2, s_4), (s_3, s_4), (s_4, s_4)\}^*[\{s_1\}^*] \\ &= \{s_1\}^* \cup \{s_2, s_4\}^*.\end{aligned}$$

The set of blocked states of G_o^∞ is $L_{G_o^\infty,B} = L^c_{G^\infty,NB} = (\{s_1\}^* \cup \{s_2, s_4\}^*)^c =$

$$\begin{aligned}&\{s_3\}\{s_1, s_2, s_3, s_4\}^* \cup \{s_1\}\{s_1\}^*\{s_2, s_3, s_4\}\{s_1, s_2, s_3, s_4\}^* \cup \\ &\{s_2, s_4\}\{s_2, s_4\}^*\{s_1, s_3\}\{s_1, s_2, s_3, s_4\}^*\end{aligned}$$

Remark. The computation of the above transduction image is straightforward, following the same spirit in Lemma 3. Thus, we have, for example, $\{(s_1, s_2), (s_1, s_4)\}^*[\{s_1\}^*] = (\{(s_1, s_2), (s_1, s_4)\}[\{s_1\}])^* = \{s_2, s_4\}^*$. One can also perform the computation of transduction image using transducers and automata, as discussed in the Preliminaries.

Remark. The set of reachable states can be computed in a similar way. Thus, the above analysis can be used for computing the sets of reachable deadlocked states and blocked states. A technique based on weak invariant simulation (Zibaeenejad & Thistle, 2014) has also been used for computing the sets of reachable deadlocked states for parameterized ring networks.

4.3. Extension to systems with idle modules

In this subsection, we consider systems where each module is either actively participating in the system evolution, i.e., at a template state, or residing in the idle state. An idle module can enter the system evolution and become active. Without loss of generality, we assume that idle modules could only become active by entering the system evolution from the set S_i of initial template states.

We use the new symbol $\# \notin \Sigma \cup S$ to represent the idle state. Then the (length-preserving) transition relation associated with the entrance of an idle module in the template state $s \in S_i$ is

$$R_{enter_s} = \{(w_1\#w_2, w_1sw_2) \mid w_1, w_2 \in (S \cup \{\#\})^*\} = id_{(S\cup\{\#\})^*}\{(\#, s)\}id_{(S\cup\{\#\})^*}$$

and then the (length-preserving) transition relation associated with the entrance of an idle module is

$$R_{enter} = \bigcup_{s\in S_i} R_{enter_s} = id_{(S\cup\{\#\})^*}\{(\#, s) \mid s \in S_i\}id_{(S\cup\{\#\})^*}$$

We have $R^{*(\circ)}_{enter} = (id_{S\cup\{\#\}} \cup \{(\#, s) \mid s \in S_i\})^*$ following the proof of Lemma 4. We observe that $R^{*(\circ)}_{enter}$ is a star language over $(S \cup \{\#\}) \times (S \cup \{\#\})$. Furthermore, in this system setup, each D^* in $R^{*(\circ)}_\sigma$ (see Lemma 4), where $\sigma \in \Sigma$ and $\varnothing \neq D \subseteq S \times S$, is replaced by $(D \cup \{(\#, \#)\})^*$ over $(S \cup \{\#\}) \times (S \cup \{\#\})$. Clearly, following the proof of Proposition 1, the reachability relation $T_\# = (\bigcup_{\sigma\in\Sigma} R_\sigma \cup R_{enter})^{*(\circ)}$ of the system is effectively a star language over $(S \cup \{\#\}) \times (S \cup \{\#\})$. Essentially, the reachability relation $T_\#$ is realized by the template $G_\# := (S \cup \{\#\}, \Sigma \cup \{enter_s \mid s \in S_i\}, \delta \cup \{(\#, enter_s, s) \mid s \in S_i\} \cup \{(\#, \sigma, \#) \mid \sigma \in \Sigma_g\})$, where $enter_s$ is a new local event, for each $s \in S_i$. The transitions $(\#, \sigma, \#)$'s are added so that the global events are not blocked by modules that are in the idle state. The sets of initial states and marked states of $G_\#$ are then $S_i \cup \{\#\}$ and $S_m \cup \{\#\}$, respectively. For deadlock and blocking analysis for systems with idle modules, we only need to analyze the template $G_\#$ as in Section 4.

The next result presents a simple connection between the sets of deadlocked states and blocked states of G^∞ and $G_\#^\infty$, whose proof is straightforward.

Proposition 3. *For any globally synchronized template G, we have $L_{G_\#^\infty,D} = L_{G^\infty,D}$ and $L_{G_\#^\infty,B} = L_{G^\infty,B} \sqcup \{\#\}^*$.*

Proof. Since any $s \in (S\cup\{\#\})^*\{\#\}(S\cup\{\#\})^*$ cannot be a deadlocked state, it is clear that $L_{G_\#^\infty,D} = L_{G^\infty,D} \subseteq S^*$. We now show that $L_{G_\#^\infty,B} \subseteq L_{G^\infty,B} \sqcup \{\#\}^*$. Let $w \in L_{G_\#^\infty,B} \subseteq (S \cup \{\#\})^*$ be any blocked state of $G_\#^\infty$. There exist some $k \geq 0$ and $w' \in S^*$ such that $w \in w' \sqcup \#^k$, since $w \in (S \cup \{\#\})^*$. It is clear that $w' \in L_{G^\infty,B}$, since otherwise we must have $w \notin L_{G_\#^\infty,B}$, for # is already a marked state and any global event is defined at state #. Thus, $w \in L_{G^\infty,B} \sqcup \{\#\}^*$. On the other hand, we have $L_{G^\infty,B} \sqcup \{\#\}^* \subseteq L_{G_\#^\infty,B}$. Indeed, let w' be any blocked state in G^∞, i.e., $w' \in L_{G^\infty,B}$, and let $k \geq 0$ be any non-negative integer, then we claim that $w' \sqcup \#^k \subseteq L_{G_\#^\infty,B}$. Indeed, if some $w \in w' \sqcup \#^k$ is not blocked in $G_\#^\infty$, then w', which corresponds to the removal of idle modules from w, is not blocked in G^∞, which results in a contradiction. ■

Example. Consider the template $G_{o,\#}$, we have $L_{G_{o,\#}^\infty,D} = L_{G^{o,\infty},D} =$

$$\begin{aligned}&\{s_1, s_4\}^*\{s_1\}\{s_1, s_4\}^*\{s_4\}\{s_1, s_4\}^* \cup \\ &\{s_1, s_4\}^*\{s_4\}\{s_1, s_4\}^*\{s_1\}\{s_1, s_4\}^*.\end{aligned}$$

$$\begin{aligned}L_{G_{o,\#}^\infty,B} = L_{G^{o,\infty},B} \sqcup \{\#\}^* = \\ \{\#\}^*\{s_3\}\{s_1, s_2, s_3, s_4, \#\}^* \cup \\ \{\#\}^*\{s_1\}\{s_1, \#\}^*\{s_2, s_3, s_4\}\{s_1, s_2, s_3, s_4, \#\}^* \cup \\ \{\#\}^*\{s_2, s_4\}\{s_2, s_4, \#\}^*\{s_1, s_3\}\{s_1, s_2, s_3, s_4, \#\}^*\end{aligned}$$

5. Entrance control for systems with idle modules

In the rest of this section, we shall investigate the problem of entrance control for systems with idle modules, to ensure deadlock freeness and blocking freeness. The problem setup is introduced below.

Entrance Control: Given any globally synchronized template $G = (S, \Sigma, \delta, S_i, S_m)$ over Σ. In order to study the entrance control problem, we shall consider the template $G_\#$ which generates the system $G_\#^\infty$ with idle modules. Let the set of uncontrollable events be Σ and let the set of controllable events be $\{enter_s \mid s \in S_i\}$. Intuitively, the supervisor can only control the entrance events. The goal now is to compute an entrance control function, or an entrance supervisor, $\mathcal{C} : (S \cup \{\#\})^*\{\#\}(S \cup \{\#\})^* \mapsto 2^{\{enter_s|s\in S_i\}}$ that disables those entrance events in $\mathcal{C}(w)$ for each state $w \in (S \cup \{\#\})^*\{\#\}(S \cup \{\#\})^*$ of $G_\#^\infty$, so that the closed-loop system has the desirable property of being deadlock free or blocking free. Here, the closed-loop system is defined to be the reachable part of the infinite state automaton, obtained from $G_\#^\infty$ by removing the outgoing transitions labeled by events in $\mathcal{C}(w)$ at each state $w \in (S\cup\{\#\})^*\{\#\}(S\cup\{\#\})^*$. It is sufficient to consider disabling entrance events, instead of entrance transitions, in the (non-deterministic) infinite state automaton $G_\#^\infty$, due to the property of symmetry, i.e., the relative positions of modules in the string encoding do not matter in the analysis of deadlock freeness and blocking freeness properties. If an transition labeled by $enter_s$ is bad for a state w, then all the transitions labeled by $enter_s$ are bad for state w.

The control strategy is standard. We repeat the following steps until there is no bad state in the resulting system:

(1) identify the set of bad states in the resulting system
(2) remove the uncontrollable predecessors of the set of bad states, i.e., remove the set L_n of states that can reach the set of bad states along paths of uncontrollable events

The set of states that is removed in this iterative process is $L^{\uparrow}_{bad} := \bigcup_{n\geq 0} L_n$. The entrance supervisor that removes exactly $L^{\uparrow}_{bad}$ is maximally permissive. Moreover, $L^{\uparrow}_{bad}$ is all the information that is needed for computing the maximally permissive entrance control function $\mathcal{C}_{\uparrow}$, given the template $G_{\#}$ (more details will be provided in Section 5.3). However, the main difficulty lies in finitely computing $L^{\uparrow}_{bad}$, since we essentially deal with infinite state automata. We shall provide the exact constructions of $L^{\uparrow}_{bad}$ and reason about their termination properties in Sections 5.1 and 5.2, for deadlock freeness and blocking freeness properties, respectively. We are also interested in the symbolic encoding of maximally permissive entrance supervisors using finite state automata, which will be discussed in Section 5.3.

5.1. Deadlock freeness entrance control

In this subsection, we show that for deadlock freeness entrance control, $L^{\uparrow}_{bad}$ is effectively regular. To that end, we need to first reason about L_n, for each $n \geq 0$, based on the definition. Let $L_0 := (\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[L_{G^{\infty}_{\#},D}]$ be the set of states that can uncontrollably evolve to a deadlocked state in system $G^{\infty}_{\#}$. Clearly, L_0 has to be removed from $G^{\infty}_{\#}$. It follows from $L_{G^{\infty}_{\#},D} = L_{G^{\infty},D} \subseteq S^*$ that $L_0 \subseteq S^*$. Then, the set of newly created deadlocked states is $(L_{G^{\infty},D} \sqcup \{\#\}) \cap \bigcap_{s\in S_i} R^{-1}_{enters}[L_0]$. Intuitively, a state $w \in (S \cup \{\#\})^*$ could become deadlocked after removing L_0 if and only if at state w the system can only execute entrance events, i.e., $w \in L_{G^{\infty},D} \sqcup \{\#\}$, and all the entrance transitions defined at state w would lead to L_0 and thus are disabled, i.e., $\forall s \in S_i, R_{enters}[\{w\}] \subseteq L_0$. Since $R_{enters}[\{w\}]$ is a singleton for any $w \in L_{G^{\infty},D} \sqcup \{\#\}$, we have $\forall s \in S_i, R_{enters}[\{w\}] \subseteq L_0$ iff $\forall s \in S_i, R_{enters}[\{w\}] \cap L_0 \neq \varnothing$. We conclude that $w \in \bigcap_{s\in S_i} R^{-1}_{enters}[L_0]$. Then $L_1 :=$

$$(\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[(L_{G^{\infty},D} \sqcup \{\#\}) \cap \bigcap_{s\in S_i} R^{-1}_{enters}[L_0]]$$

is the set of states that can uncontrollably evolve to a newly created deadlocked state. Clearly, we have $L_1 \subseteq S^* \sqcup \{\#\}$. After removing L_1, new deadlocked states are again created and thus have to be removed in exactly the same manner. In general, we have the following formula that recursively defines the set L_n to be removed at each iteration, which is the uncontrollable predecessors of the set of newly created deadlocked states, based on the definition of L_n in the standard procedure. We here shall remark that L_n is commutative, for each $n \geq 0$, due to the property of symmetry, i.e., the relative positions of modules do not matter. $L^{\uparrow}_{bad} = \bigcup_{n\geq 0} L_n$ is also commutative.

Proposition 4. *For each $n \geq 0$, the uncontrollable predecessors of the set of newly created deadlocked states in the $(n+1)$th iteration is $L_{n+1} =$*

$$(\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[(L_{G^{\infty},D} \sqcup \{\#\}^{n+1}) \cap \bigcap_{s\in S_i} R^{-1}_{enters}[L_n]] \subseteq S^* \sqcup \{\#\}^{n+1}$$

Proof. We prove the statement by induction on n. The statement holds for $n = 0$, as we have shown before. Suppose it holds for some $k \geq 0$ that $L_{k+1} =$

$$(\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[(L_{G^{\infty},D} \sqcup \{\#\}^{k+1}) \cap \bigcap_{s\in S_i} R^{-1}_{enters}[L_k]] \subseteq S^* \sqcup \{\#\}^{k+1}$$

Now consider the case when $n = k+1$. Intuitively, a state $w \in (S \cup \{\#\})^*$ could become deadlocked after removing L_{k+1}, after having removed $L_0, L_1, \ldots, L_k$, if and only if (1) at state w the system can only execute entrance events and enter L_{k+1}, i.e., $w \in L_{G^{\infty},D} \sqcup \{\#\}^{k+2}$, by the induction hypothesis that $L_{k+1} \subseteq S^* \sqcup \{\#\}^{k+1}$, and (2) all the entrance transitions defined at state w would lead to L_{k+1} and thus are disabled, i.e., $\forall s \in S_i, R_{enters}[\{w\}] \subseteq L_{k+1}$. Since L_{k+1} is commutative, $R_{enters}[\{w\}] \subseteq L_{k+1}$ iff $R_{enters}[\{w\}] \cap L_{k+1} \neq \varnothing$. We then conclude that $w \in \bigcap_{s\in S_i} R^{-1}_{enters}[L_{k+1}]$. The set of newly created deadlocked states is $(L_{G^{\infty},D} \sqcup \{\#\}^{k+2}) \cap \bigcap_{s\in S_i} R^{-1}_{enters}[L_{k+1}]$. Then $L_{k+2} =$

$$(\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[(L_{G^{\infty},D} \sqcup \{\#\}^{k+2}) \cap \bigcap_{s\in S_i} R^{-1}_{enters}[L_{k+1}]] \subseteq S^* \sqcup \{\#\}^{k+2}$$

This completes the inductive step. ■

The above formula for L_n is obtained by following the definition. In its present form, however, it cannot be directly used for computing $L^{\uparrow}_{bad}$. We shall simplify it and present a closed formula for L_n, which can then be used for finitely computing $L^{\uparrow}_{bad}$.

Lemma 6. *If $L_1 \neq \varnothing$, then, for each $n \geq 1$,*

$$L_n = (\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[L_{G^{\infty},D} \sqcup \{\#\}^n]$$

Proof. Suppose $L_1 \neq \varnothing$, then there exists $w \in L_{G^{\infty},D}$ and some $w_1 \in w \sqcup \#$ such that $\forall s \in S_i, R_{enters}[\{w_1\}] \cap L_0 \neq \varnothing$. Due to the property of symmetry, we can assume $w_1 = w\#$. Then, we have $R_{enters}[\{w_1\}] = \{ws\}$. It follows that $\forall s \in S_i, (\bigcup_{\sigma\in\Sigma} R_{\sigma})^{*(\circ)}[\{ws\}] \cap L_{G^{\infty},D} \neq \varnothing$. Since $w \in L_{G^{\infty},D}$, we have

$$(\bigcup_{\sigma\in\Sigma} R_{\sigma})^{*(\circ)}[\{ws\}] = \{w\}(\bigcup_{\sigma\in\Sigma_l} R_{\sigma})^{*(\circ)}[\{s\}] = \{w\}(\bigcup_{\sigma\in\Sigma_l} E_{\sigma})^{*(\circ)}[\{s\}]$$

Intuitively, since $w \in L_{G^{\infty},D}$, only sequences of local event transitions may be executed when at state $ws \in S^*$, and furthermore, only the last module may execute sequences of local event transitions. It follows that for each $s \in S_i$, there exists some $s' \in (\bigcup_{\sigma\in\Sigma_l} E_{\sigma})^{*(\circ)}[\{s\}]$ such that $ws' \in L_{G^{\infty},D}$. Clearly, there is no local event transition defined at state s'. Thus, $L_1 \neq \varnothing$ implies that, for each $s \in S_i$, there exists a (possibly empty) sequence of local event transitions that lead to a state s', where there is no local event transition defined. Now, to show $L_n = (\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[L_{G^{\infty},D} \sqcup \{\#\}^n]$, by Proposition 4, we only need to show that for each $n \geq 1$, $L_{G^{\infty},D} \sqcup \{\#\}^n \subseteq \bigcap_{s\in S_i} R^{-1}_{enters}[L_{n-1}]$. The proof can be carried out by induction on n.

Now we first show that $L_{G^{\infty},D} \sqcup \{\#\} \subseteq \bigcap_{s\in S_i} R^{-1}_{enters}[L_0]$. Indeed, for any $w' \in L_{G^{\infty},D}$ and for any $s \in S_i$, we have

$$(\bigcup_{\sigma\in\Sigma} R_{\sigma})^{*(\circ)}[\{w's\}] = \{w'\}(\bigcup_{\sigma\in\Sigma_l} E_{\sigma})^{*(\circ)}[\{s\}]$$

Thus, $w's' \in (\bigcup_{\sigma\in\Sigma} R_{\sigma})^{*(\circ)}[\{w's\}]$, where s' has been defined in the last paragraph. Furthermore, since $w' \in L_{G^{\infty},D}$ and there is no local event transition defined at state s', we conclude that $w's' \in L_{G^{\infty},D}$. Thus, $(\bigcup_{\sigma\in\Sigma} R_{\sigma})^{*(\circ)}[\{w's\}] \cap L_{G^{\infty},D} \neq \varnothing$. It follows from the above analysis that $w'\# \in \bigcap_{s\in S_i} R^{-1}_{enters}[L_0]$. since $w' \in L_{G^{\infty},D}$ is arbitrary, due to the property of symmetry, we conclude that $L_{G^{\infty},D} \sqcup \{\#\} \subseteq \bigcap_{s\in S_i} R^{-1}_{enters}[L_0]$ and thus $L_1 = (\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[L_{G^{\infty},D} \sqcup \{\#\}]$. This corresponds to the base case when $n = 1$.

Suppose it holds for some $k \geq 1$ that

$$(L_{G^{\infty},D} \sqcup \{\#\}^k) \subseteq \bigcap_{s\in S_i} R^{-1}_{enters}[L_{k-1}] \quad \text{and}$$

$$L_k = (\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[L_{G^{\infty},D} \sqcup \{\#\}^k].$$

Now consider the case when $n = k+1$. For each $w' \in L_{G^{\infty},D}$ and each $s \in S_i$, there exists some $s' \in (\bigcup_{\sigma\in\Sigma_l} E_{\sigma})^{*(\circ)}[\{s\}]$ such that $w's' \in L_{G^{\infty},D}$, as has been shown before under the assumption $L_1 \neq \varnothing$. Thus, $w's'\#^k \in L_{G^{\infty},D} \sqcup \{\#\}^k$. It then follows that $(\bigcup_{\sigma\in\Sigma} R_{\sigma})^{*(\circ)}[\{w's\#^k\}] \cap (L_{G^{\infty},D} \sqcup \{\#\}^k) \neq \varnothing$ and thus $w's\#^k \in (\bigcup_{\sigma\in\Sigma} R^{-1}_{\sigma})^{*(\circ)}[L_{G^{\infty},D} \sqcup \{\#\}^k] = L_k$, by induction hypothesis. Thus,

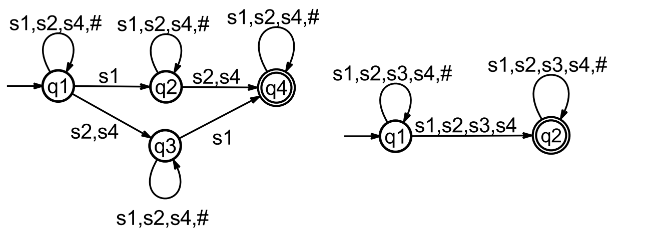


Fig. 3. The representation of $L_{bad}^{\uparrow}$ for $G_{o,\#}^{\infty}$: (left) deadlock freeness entrance control; (right) blocking freeness entrance control.

for each $w' \in L_{G^\infty,D}$ and each $s \in S_i$, $w'\#^{k+1} \in R_{enter_s}^{-1}[\{w's\#^k\}] \subseteq R_{enter_s}^{-1}[L_k]$. That is, $w'\#^{k+1} \in \bigcap_{s\in S_i} R_{enter_s}^{-1}[L_k]$. Due to the arbitrary choice of $w' \in L_{G^\infty,D}$ and the property of symmetry, we have

$$(L_{G^\infty,D} \sqcup \{\#\}^{k+1}) \subseteq \bigcap_{s\in S_i} R_{enter_s}^{-1}[L_k] \quad \text{and thus}$$

$$L_{k+1} = (\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G^\infty,D} \sqcup \{\#\}^{k+1}]$$

This completes the inductive step. ■

Notice that, according to Lemma 6, in general the standard procedure for computing $L_{bad}^{\uparrow}$ does not terminate even for the case of deadlock freeness entrance control, since $L_n \neq \varnothing$ for each $n \geq 0$ if $L_1 \neq \varnothing$. Fortunately, the formula of L_n based on Lemma 6 does have a simple pattern that allows for a closed formula for $L_{bad}^{\uparrow}$.

Proposition 5. *If $L_1 = \varnothing$, then*

$$L_{bad}^{\uparrow} = L_0 = (\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G^\infty,D}].$$

Otherwise, $L_{bad}^{\uparrow} = (\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G^\infty,D} \sqcup \{\#\}^*]$.

Proof. If $L_1 = \varnothing$, then clearly the standard procedure terminates and then $L_{bad}^{\uparrow} = L_0 = (\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G^\infty,D}]$. If $L_1 \neq \varnothing$, then $L_{bad}^{\uparrow} = L_0 \cup \bigcup_{n=1}^{\infty} L_n =$

$$(\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G^\infty,D}] \cup (\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G^\infty,D} \sqcup \{\#\}^+]$$
$$= (\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G^\infty,D} \sqcup \{\#\}^*].$$
■

The standard control procedure does not terminate if and only if $L_1 \neq \varnothing$. It is helpful to characterize the condition $L_1 \neq \varnothing$ as follows.

Proposition 6. *$L_1 \neq \varnothing$ if and only if $L_{G^\infty,D} \neq \varnothing$ and for each $s \in S_i$, there exists a (possibly empty) sequence of local event transitions that lead to some state s' in G, where there is no local event transition defined.*

Proof. only if: this has already been shown in the proof of Lemma 6. Note that $L_1 \neq \varnothing$ implies that $L_{G^\infty,D} \neq \varnothing$.

if: suppose $L_1 = \varnothing$. Then, by definition, we have $(L_{G^\infty,D} \sqcup \{\#\}) \cap \bigcap_{s\in S_i} R_{enter_s}^{-1}[L_0] = \varnothing$. Suppose $L_{G^\infty,D} \neq \varnothing$. Now, we let $w \in L_{G^\infty,D}$ be some deadlocked state. Then, we have $w\# \notin \bigcap_{s\in S_i} R_{enter_s}^{-1}[L_0]$. Then, there exists some $s \in S_i$, $ws \notin L_0$. That is, $(\bigcup_{\sigma\in\Sigma} R_\sigma)^{*(\circ)}[\{ws\}] \cap L_{G^\infty,D} = \varnothing$. We have $\{w\}(\bigcup_{\sigma\in\Sigma_l} E_\sigma)^{*(\circ)}[\{s\}] \cap L_{G^\infty,D} = \varnothing$. This implies that, each local event transition is defined at each state in $(\bigcup_{\sigma\in\Sigma_l} E_\sigma)^{*(\circ)}[\{s\}]$. In other words, for any sequence of local event transitions defined at s, there is always a local event transition defined at the state that is reached. ■

Example. For $G_{o,\#}^{\infty}$, we note that $s_1s_4 \in L_{G_o^\infty,D}$ and thus $s_1s_4\# \in L_{G_o^\infty,D} \sqcup \{\#\}$. Also, $R_{enter_s}[\{s_1s_4\#\}] = s_1s_4s_1 \in L_{G_o^\infty,D}$ and thus we have $s_1s_4\# \in \bigcap_{s\in S_i} R_{enter_s}^{-1}[L_{G_o^\infty,D}] \subseteq \bigcap_{s\in S_i} R_{enter_s}^{-1}[L_0]$. Then, we conclude $s_1s_4\# \in (L_{G_o^\infty,D} \sqcup \{\#\}) \cap \bigcap_{s\in S_i} R_{enter_s}^{-1}[L_0]$. Thus, we conclude $s_1s_4\# \in L_1$ and $L_1 \neq \varnothing$. Alternatively, from Proposition 6, we conclude that $L_1 \neq \varnothing$, since there is no local event transition that is defined at state s_1 in G_o and $G_{o,\#}^{\infty} \neq \varnothing$. By Proposition 5, we conclude that $L_{bad}^{\uparrow} =$

$$(\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G_o^\infty,D} \sqcup \{\#\}^*] = ((id_{S\cup\{\#\}} \cup$$
$$\{(s_3,s_2),(s_4,s_2)\})^* \cup \{(s_1,s_2),(s_1,s_4),(\#,\#)\}^* \cup$$
$$\{(s_2,s_1),(s_3,s_1),(s_4,s_1),(\#,\#)\}^* \cup$$
$$\{(s_2,s_2),(s_3,s_2),(s_4,s_2),(s_2,s_4),(s_3,s_4),(s_4,s_4),(\#,\#)\}^*)$$
$$[\{s_1,s_4,\#\}^*\{s_1\}\{s_1,s_4,\#\}^*\{s_4\}\{s_1,s_4,\#\}^* \cup$$
$$\{s_1,s_4,\#\}^*\{s_4\}\{s_1,s_4,\#\}^*\{s_1\}\{s_1,s_4,\#\}^*] =$$
$$\{s_1,s_2,s_4,\#\}^*\{s_1\}\{s_1,s_2,s_4,\#\}^*\{s_2,s_4\}\{s_1,s_2,s_4,\#\}^* \cup$$
$$\{s_1,s_2,s_4,\#\}^*\{s_2,s_4\}\{s_1,s_2,s_4,\#\}^*\{s_1\}\{s_1,s_2,s_4,\#\}^*$$

The automaton representation of $L_{bad}^{\uparrow}$ for the deadlock freeness entrance control of $G_{o,\#}^{\infty}$ is shown in the left of Fig. 3.

5.2. Blocking freeness entrance control

In this subsection, we show that for blocking freeness entrance control, $L_{bad}^{\uparrow}$ is also effectively regular.

Let $L_0 := (\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G_\#^\infty,B}]$ be the set of states that can uncontrollably evolve to a blocked state in $G_\#^\infty$. Clearly, L_0 has to be removed from $G_\#^\infty$. It turns out that after removing L_0 from $G_\#^\infty$ the resultant infinite state automaton is blocking free. This is because disabling entrance transitions will never create new blocking states. Intuitively, staying at the idle state is desirable since it is already a marked state. Thus, $L_{bad}^{\uparrow} = L_0 = (\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G_\#^\infty,B}]$ is the set of states that have to be removed to ensure blocking freeness. Clearly, $L_{bad}^{\uparrow}$ is commutative.

Example. For $G_{o,\#}^{\infty}$, we have $L_{bad}^{\uparrow} =$

$$(\bigcup_{\sigma\in\Sigma} R_\sigma^{-1})^{*(\circ)}[L_{G_{o,\#}^\infty,B}] = ((id_{S\cup\{\#\}} \cup$$
$$\{(s_3,s_2),(s_4,s_2)\})^* \cup \{(s_1,s_2),(s_1,s_4),(\#,\#)\}^* \cup$$
$$\{(s_2,s_1),(s_3,s_1),(s_4,s_1),(\#,\#)\}^* \cup$$
$$\{(s_2,s_2),(s_3,s_2),(s_4,s_2),(s_2,s_4),(s_3,s_4),(s_4,s_4),(\#,\#)\}^*)$$
$$[\{\#\}^*\{s_3\}\{s_1,s_2,s_3,s_4,\#\}^* \cup$$
$$\{\#\}^*\{s_1\}\{s_1,\#\}^*\{s_2,s_3,s_4\}\{s_1,s_2,s_3,s_4,\#\}^* \cup$$
$$\{\#\}^*\{s_2,s_4\}\{s_2,s_4,\#\}^*\{s_1,s_3\}\{s_1,s_2,s_3,s_4,\#\}^*] =$$
$$\{s_1,s_2,s_3,s_4,\#\}^*\{s_1,s_2,s_3,s_4\}\{s_1,s_2,s_3,s_4,\#\}^*$$

The automaton representation of $L_{bad}^{\uparrow}$ for the blocking freeness entrance control of $G_{o,\#}^{\infty}$ is shown in the right of Fig. 3.

5.3. Encoding of entrance control function

In this subsection, we shall explain how to encode the maximally permissive entrance supervisors using finite state automata, assuming $L_{bad}^{\uparrow}$ is effectively a commutative regular language. The commutativity assumption indeed holds for most realistic properties, due to symmetry, including the deadlock freeness and blocking freeness properties.

It is apparent that entrance control only needs to be applied at the set $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad}$ and remove the set of entrance transitions into $L^{\uparrow}_{bad}$. Instead of the domain $(S\cup\{\#\})^*\{\#\}(S\cup\{\#\})^*$, it is sufficient to consider the domain of the maximally permissive entrance control function $\mathcal{C}_{\uparrow}$ to be $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad}$. The maximally permissive entrance control function is then

$$\mathcal{C}_{\uparrow} : (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad} \mapsto 2^{\{enters_s|s\in S_i\}}$$

where $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad} \subseteq (S\cup\{\#\})^*\{\#\}(S\cup\{\#\})^*$. And, for each $w \in (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad}$, we have $\mathcal{C}_{\uparrow}(w) = \{enters_s \mid R_{enters_s}[\{w\}] \cap L^{\uparrow}_{bad} \neq \varnothing\}$. There are two aspects of the maximally permissive entrance control function that need to be finitely encoded, namely, the domain and the function value defined at each element of the domain. It is more convenient to enlarge the domain and encode the maximally permissive entrance control function as

$$\mathcal{C}_{\uparrow} : (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] \mapsto 2^{\{enters_s|s\in S_i\}},$$

by ignoring the definition of $\mathcal{C}_{\uparrow}$ at $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] \cap L^{\uparrow}_{bad}$. This is straightforward once we have finite state automata representation of $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$ and $L^{\uparrow}_{bad}$.

Let $L^{\uparrow}_{bad}$ be recognized by a finite state automaton $M = (Q, S\cup\{\#\}, \delta, Q_i, Q_m)$. Then $\mathcal{C}_{\uparrow}$ is encoded by the finite state automaton $M_{\mathcal{C}_{\uparrow}} =$

$$(Q\times(S_i\cup\{\varnothing\}), S\cup\{\#\}, \delta_{\mathcal{C}_{\uparrow}}, Q_i\times\{\varnothing\}, Q_m\times S_i)$$

where $\delta_{\mathcal{C}_{\uparrow}}$ is the relation defined such that

(1) For each $\sigma \in S\cup\{\#\}, q, q' \in Q, ((q,\varnothing),\sigma,(q',\varnothing)) \in \delta_{\mathcal{C}_{\uparrow}}$ iff $(q,\sigma,q')\in\delta$
(2) For each $s' \in S_i, q, q' \in Q, ((q,\varnothing),\#,(q',s')) \in \delta_{\mathcal{C}_{\uparrow}}$ iff $(q,s',q')\in\delta$
(3) For each $s' \in S_i, \sigma \in S\cup\{\#\}, q, q' \in Q, ((q,s'),\sigma,(q',s')) \in \delta_{\mathcal{C}_{\uparrow}}$ iff $(q,\sigma,q')\in\delta$

Indeed, $M_{\mathcal{C}_{\uparrow}}$ is constructed to recognize exactly the domain $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$ of $\mathcal{C}_{\uparrow}$, and it encodes $\mathcal{C}_{\uparrow}$ in the following sense: for any state $w \in (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad}$ that requires entrance control, we have

$$M^{EC}_{\mathcal{C}_{\uparrow}}(w) := \{enters_s \mid s \in P_2(\delta_{\mathcal{C}_{\uparrow}}(Q_i\times\{\varnothing\}, w)\cap(Q_m\times S_i))\} = \mathcal{C}_{\uparrow}(w).$$

Here $\delta_{\mathcal{C}_{\uparrow}}(Q_i\times\{\varnothing\}, w)$[3] is used to denote the set of states that can be reached from initial states in $Q_i\times\{\varnothing\}$ after executing string w and P_2 is used to denote the projection operator that maps each tuple $(q,s)\in Q_m\times S_i$ to its second component $s\in S_i$, where the domain of P_2 has been naturally extended to sets of tuples. The idea of the above finite state automaton encoding is briefly explained as follows: for each $w\in L^{\uparrow}_{bad}$, if $w = w_1 s w_2$ for some $w_1, w_2 \in (S\cup\{\#\})^*$ and $s\in S_i$, then we have $w_1\#w_2 \in (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$. Thus, each string in $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$ can be obtained by replacing, in a non-deterministic fashion, an occurrence of $s\in S_i$ in each string $w\in L^{\uparrow}_{bad}$ with #. The second component $\varnothing$ in $(q,\varnothing)$ is used to mark that no replacement has been made so far up to state q, and the second component $s'\in S_i$ in (q,s') is used to mark that an s' has been replaced up to state q. The transition $((q,\varnothing),\#,(q',s'))$ is used to mark the exact place where the replacement is made and the exact element $s'\in S_i$ that is replaced. The above claim is now summarized in the following proposition.

Proposition 7. *The following results hold.*

(1) $L_m(M_{\mathcal{C}_{\uparrow}}) = (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$.
(2) *For each* $w\in L_m(M_{\mathcal{C}_{\uparrow}})$, *we have*

$$M^{EC}_{\mathcal{C}_{\uparrow}}(w) = \{enters_s \mid R_{enters_s}[\{w\}]\cap L^{\uparrow}_{bad} \neq \varnothing\}$$

Thus, for each state $w \in (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad}$, *we have* $M^{EC}_{\mathcal{C}_{\uparrow}}(w) = \mathcal{C}_{\uparrow}(w)$.

Proof. (1) Let $w\in(S\cup\{\#\})^*$ be any string that is accepted by $M_{\mathcal{C}_{\uparrow}}$, i.e., $w\in L_m(M_{\mathcal{C}_{\uparrow}})$. Then, by definition, there exist some $k\geq 0$, $n\geq 0$, $q_0, q_1, \ldots, q_k, q'_0, q'_1, \ldots, q'_n \in Q$, where $q_0\in Q_i, q'_n\in Q_m$, $\sigma_1,\sigma_2,\ldots,\sigma_k,\sigma'_1,\sigma'_2,\ldots,\sigma'_n \in S\cup\{\#\}$ and some $s'\in S_i$ such that

(a) $\forall i\in\{0,1,\ldots,k-1\}, ((q_i,\varnothing),\sigma_{i+1},(q_{i+1},\varnothing))\in\delta_{\mathcal{C}_{\uparrow}}$
(b) $((q_k,\varnothing),\#,(q'_0,s'))\in\delta_{\mathcal{C}_{\uparrow}}$
(c) $\forall i\in\{0,1,\ldots,n-1\}, ((q'_i,s'),\sigma'_{i+1},(q'_{i+1},s'))\in\delta_{\mathcal{C}_{\uparrow}}$

We have $w = w_1\#w_2$, where $w_1 = \sigma_1\sigma\ldots\sigma_k$ and $w_2 = \sigma'_1\sigma'_2\ldots\sigma'_n$. By the definition of $\delta_{\mathcal{C}_{\uparrow}}$, we have

(a′) $\forall i\in\{0,1,\ldots,k-1\}, (q_i,\sigma_{i+1},q_{i+1})\in\delta$
(b′) $(q_k,s',q'_0)\in\delta$
(c′) $\forall i\in\{0,1,\ldots,n-1\}, (q'_i,\sigma'_{i+1},q'_{i+1})\in\delta$

We then have $w_1 s' w_2 = \sigma_1\sigma\ldots\sigma_k s'\sigma'_1\sigma'_2\ldots\sigma'_n \in L^{\uparrow}_{bad}$ and thus $w = w_1\#w_2 \in (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$. We conclude that $L_m(M_{\mathcal{C}_{\uparrow}}) \subseteq (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$.

Let w be any string of $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$. We have $w = w_1\#w_2$ for some $w_1, w_2 \in (S\cup\{\#\})^*$ and there exists some $s'\in S_i$, such that $w_1 s' w_2 \in L^{\uparrow}_{bad}$. There exist some $k\geq 0, n\geq 0, \sigma_1,\sigma_2,\ldots,\sigma_k,\sigma'_1,\sigma'_2,\ldots,\sigma'_n \in S\cup\{\#\}$, such that $w_1 = \sigma_1\sigma_2\ldots\sigma_k$ and $w_2 = \sigma'_1\sigma'_2\ldots\sigma'_n$. Then, there exist $q_0, q_1, \ldots, q_k, q'_0, q'_1, \ldots, q'_n \in Q$, where $q_0\in Q_i, q'_n\in Q_m$, such that (a′), (b′), (c′) hold. Then, by the definition of $\delta_{\mathcal{C}_{\uparrow}}$, (a), (b), (c) hold. That is, we have $w = w_1\#w_2 = \sigma_1\sigma\ldots\sigma_k\#\sigma'_1\sigma'_2\ldots\sigma'_n \in L_m(M_{\mathcal{C}_{\uparrow}})$. We then conclude that $(\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] \subseteq L_m(M_{\mathcal{C}_{\uparrow}})$.

Thus, we have $L_m(M_{\mathcal{C}_{\uparrow}}) = (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}]$.

(2) In the rest, we show that for each $w\in L_m(M_{\mathcal{C}_{\uparrow}})$, we have

$$M^{EC}_{\mathcal{C}_{\uparrow}}(w) = \{enters_s \mid R_{enters_s}[\{w\}]\cap L^{\uparrow}_{bad} \neq \varnothing\}$$

It then follows that for each $w \in (\bigcup_{s\in S_i} R^{-1}_{enters_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad}$, we have $M^{EC}_{\mathcal{C}_{\uparrow}}(w) = \mathcal{C}_{\uparrow}(w)$.

Indeed, let w be any string of $L_m(M_{\mathcal{C}_{\uparrow}})$. There exist some $k\geq 0, n\geq 0, \sigma_1,\sigma_2,\ldots,\sigma_k,\sigma'_1,\sigma'_2,\ldots,\sigma'_n \in S\cup\{\#\}$ such that $w = \sigma_1\sigma\ldots\sigma_k\#\sigma'_1\sigma'_2\ldots\sigma'_n$.

Let $enters_{s'}$ be any element of $M^{EC}_{\mathcal{C}_{\uparrow}}(w)$, i.e., let s' be any element of $P_2(\delta_{\mathcal{C}_{\uparrow}}(Q_i\times\{\varnothing\}, w)\cap(Q_m\times S_i))$. Then there exist some $q_0, q_1, \ldots, q_k, q'_0, q'_1, \ldots, q'_n \in Q$, where $q_0\in Q_i, q'_n\in Q_m$, such that (a), (b), (c) hold. By the definition of $\delta_{\mathcal{C}_{\uparrow}}$, it follows that (a′), (b′), (c′) hold and thus $\sigma_1\sigma\ldots\sigma_k s'\sigma'_1\sigma'_2\ldots\sigma'_n \in L^{\uparrow}_{bad}$. It then follows that $R_{enters_{s'}}[\{w\}]\cap L^{\uparrow}_{bad}\neq\varnothing$ and thus $enters_{s'} \in \{enters_s \mid R_{enters_s}[\{w\}]\cap L^{\uparrow}_{bad}\neq\varnothing\}$. We conclude that $M^{EC}_{\mathcal{C}_{\uparrow}}(w) \subseteq \{enters_s \mid R_{enters_s}[\{w\}]\cap L^{\uparrow}_{bad}\neq\varnothing\}$.

On the other hand, let $enters_{s'}$ be any element of $\{enters_s \mid R_{enters_s}[\{w\}]\cap L^{\uparrow}_{bad}\neq\varnothing\}$. That is, $R_{enters_{s'}}[\{w\}]\cap L^{\uparrow}_{bad}\neq\varnothing$. Then $\sigma_1\sigma\ldots\sigma_k s'\sigma'_1\sigma'_2\ldots\sigma'_n \in L^{\uparrow}_{bad}$. Then there exist some $q_0, q_1, \ldots, q_k, q'_0, q'_1, \ldots, q'_n \in Q$, where $q_0\in Q_i, q'_n\in Q_m$, such that (a′), (b′), (c′) hold. Then, by the definition of $\delta_{\mathcal{C}_{\uparrow}}$, (a), (b), (c) hold. Thus, we have $s'\in P_2(\delta_{\mathcal{C}_{\uparrow}}(Q_i\times\{\varnothing\}, w)\cap(Q_m\times S_i))$ and then

[3] We here use a function notation of $\delta_{\mathcal{C}_{\uparrow}}$ only for convenience.

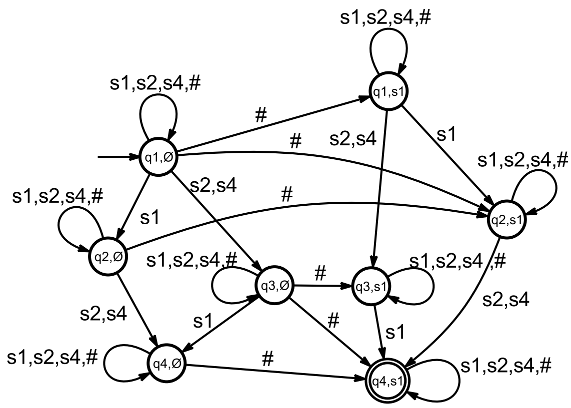


Fig. 4. Representation of maximally permissive deadlock freeness entrance control function for $G^{\infty}_{o,\#}$.

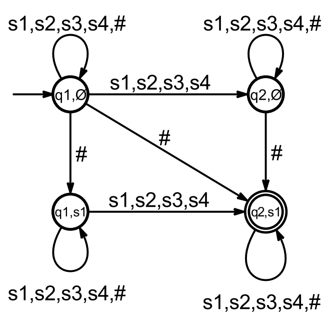


Fig. 5. Representation of maximally permissive blocking freeness entrance control function for $G^{\infty}_{o,\#}$.

$enter_{s'} \in M^{EC}_{C_\uparrow}(w) = \{enter_s \mid s \in P_2(\delta_{C_\uparrow}(Q_i \times \{\varnothing\}, w) \cap (Q_m \times S_i))\}$. We then conclude that $\{enter_s \mid R_{enter_s}[\{w\}] \cap L^{\uparrow}_{bad} \neq \varnothing\} \subseteq M^{EC}_{C_\uparrow}(w)$. ■

Example. The automata representation $M_{C_\uparrow}$ of the maximally permissive entrance control functions for $G^{\infty}_{o,\#}$ are shown in Figs. 4 and 5, respectively, for the deadlock freeness and blocking freeness properties. Consider deadlock freeness entrance control and the state $w = s_2\#s_4 \in (\bigcup_{s\in S_i} R^{-1}_{enter_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad}$. There is a run of the automaton in Fig. 4 on w that reaches the marked state (q_4, s_1), so the entrance event $enter_{s_1}$ has to be disabled at state w. Now, consider blocking freeness entrance control and the state $w = \# \in (\bigcup_{s\in S_i} R^{-1}_{enter_s})[L^{\uparrow}_{bad}] - L^{\uparrow}_{bad}$. There is a run of the automaton in Fig. 5 on w that reaches the marked state (q_2, s_1), so the entrance event $enter_{s_1}$ has to be disabled at state w.

6. Discussions and conclusions

We have studied the globally synchronized templates and obtained a characterization of the expressive power of their symbolic reachability relations. The symbolic reachability analysis could provide guidance for system modeling and design. For example, we can conclude that the mutual exclusion property, which states that no two modules can be in their critical section at the same time, cannot be modeled by any (uncontrolled) globally synchronized template. Although globally synchronized templates lack theoretical expressiveness, they are already sufficiently expressive to model protocols in timed discrete-event systems where the tick of clock is one of the global events. Furthermore, control mechanism can dramatically increase the expressive power of globally synchronized templates, and thus the limitation of expressive power can be alleviated through a proper design of centralized supervisors or more expressive control templates, if distributed control is preferred. The application of the reachability analysis to the entrance control problem is also illustrated in this work. In future work, we will report our recent development on parameterized supervisor synthesis and compare different classes of parameterized systems such as rendezvous and broadcast templates (Esparza, Finkel, & Mayr, 1999; Nazari & Thistle, 2012; Su, 2013) with globally synchronized templates.

Acknowledgments

The authors would like to thank the anonymous reviewers for comments and suggestions that improve the quality of this work. The authors are funded by Natural Sciences and Engineering Research Council (NSERC) Grant Number DG-7399, Romanian grant PN-III-P4-ID-PCE-2016-0210, National Natural Science Foundation of China Grant CNSF-61374058, Australian Research Council Grant DP-130100156 and Singapore Ministry of Education Tier 1 Academic Research Grant M4011221.040 RG84/13.

References

Abdulla, P. A., Jonsson, B., Nilsson, M., & Saksena, M. (2004). A survey of regular model checking. In *CONCUR, LNCS*: *vol. 3170*. (pp. 35–48).

Aminof, B., Kotek, T., Rubin, S., Spegni, F., & Veith, H. (2014). Parameterized model checking of rendezvous systems. In *CONCUR, LNCS*: *vol. 8704*. (pp. 109–124).

Apt, K. R., & Kozen, D. C. (1986). Limits for automatic verification of finite-state concurrent systems. *Information Processing Letters*, *22*(6), 307–309.

Bherer, H., Desharnais, J., & St-Denis, R. (2009). Control of parameterized discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*, *19*(2), 213–265.

Boigelot, B., Legay, A., & Wolper, P. (2003). Iterating transducers in the large. In *CAV, LNCS*: *vol. 2725*. (pp. 223–235).

Bouajjani, A., Habermehl, P., & Vojnara, T. (2004). Abstract regular model checking. In *CAV, LNCS*: *vol. 3114*. (pp. 372–386).

Bouajjani, A., Jonsson, B., Nilsson, M., & Touili, T. (2000). Regular model checking. In *LNCS*: *vol. 1855*. *Computer aided verification* (pp. 403–418).

Bouajjani, A., Legay, A., & Wolper, P. (2005). Handling liveness properties in (ω-)regular model checking. *Electronic Notes in Theoretical Computer Science*, *138*(3), 101–115.

Dams, D., Lakhnech, Y., & Steffen, M. (2001). Iterating transducers. In *CAV, LNCS*: *vol. 2102*. (pp. 286–297).

Engelfriet, J. (2007). A Kleene characterization of computability. In *Information processing letters (Vol. 101)* (pp. 139–140).

Esparza, J. (2014). Keeping a crowd safe: on the complexity of parameterized verification. In *International symposium on theoretical aspects of computer science* (pp. 1–10).

Esparza, J., Finkel, A., & Mayr, R. (1999). On the verification of broadcast protocols. In *14th IEEE symposium on logic in computer science* (pp. 352-359).

Gall, T. Le, Jeannet, B., & Marchand, H. (2005). Supervisory control of infinite symbolic systems using abstract interpretation, In *44th IEEE conference on decision and control and European control conference* (pp. 30–35).

Habermehl, P., & Vojnar, T. (2005). Regular model checking using inference of regular languages. In *Proceedings of the 6th international workshop on verification of infinite-state systems, Vol. 138*, no. 3. (pp. 21–36).

Harrison, M. A. (1978). *Introduction to formal language theory*. Addison-Wesley Publishing Company.

Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to automata theory, languages, and computation*. Massachusetts: Addison-Wesley, Reading.

Jonsson, B., & Nilsson, M. (2000). Transitive closures of regular relations for verifying infinite state systems. In *TACAS, LNCS*: *vol. 1785*. (pp. 220–235).

Kozen, D. (1991). A completeness theorem for Kleene algebras and the algebra of regular events, In *6th IEEE Symposium on Logic in Computer Science* (pp. 214-225).

Kumar, R., & Garg, V. K. (1993). Predicates and predicate transformers for supervisory control of discrete event dynamical systems. *IEEE Transactions on Automatic Control*, *38*(2), 232–247.

Kumar, R., & Garg, V. K. (2005). On computation of state avoidance control for infinite state systems in assignment program framework. *IEEE Transactions on Automation Science and Engineering*, *2*(1), 87–91.

Lin, L., Stefanescu, A., & Su, R. (2016). On distributed and parameterized supervisor synthesis problems. *IEEE Transactions on Automatic Control*, *61*(3), 777–782.

Lin, L., Stefanescu, A., Su, R., Wang, W., & Shehabinia, A. R. (2014). Towards decentralized synthesis: decomposable sublanguage and joint observability problems. In *American control conference* (pp. 2047-2052).

Nazari, K., & Thistle, J. G. (2012). Blocking in fully connected networks of arbitrary size. *IEEE Transactions on Automatic Control*, *57*(5), 1233–1242.

Rohloff, K., & Lafortune, S. (2006). The verification and control of interacting similar discrete event systems. *SIAM Journal on Control and Optimization*, *45*(2), 634–667.

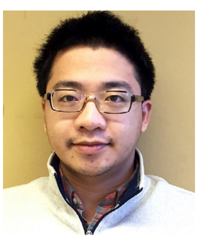
Su, R. (2013). Discrete-event modeling of multi-agent systems with broadcasting-based parallel composition. *Automatica*, *49*(11), 3502–3506.

Terlutte, A., & Simplot, D. (2000). Iteration of rational transductions. *Theoretical Informatics and Applications*, *34*(2), 99–129.

Touili, T. (2001). Regular model checking using widening techniques. *Electronic Notes in Theoretical Computer Science*, *50*(4), 342–356.

Wang, W., Su, R., & Lin, L. (2013). On analysis of deadlock and blocking freeness in isomorphic module systems. In *American control conference* (pp. 923-928).

Zibaeenejad, M. H., & Thistle, J. G. (2014). Weak invariant simulation and analysis of parameterized networks. *IEEE Transactions on Automatic Control*, *59*(8), 2024–2037.

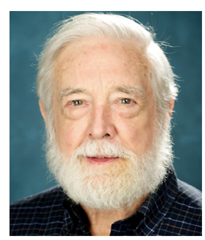


Liyong Lin received the B.E. degree and Ph.D. degree in electrical engineering in 2011 and 2016, respectively, both from Nanyang Technological University, where he has also worked as a project officer. Since June 2016, he has been a postdoctoral fellow in the University of Toronto. His main research interests include supervisory control theory and formal methods. He previously was an undergraduate intern in Data Storage Institute, where he worked on single and dual-stage servomechanism of hard disk drives.

Alin Stefanescu received the M.Sc. degree from the University of Bucharest, Romania, in 2000, and the Ph.D. degree in computer science from the University of Stuttgart, Germany, in 2006. He is currently a Professor at the University of Bucharest, Romania. His research career path has included both academic and industry. In academia, he investigated software validation, verification, and synthesis at European Universities in Edinburgh, U.K.; Munich, Germany; Bucharest, Romania; and Konstanz, Germany. In industry, at SAP Research, Darmstadt, Germany, he made contributions to model-based testing for service-oriented architectures and transferring the results into the SAP development groups.

Weilin Wang received the M.S. degree in electrical engineering: systems, the M.S.E. degree in industrial engineering, and the Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, in 2003, 2006, and 2007, respectively. He is currently a Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning that is with School of Optical–Electrical& Computer Engineering at University of Shanghai for Science and Technology. He is also an academic in Faculty of Information Technology at Monash University. His research interests include model checking, synthesis of sensor activation and communication strategies for control and diagnosis of discrete event systems, traffic signal coordination, coverage and cross layer optimization in wireless sensor networks, and cooperative control for multi-robot systems. Dr. Wang is in the Shanghai Thousand Talents Plan Program.

Rong Su received the B.E. degree in Automatic Control from the University of Science and Technology of China in 1997, and the M.A.Sc. and Ph.D. degrees both in electrical engineering from the University of Toronto in 2000 and 2004, respectively. Since then he was affiliated with University of Waterloo and Technical University of Eindhoven before he joined Nanyang Technological University in 2010. His research interests include discrete event systems, supervisory control, model-based fault diagnosis, multi-agent systems, optimization and scheduling with applications in green buildings, flexible manufacturing, power management and intelligent transportation systems. In the aforementioned areas, he has over 130 publications in journals, book chapters, and conference proceedings, and two patents. Dr. Su is an Associate Editor for Journal of Discrete Event Dynamic Systems: Theory and Applications, Journal of Control and Decision, and Transactions of the Institute of Measurement and Control, and the Chair of IEEE Control Systems Society Technical Committee on Smart Cities.



W. Murray Wonham received the B.Eng. degree in engineering physics from McGill University in 1956, and the Ph.D. in control engineering from the University of Cambridge (U.K.) in 1961. From 1961 to 1969 he was associated with several U.S. research groups in control. Since 1970 he has been a faculty member in Systems Control, with the Department of Electrical and Computer Engineering of the University of Toronto. His research interests have included stochastic control and filtering, geometric multivariable control, and discrete-event systems. He is the author of "Linear Multivariable Control: A Geometric Approach" (Springer-Verlag: 3rd ed. 1985), co-author (with C. Ma) of "Nonblocking Supervisory Control of State Tree Structures (Springer-Verlag: 2005), and co-author (with K. Cai) of "Supervisor Localization: A Top-Down Approach to Distributed Control of Discrete-Event Systems" (Springer-Verlag: 2015).

Wonham is a Fellow of the Royal Society of Canada, a Life Fellow of the IEEE, a Foreign Member of the (U.S.) National Academy of Engineering, and an Honorary Professor of the Beijing University of Aeronautics and Astronautics. In 1987 he received the IEEE Control Systems Science and Engineering Award and in 1990 was Brouwer Medalist of the Netherlands Mathematical Society. In 1996 he was appointed University Professor in the University of Toronto, and in 2000 University Professor Emeritus.