Model abstraction of nondeterministic finite-state automata in supervisor synthesis

Citation for published version (APA):
Su, R., Schuppen, van, J. H., & Rooda, J. E. (2010). Model abstraction of nondeterministic finite-state automata in supervisor synthesis. *IEEE Transactions on Automatic Control*, *55*(11), 2527-2541. https://doi.org/10.1109/TAC.2010.2046931

DOI:
10.1109/TAC.2010.2046931

Document status and date:
Published: 01/01/2010

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Model Abstraction of Nondeterministic Finite-State Automata in Supervisor Synthesis

Rong Su, Jan H. van Schuppen, *Member, IEEE*, and Jacobus E. Rooda, *Member, IEEE*

***Abstract*—Blockingness is one of the major obstacles that need to be overcome in the Ramadge-Wonham supervisory synthesis paradigm, especially for large systems. In this paper, we propose an abstraction technique to overcome this difficulty. We first provide details of this abstraction technique, then describe how it can be applied to a supervisor synthesis problem, where plant models are nondeterministic but specifications and supervisors are deterministic. We show that a nonblocking supervisor for an abstraction of a plant under a specification is guaranteed to be a nonblocking supervisor of the original plant under the same specification. The reverse statement is also true, if we impose an additional constraint in the choice of the alphabet of abstraction, i.e., every event, which is either observable or labels a transition to a marker state, is contained in the alphabet of abstraction.**

***Index Terms*—Automaton abstraction, discrete-event systems, nondeterministic finite-state automata, supervisor synthesis.**

I. Introduction

THE automaton-based Ramadge-Wonham (RW) supervisory control paradigm first appeared in the control literature in 1982, which was subsequently summarized in the well known journal papers [18], [26]. Since then there has been a large volume of literature under the same paradigm. In the RW paradigm one of the main problems is to synthesize a supervisor for a plant such that the closed-loop behavior is *nonblocking*, *controllable* [18], *observable* or *normal* [11], and satisfies some prescribed requirements. The main difficulty of supervisor synthesis is to achieve nonblockingness because the total number of states of a plant model increases quickly when the number of local components increases, due to the synchronous product which incurs Cartesian product over automata. To overcome this difficulty, some authors attempt to introduce sufficient conditions which allow local supervisor synthesis. For example, in [27] the authors propose the concept of *modularity*, which is then extended to the concept of *local modularity* in [17]. When local supervisors are (locally) modular, a globally nonblocking supervisory control is achieved. Nevertheless, testing (local) modularity itself usually imposes prohibitive computational complexity. Another notable work is presented in [10], where, by imposing *interface consistency* and *level-wise controllability* among subsystems and local supervisors in a hierarchical setup, a very large nonblocking control problem may be solved, e.g. the size of the state set reaches 10^{21} in the Atelier Interétablissement de Productique (AIP) example [10]. But the approach does not tell how to deliberately and systematically design interfaces that allow synthesis of local supervisors that satisfy those properties. Instead, it assumes that those interfaces are given before synthesis, as mentioned in [9]. In [12] the authors present an interesting approach, which is aimed at synthesizing a state-feedback supervisor. The authors represent product states as *state tree structures*, upon which the power of symbolic computation (as manifested by the manipulation of binary decision diagrams) is fully utilized. It has been shown in [12] that a system with 10^{24} states can be accommodated. Nevertheless, this approach is essentially a centralized approach, and it does not deal with cases when only partial observations of states are available for control. In this paper we will discuss the usage of abstraction to reduce complexity in synthesizing nonblocking supervisors, where partial observation may be present.

Our first contribution is to present a novel automaton-based abstraction technique. The idea of abstraction has been known in the literature, e.g. in [2] abstraction is used in the modular and hierarchical supervisor synthesis; it is also used in [16] for testing the nonblocking property, and in [19] for decentralized control. Nevertheless, their approaches are language-based, and rely on natural projections that satisfy the *observer* property [23]. Although a natural projection can always be modified to become an observer (with respect to a specific language) [24], such a modification has a potential drawback in the sense that the alphabet of the codomain of the projection may be fairly large for the sake of achieving the observer property, and the consequence is that the size of the projected image may not be small enough to allow supervisor synthesis for large systems. Our abstraction technique is automaton-based, which computes an abstraction for any pre-specified abstraction alphabet, and guarantees that the abstraction is suitable for supervisor synthesis. Thus, the drawback of the language-based abstraction techniques is avoided in our approach. Several strategies for automaton abstraction have been proposed, e.g., in [4], [5], [7], [13], [22]. Among them, [22] aims to achieve weak bisimilarity between an automaton and its abstraction. In [4], [5], [7], [13] the authors first use special events, which are called *silent* events and usually denoted by τ, or τ_c and τ_u when distinguishing controllable and uncontrollable events is necessary, to replace internal events that are not in the abstraction alphabet.

Manuscript received May 19, 2008; revised October 24, 2008; accepted March 22, 2010. First published April 05, 2010; current version published November 03, 2010. Recommended by Associate Editor E. Fabre.

R. Su is with the School of Electrical and Electronic Enginnering, Nanyang Technological University, Singapore, 639798 (email: rsu@ntu.edu.sg).

J. H. van Schuppen is with Centrum voor Wiskunde en Informatica (CWI), Amsterdam 1090 GB, The Netherlands (e-mail: j.h.van.schuppen@cwi.nl).

J. E. Rooda is with the Systems Engineering Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven 5600 MB, The Netherlands (e-mail: j.e.rooda@tue.nl).

Digital Object Identifier 10.1109/TAC.2010.2046931

Then they apply heuristic rewriting rules to ensure that appropriate equivalence relations hold between automata before and after rewriting, e.g., conflict equivalence in [4], [7], supervision equivalence in [5] and synthesis equivalence in [13]. The primary goal of our approach is to create an abstraction for an automaton G, which is not necessarily weak bisimilar to G, such that any automaton S, whose alphabet is the same as that of the abstraction and is nonconflicting with the abstraction, must be nonconflicting with G. If we impose an additional constraint in the choice of the alphabet of abstraction, then it is also true that S is nonconflicting with G implies that S is nonconflicting with the abstraction—at this point, our approach is close to achieving conflict equivalence, but it does not require silent events and heuristic rewriting rules.

Our second contribution is to show how the proposed abstraction technique can be applied to a synthesis problem, where the plant model is nondeterministic but the specification and the supervisor are deterministic. There exists a large body of publications on supervisor synthesis for nondeterministic systems. For example, in [1] both plant and supervisor models are nondeterministic and different types of deterministic or nondeterministic specifications are considered. In [6], [8] the plant is considered to be nondeterministic and both the specification and the supervisor are deterministic. In [15] the plant and the specification are nondeterministic but the supervisor is deterministic. In [28], [29] the plant and the specification are nondeterministic and the supervisor can also be nondeterministic. The main difference between these papers and ours is that, we focus on how to use automaton abstraction in synthesis to reduce computational complexity. We consider a nondeterministic plant because an abstraction of a deterministic plant is usually nondeterministic. We consider a deterministic specification and a deterministic supervisor because they are typical in industrial systems, and they allow automaton abstraction to be used in synthesis. We are still investigating whether the proposed abstraction technique is also applicable to cases with nondeterministic requirements and supervisors. Although [5], [7], [13], [22] also utilize abstraction in synthesis, their abstraction techniques are different from ours. Because the main objective of this paper is to establish a connection between the existence of a nonblocking supervisor for a plant model and the existence of a nonblocking supervisor for an abstract model created by our abstraction technique, details of how to synthesize a nonblocking supervisor based on nondeterministic finite-state automata are not mentioned in this paper, but addressed in [21]. We also introduce the concept of *state normality*, which allows for the computation of a supremal nonblocking state-normal supervisor for a nondeterministic system.

This paper is organized as follows. In Section II we introduce an abstraction technique over nondeterministic automata. In Section III we show the usage of the proposed abstraction technique in supervisor synthesis. After an illustrative example in Section IV, conclusions are stated in Section V. Long proofs are presented in the Appendix.

II. Automaton Abstraction and Relevant Properties

In this section we follow the notations used in [25]. We first briefly review concepts related to languages and automata, then introduce the concept of automaton abstraction. After that, we present properties of abstraction which are used in supervisor synthesis.

A. Concepts of Languages, Automata and Abstraction

Let Σ be a finite alphabet, and Σ^* denote the Kleene closure of Σ, i.e., the collection of all finite sequences of events taken from Σ. Given two strings $s, t \in \Sigma^*$, s is called a *prefix substring* of t, written as $s \leq t$, if there exists $s' \in \Sigma^*$ such that $ss' = t$, where ss' denotes the concatenation of s and s'. We use ϵ to denote the empty string of Σ^* such that for any string $s \in \Sigma^*$, $\epsilon s = s\epsilon = s$. A subset $L \subseteq \Sigma^*$ is called a *language*. $\overline{L} = \{s \in \Sigma^* | (\exists t \in L) s \leq t\} \subseteq \Sigma^*$ is called the *prefix closure* of L. L is called *prefix closed* if $L = \overline{L}$. Given two languages $L, L' \subseteq \Sigma^*$, let $LL' := \{ss' \in \Sigma^* | s \in L \wedge s' \in L'\}$ be the concatenation of L and L', which contains every string obtainable by concatenating one string from L and one string from L'.

Let $\Sigma' \subseteq \Sigma$. A mapping $P : \Sigma^* \to \Sigma'^*$ is called the *natural projection* with respect to (Σ, Σ'), if

1) $P(\epsilon) = \epsilon$;
2) $(\forall \sigma \in \Sigma) P(\sigma) : \begin{cases} \sigma & \text{if } \sigma \in \Sigma' \\ \epsilon & \text{otherwise} \end{cases}$;
3) $(\forall s\sigma \in \Sigma^*) P(s\sigma) = P(s)P(\sigma)$.

Given a language $L \subseteq \Sigma^*$, $P(L) := \{P(s) \in \Sigma'^* | s \in L\}$. The inverse image mapping of P is

$$P^{-1} : 2^{\Sigma'^*} \to 2^{\Sigma^*} : L \mapsto P^{-1}(L) := \{s \in \Sigma^* | P(s) \in L\}.$$

Given $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$, the *synchronous product* of L_1 and L_2 is defined as $L_1 || L_2 := P_1^{-1}(L_1) \cap P_2^{-1}(L_2)$, where $P_1 : (\Sigma_1 \cup \Sigma_2)^* \to \Sigma_1^*$ and $P_2 : (\Sigma_1 \cup \Sigma_2)^* \to \Sigma_2^*$ are natural projections. Clearly, $||$ is commutative and associative. Next, we introduce automaton product and abstraction.

A *nondeterministic finite-state automaton* is a 5-tuple $G = (X, \Sigma, \xi, x_0, X_m)$, where X stands for the state set, Σ for the alphabet, $\xi : X \times \Sigma \to 2^X$ for the nondeterministic transition function, x_0 for the initial state and $X_m \subseteq X$ for the marker state set. As usual, the domain of ξ is extended to $X \times \Sigma^*$. If for all $x \in X$ and $\sigma \in \Sigma$, $\xi(x, \sigma)$ contains no more than one element, then G is called *deterministic*. Let

$$B(G) := \{s \in \Sigma^* | (\exists x \in \xi(x_0, s)) (\forall s' \in \Sigma^*) \xi(x, s') \cap X_m = \varnothing\}.$$

Any string $s \in B(G)$ can lead to a state x, from which no marker state is reachable, i.e. for any $s' \in \Sigma^*$, $\xi(x, s') \cap X_m = \varnothing$. Such a state x is called a *blocking state* of G, and we call $B(G)$ the *blocking set* of G. A state that is not a blocking state is called a *nonblocking state*. We say G is *nonblocking* if $B(G) = \varnothing$. For each $x \in X$, we define another set $N_G(x) := \{s \in \Sigma^* | \xi(x, s) \cap X_m \neq \varnothing\}$, and call $N_G(x_0)$ the *nonblocking set* of G, which is simply the set of all strings recognized by G. For the notation simplicity, we use $N(G)$ to denote $N_G(x_0)$. It is possible that $B(G) \cap \overline{N(G)} \neq \varnothing$, due to nondeterminism. Let $L(G) := \{s \in \Sigma^* | \xi(x_0, s) \neq \varnothing\}$ be the *closed behavior* of G.

Given two nondeterministic automata $G_i = (X_i, \Sigma_i, \xi_i, x_{i,0}, X_{i,m})$ $(i = 1, 2)$, the *product* of G_1 and G_2, written as $G_1 \times G_2$, is an automaton such that

$$G_1 \times G_2 = (X_1 \times X_2, \Sigma_1 \cup \Sigma_2, \xi_1 \times \xi_2, (x_{1,0}, x_{2,0}), X_{1,m} \times X_{2,m})$$

where $\xi_1 \times \xi_2 : X_1 \times X_2 \times (\Sigma_1 \cup \Sigma_2) \to 2^{X_1 \times X_2}$ is defined as follows:

$$(\xi_1 \times \xi_2)\left((x_1, x_2), \sigma\right) := \begin{cases} \xi_1(x_1, \sigma) \times \{x_2\} & \text{if } \sigma \in \Sigma_1 - \Sigma_2 \\ \{x_1\} \times \xi_2(x_2, \sigma) & \text{if } \sigma \in \Sigma_2 - \Sigma_1 \\ \xi_1(x_1, \sigma) \times \xi_2(x_2, \sigma) & \text{if } \sigma \in \Sigma_1 \cap \Sigma_2. \end{cases}$$

Clearly, × is commutative and associative. $\xi_1 \times \xi_2$ is extended to $X_1 \times X_2 \times (\Sigma_1 \cup \Sigma_2)^* \to 2^{X_1 \times X_2}$. By a slight abuse of notations, from now on we use $G_1 \times G_2$ to denote its *reachable* part, which contains all states reachable from $(x_{1,0}, x_{2,0})$ by $\xi_1 \times \xi_2$ and relevant transitions between each pair of these states. Next, we introduce automaton abstraction, which requires the following concept of marking weak bisimilarity.

Definition 1: Given $G = (X, \Sigma, \xi, x_0, X_m)$, let $\Sigma' \subseteq \Sigma$ and $P : \Sigma^* \to \Sigma'^*$ be the natural projection. A *marking weak bisimulation* relation on X with respect to Σ' is an equivalence relation $R \subseteq \{(x, x') \in X \times X | x \in X_m \iff x' \in X_m\}$ such that, for all $(x, x') \in R$, $s \in \Sigma^*$ and $y \in \xi(x, s)$

$$(\exists s' \in \Sigma^*) P(s) = P(s') \wedge (\exists y' \in \xi(x', s'))\, (y, y') \in R.$$

The largest marking weak bisimulation relation on X with respect to Σ' is called *marking weak bisimilarity* on X with respect to Σ', written as $\approx_{\Sigma', G}$. □

Marking weak bisimilarity is almost the same as weak bisimilarity described in [14], except for the special treatment on marker states. We now introduce abstraction.

Definition 2: Given $G = (X, \Sigma, \xi, x_0, X_m)$, let $\Sigma' \subseteq \Sigma$. The *automaton abstraction* of G with respect to $\approx_{\Sigma', G}$ is an automaton $G/ \approx_{\Sigma', G} := (Z, \Sigma', \delta, z_0, Z_m)$ where

1) $Z := X/ \approx_{\Sigma', G} := \{\langle x \rangle := \{x' \in X | (x, x') \in \approx_{\Sigma', G}\} | x \in X\}$;
2) $z_0 := \langle x_0 \rangle$;
3) $Z_m := \{z \in Z | z \cap X_m \neq \varnothing\}$;
4) $\delta : Z \times \Sigma' \to 2^Z$, where for any $(z, \sigma) \in Z \times \Sigma'$, $\delta(z, \sigma) := \{z' \in Z | (\exists x \in z)(\exists u, u' \in (\Sigma - \Sigma')^*) \xi(x, u\sigma u') \cap z' \neq \varnothing\}$.

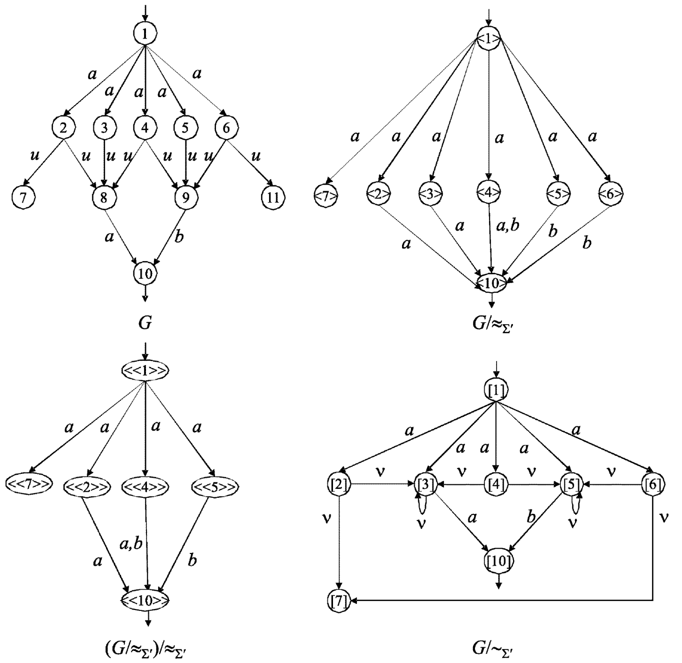
□

The time complexity of computing $G/ \approx_{\Sigma', G}$ mainly results from computing $X/ \approx_{\Sigma', G}$, which can be estimated as follows. We first define a new automaton $G'' = (X, \Sigma' \cup \{\nu\}, \xi'', x_0, X_m)$, where ν is called the *silent* event, which denotes all events in $\Sigma - \Sigma'$, and for all $x, x' \in X$, if there exist $u\sigma u' \in \Sigma^*$ with $u, u' \in (\Sigma - \Sigma')^*$ and $\sigma \in \Sigma'$ such that $x' \in \xi(x, u\sigma u')$, then $x' \in \xi''(x, \sigma)$; if there exists $u \in (\Sigma - \Sigma')^*$ such that $x' \in \xi(x, u)$, then $x' \in \xi''(x, \nu)$. We can show that $X/ \approx_{\Sigma' \cup \{\nu\}, G''}$ is equal to $X/ \approx_{\Sigma', G}$. The total number of transitions in G'' is no more than mn^2, where $n = |X|$ and $m = |\Sigma' \cup \{\nu\}|$. Based on a result in [3], the time complexity of computing $X/ \approx_{\Sigma' \cup \{\nu\} G''}$ is $O(mn^2 \log n)$ if we ignore the complexity caused by checking the condition "$x \in X_m \iff x' \in X_m$" in Def. 1. If we consider this extra condition, which requires comparing at most $(1/2)n(n-1)$ pairs of states in the worst case, then the overall complexity is $O((1/2)n(n-1) + mn^2 \log n) = O(mn^2 \log n)$. From now on, when G is clear from the context, we simply use $\approx_{\Sigma'}$ to denote $\approx_{\Sigma', G}$, and use $\langle x \rangle_{\Sigma'}$ for an element of $X/ \approx_{\Sigma', G}$. If Σ' is also clear from the context, then we simply use $\langle x \rangle$ for $\langle x \rangle_{\Sigma'}$. In other comparable automaton-based abstraction techniques, e.g., [4], [7], [13], [22], the weak bisimilarity is also used, except that in their definition two equivalent states need not have the same marking status, which may potentially make the size of the quotient state set under their construction slightly smaller than the size of $X/ \approx_{\Sigma'}$. On the other hand, in those techniques the definition of δ utilizes the following standard quotient construction:

$$\delta(z, \sigma) := \begin{cases} \{z' \in Z | (\exists x \in z)(\exists x' \in z') \\ \quad x' \in \xi(x, \sigma)\} & \text{if } \sigma \neq \nu \\ \{z' \in Z | (\exists x \in z)(\exists x' \in z') \\ \quad (\exists \sigma' \in \Sigma - \Sigma') x' \in \xi(x, \sigma')\} & \text{if } \sigma = \nu. \end{cases}$$

Our definition of δ is nonstandard in the sense that, two quotient states are only connected by events in Σ', and ν is not used. As a result of this nonstandard definition, two different quotient states in $G/ \approx_{\Sigma'}$ may become equivalent in $(G/ \approx_{\Sigma'})/ \approx_{\Sigma'}$, which usually makes $(G/ \approx_{\Sigma'})/ \approx_{\Sigma'}$ smaller than $G/ \approx_{\Sigma'}$. In the next section we will see that, $G/ \approx_{\Sigma'}$ can be replaced by $(G/ \approx_{\Sigma'})/ \approx_{\Sigma'}$ in supervisor synthesis. There exists a procedure that computes $(G/ \approx_{\Sigma'})/ \approx_{\Sigma'}$ directly from G without applying the abstraction procedure twice, and the complexity of computing $(X/ \approx_{\Sigma'})/ \approx_{\Sigma'}$ is equal to $O(m'n^2 \log n)$, where $m' = |\Sigma'| = m - 1$. Owing to the limited space, we will not discuss this procedure in this paper. As a comparison, we use $G/ \sim_{\Sigma'}$ to denote the standard quotient construction under the weak bisimilarity. Then $(G/ \sim_{\Sigma'})/ \sim_{\Sigma'}$ is the same as $G/ \sim_{\Sigma'}$ (under automaton isomorphism), whose size is close to that of $G/ \approx_{\Sigma'}$. Thus, in practice our technique can obtain smaller abstractions than the standard quotient construction can achieve, which is illustrated in the following example.

Let $G = (X, \Sigma, \xi, x_0, X_m)$ be a nondeterministic automaton depicted in Fig. 1, where $\Sigma = \{a, b, u\}$. Assume $\Sigma' = \{a\ \ b\}$. Then we have the quotient state set $X/ \approx_{\Sigma'} = \{\langle 1 \rangle = \{1\}, \langle 2 \rangle = \{2\}, \langle 3 \rangle = \{3, 8\}, \langle 4 \rangle = \{4\}, \langle 5 \rangle = \{5, 9\}, \langle 6 \rangle = \{6\}, \langle 7 \rangle = \{7, 11\}, \langle 10 \rangle = \{10\}\}$. The abstraction $G/ \approx_{\Sigma'}$ is depicted in Fig. 1. We can check that, in $G/ \approx_{\Sigma'}$ states $\langle 2 \rangle$ and $\langle 3 \rangle$ are equivalent under $\approx_{\Sigma'}$, and so are $\langle 5 \rangle$ and $\langle 6 \rangle$. This happens because the transition map in our definition of abstraction is nonstandard, making the path from state 2 to the blocking state 7 (and from state 6 to state 11) disappears in $G/ \approx_{\Sigma'}$. The abstraction $(G/ \approx_{\Sigma'})/ \approx_{\Sigma'}$ is depicted in Fig. 1, where $(X/ \approx_{\Sigma'})/ \approx_{\Sigma'} = \{\langle\langle 1 \rangle\rangle = \{\langle 1 \rangle\}, \langle\langle 2 \rangle\rangle = \{\langle 2 \rangle, \langle 3 \rangle\}, \langle\langle 4 \rangle\rangle = \{\langle 4 \rangle\}, \langle\langle 5 \rangle\rangle = \{\langle 5 \rangle, \langle 6 \rangle\}, \langle\langle 7 \rangle\rangle = \{\langle 7 \rangle\}, \langle\langle 10 \rangle\rangle = \{\langle 10 \rangle\}\}$. As a comparison, we apply the standard quotient construction on G. To distinguish elements of $X/ \sim_{\Sigma'}$ from those of $X/ \approx_{\Sigma'}$, we use $[x]$ for a quotient state under $\sim_{\Sigma'}$. We have $X/ \sim_{\Sigma'} = \{[1] = \{1\}, [2] = \{2\}, [3] = \{3, 8\}, [4] = \{4\}, [5] = \{5, 9\}, [6] = \{6\}, [7] = \{7, 11\}, [10] = \{10\}\}$. We can see that $X/ \approx_{\Sigma'} = X/ \sim_{\Sigma'}$ because both quotient sets are constructed based on the weak bisimilarity. The quotient automaton $G/ \sim_{\Sigma'}$ is depicted in Fig. 1, which is different from $G/ \approx_{\Sigma'}$ and has more states and transitions than $(G/ \approx_{\Sigma'})/ \approx_{\Sigma'}$ has. In this example, we can see that our abstraction technique does enjoy some computational advantage over other automaton-based



G

$G/\approx_{\Sigma'}$

$(G/\approx_{\Sigma'})/\approx_{\Sigma'}$

$G/\sim_{\Sigma'}$

Fig. 1. Example 1: $G, G/\approx_{\Sigma'}, (G/\approx_{\Sigma'})/\approx_{\Sigma'}$ and $G/\sim_{\Sigma'}$.

abstraction techniques, which utilize the standard quotient construction. Next, we present properties of automaton abstraction.

B. Properties of Automaton Abstraction

We first introduce two more concepts, which are important for applying the aforementioned automaton abstraction in supervisor synthesis.

Definition 3: An automaton $G = (X, \Sigma, \xi, x_0, X_m)$ is *marking aware* with respect to $\Sigma' \subseteq \Sigma$, if

$$(\forall x \in X - X_m)(\forall s \in \Sigma^*)\xi(x,s) \cap X_m \neq \varnothing \Rightarrow P(s) \neq \epsilon$$

where $P : \Sigma^* \to \Sigma'^*$ is the natural projection. □

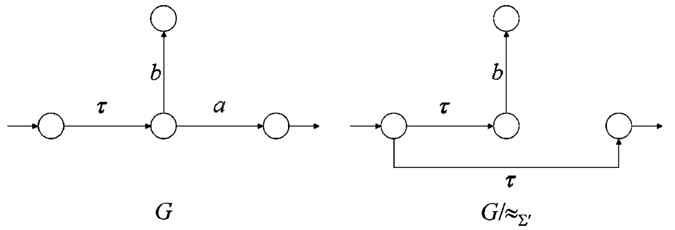
If G is marking aware with respect to Σ', then any string s reaching a marker state from a non-marker state must contain at least one event in Σ'. A sufficient and necessary condition to make G marking aware with respect to Σ' is to put in Σ' every event that labels a transition from a non-marker state to a marker state, namely $\{\sigma \in \Sigma | (\exists x \in X - X_m)(\exists x' \in X_m) x' \in \xi(x,\sigma)\} \subseteq \Sigma'$.

Definition 4: Given an alphabet Σ, we bring in a new event symbol $\tau \notin \Sigma$, and call $G^\tau = (X^\tau, \Sigma \cup \{\tau\}, \xi^\tau, x_0^\tau, X_m^\tau)$ *standardized* if

1) $x_0^\tau \notin X_m^\tau \wedge (\forall x \in X^\tau)[\xi^\tau(x,\tau) \neq \varnothing \iff x = x_0^\tau]$;
2) $(\forall \sigma \in \Sigma)\xi^\tau(x_0^\tau, \sigma) = \varnothing$;
3) $(\forall x \in X^\tau)(\forall \sigma \in \Sigma \cup \{\tau\}) x_0^\tau \notin \xi^\tau(x,\sigma)$.

□

A standardized automaton is nothing but an automaton, in which x_0^τ is not marked and has only outgoing τ transitions with no incoming transitions, and no state except x_0^τ has outgoing τ transition. For an ordinary automaton $G = (X, \Sigma, \xi, x_0, X_m)$ we can *standardize* it (i.e., convert it into a standardized automaton) by simply (1) extending the alphabet to $\Sigma \cup \{\tau\}$, (2) adding a new state x_0^τ, and (3) defining a new transition map ξ^τ such that $\xi^\tau(x_0^\tau, \tau) = \{x_0\}$ and for any $(x, \sigma) \in X \times \Sigma$ we have $\xi^\tau(x,\sigma) = \xi(x,\sigma)$. The resulting automaton $G^\tau = (X \cup \{x_0^\tau\}, \Sigma \cup \{\tau\}, \xi^\tau, x_0^\tau, X_m)$ is a standardized automaton. From now on, unless specified explicitly, we assume that every alphabet contains τ. Thus, if we say Σ_1 and Σ_2 are two alphabets, then $\tau \in \Sigma_1 \cap \Sigma_2$; and if we say $\Sigma' \subseteq \Sigma$ is an alphabet, then $\tau \in \Sigma'$. Let $\phi(\Sigma)$ be the collection of all standardized finite-state automata, whose alphabet is Σ. By a slight abuse of notation, we use G to denote a standardized automaton G^τ. We can easily see that the product of two standardized automata is still a standardized automaton, and abstraction of a standardized automaton is also standardized as long as τ is in the abstraction alphabet. The concepts of marking awareness and standardized automata are used in the following result, which is extensively used in this paper.



G

$G/\approx_{\Sigma'}$

Fig. 2. Example 2: G and $G/\approx_{\Sigma'}$.

Proposition 1: Given alphabets Σ and Σ' with $\Sigma' \subseteq \Sigma$, let $G \in \phi(\Sigma)$ and $P : \Sigma^* \to \Sigma'^*$ be the natural projection. Then

1) $P(B(G)) \subseteq B(G/\approx_{\Sigma'})$ and $P(N(G)) = N(G/\approx_{\Sigma'})$.
2) If G is marking aware with respect to Σ', then $P(B(G)) = B(G/\approx_{\Sigma'})$.

□

The proof is given in the Appendix, which indicates that, if G is not standardized, then we may not always have $P(B(G)) \subseteq B(G/\approx_{\Sigma'})$ and $P(N(G)) = N(G/\approx_{\Sigma'})$, which are critically important in abstraction-based synthesis.

As an illustration of Prop. 1, Fig. 2 depicts an example, where $\Sigma = \{\tau, a, b\}$ and $\Sigma' = \{\tau, b\}$. We can check that $P(N(G)) = \{\tau\} = N(G/\approx_{\Sigma'})$. But $P(B(G)) = \{\tau b\}$ and $B(G/\approx_{\Sigma'}) = \{\tau, \tau b\}$, namely $P(B(G)) \subset B(G/\approx_{\Sigma'})$. In this example, to make G marking aware with respect to Σ', a must be included in Σ'. If we set $\Sigma' = \{\tau, a\}$ then $P(B(G)) = B(G/\approx_{\Sigma'})$, as predicted in Prop. 1.

To show the usefulness of automaton abstraction in supervisor synthesis, we need the following concept.

Definition 5: Given automata $G_i = (X_i, \Sigma_i, \xi_i, x_{i,0}, X_{i.m})$ $(i = 1, 2)$, we say G_1 is *nonblocking preserving* with respect to G_2, denoted as $G_1 \sqsubseteq G_2$, if $B(G_1) \subseteq B(G_2)$, $N(G_1) = N(G_2)$ and for all $s \in \overline{N(G_1)}$ and all $x_1 \in \xi_1(x_{1,0}, s)$, there exists $x_2 \in \xi_2(x_{2,0}, s)$ such that

$$N_{G_2}(x_2) \subseteq N_{G_1}(x_1) \wedge [x_1 \in X_{1,m} \iff x_2 \in X_{2,m}].$$

We say G_1 is *nonblocking equivalent* to G_2, denoted as $G_1 \cong G_2$, if $G_1 \sqsubseteq G_2$ and $G_2 \sqsubseteq G_1$. □

By Def. 5, if G_1 is nonblocking preserving w.r.t. G_2 then their nonblocking behaviors are equal, but G_2's blocking behavior may be larger. The last condition is used to guarantee that nonblocking preserving is conserved under automaton product and abstraction. If additionally G_2 is nonblocking preserving w.r.t. G_1, then they are nonblocking equivalent. We now present a few results.

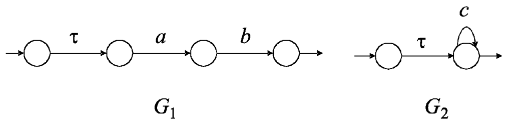


Fig. 3. Example 3: automata G_1 and G_2.

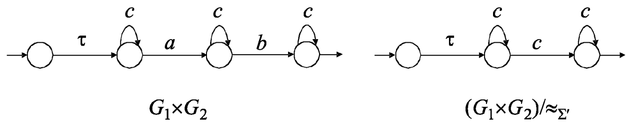


Fig. 4. Example 3: automata $G_1 \times G_2$ and $(G_1 \times G_2)/\approx_{\Sigma'}$.

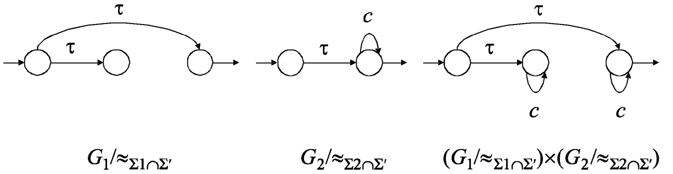

τ
τ
τ
c
τ
τ
c
c
$G_1/{\approx_{\Sigma1\cap\Sigma'}}$
$G_2/{\approx_{\Sigma2\cap\Sigma'}}$
$(G_1/{\approx_{\Sigma1\cap\Sigma'}}){\times}(G_2/{\approx_{\Sigma2\cap\Sigma'}})$

Fig. 5. Example 3: automata $G_1/\approx_{\Sigma_1\cap\Sigma'}, G_2/\approx_{\Sigma_2\cap\Sigma'}$ and $(G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$.

Proposition 2: Given $G_1, G_2 \in \phi(\Sigma), G_3 \in \phi(\Sigma')$, if $G_1 \sqsubseteq G_2$ then $G_1 \times G_3 \sqsubseteq G_2 \times G_3$. □

Corollary 1: Given $G_1, G_2 \in \phi(\Sigma), G_3 \in \phi(\Sigma')$, if $G_1 \cong G_2$ then $G_1 \times G_3 \cong G_2 \times G_3$. □

By Prop. 2 and Cor. 1 nonblocking preserving and equivalence are invariant under automaton product.

Proposition 3: Given $G_i \in \phi(\Sigma_i)$ with $i = 1, 2$, let $\Sigma' \subseteq \Sigma_1 \cup \Sigma_2$. If $\Sigma_1 \cap \Sigma_2 \subseteq \Sigma'$, then

1) $(G_1 \times G_2)/\approx_{\Sigma'} \sqsubseteq (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$.

2) If additionally G_i $(i = 1, 2)$ is marking aware with respect to $\Sigma_i \cap \Sigma'$, then

$$(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'}).$$

□

By Prop. 3, the abstraction of the automaton product is nonblocking preserving with respect to the product of the abstractions; if in addition the marking awareness is imposed then the nonblocking preserving relation can be replaced by the nonblocking equivalence relation. To illustration Prop. 3 we present a simple example. Suppose we have $\Sigma_1 = \{\tau, a, b\}$ and $\Sigma_2 = \{\tau, c\}$. Let $G_1 \in \phi(\Sigma_1)$ and $G_2 \in \phi(\Sigma_2)$ be as shown in Fig. 3, and $\Sigma' = \{\tau, c\} \supseteq \Sigma_1 \cap \Sigma_2$. The results of $G_1 \times G_2$ and $(G_1 \times G_2)/\approx_{\Sigma'}$ are depicted in Fig. 4, and $G_1/\approx_{\Sigma_1\cap\Sigma'}$, $G_2/\approx_{\Sigma_2\cap\Sigma'}$, $(G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$ are in Fig. 5. Clearly

$$(G_1 \times G_2)/\approx_{\Sigma'} \sqsubseteq (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'}).$$

But because

$$B\left((G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})\right) \nsubseteq B\left((G_1 \times G_2)/\approx_{\Sigma'}\right)$$

it is not true that

$$(G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'}) \sqsubseteq (G_1 \times G_2)/\approx_{\Sigma'}.$$

To make G_1 and G_2 marking aware, we need to set $\Sigma' = \{\tau, b\}$. Then by using the same procedure we can check that

$$(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$$

as predicted by Prop. 3.

Theorem 1: Given two alphabets Σ and Σ' with $\Sigma' \subseteq \Sigma$, let $G \in \phi(\Sigma)$ and $S \in \phi(\Sigma')$. Then

1) $B((G/\approx_{\Sigma'}) \times S) = \varnothing \Rightarrow B(G \times S) = \varnothing$;

2) If G is marking aware with respect to Σ', then $B((G/\approx_{\Sigma'}) \times S) = \varnothing$ if and only if $B(G \times S) = \varnothing$.

□

Proof: Let $P : \Sigma^* \rightarrow \Sigma'^*$ be the natural projection

$$\begin{aligned}
& B\left((G/\approx_{\Sigma'}) \times S\right) = \varnothing \\
\Longleftrightarrow\quad & B\left((G/\approx_{\Sigma'}) \times (S/\approx_{\Sigma'})\right) = \varnothing \\
& \text{because } S/\approx_{\Sigma'} \cong S \text{ and by Corollary 1} \\
\Rightarrow\quad & B\left((G \times S)/\approx_{\Sigma'}\right) = \varnothing \text{ by Prop. 3} \\
\Rightarrow\quad & P\left(B(G \times S)\right) = \varnothing \text{ by } \mathit{Prop.}\ 1 \\
\Longleftrightarrow\quad & B(G \times S) = \varnothing.
\end{aligned}$$

Thus, $B((G/\approx_{\Sigma'}) \times S) = \varnothing \Rightarrow B(G \times S) = \varnothing$.

Clearly, S is marking aware with respect to Σ' because $S \in \phi(\Sigma')$. If G is also marking aware with respect to Σ', then by Prop. 3, we have

$$B\left((G \times S)/\approx_{\Sigma'}\right) = B\left((G/\approx_{\Sigma'}) \times (S/\approx_{\Sigma'})\right). \quad (1)$$

Furthermore, $G \times S$ is also marking aware with respect to Σ' because both G and S are marking aware with respect to Σ'. By Prop. 1 we get that

$$P\left(B(G \times S)\right) = B\left((G \times S)/\approx_{\Sigma'}\right). \quad (2)$$

Thus we have

$$\begin{aligned}
& B\left((G/\approx_{\Sigma'}) \times S\right) = \varnothing \\
\Longleftrightarrow\quad & B\left((G/\approx_{\Sigma'}) \times (S/\approx_{\Sigma'})\right) = \varnothing \\
\Longleftrightarrow\quad & B\left((G \times S)/\approx_{\Sigma'}\right) = \varnothing \text{ by Equation 1} \\
\Longleftrightarrow\quad & P\left(B(G \times S)\right) = \varnothing \text{ by Equation 2} \\
\Longleftrightarrow\quad & B(G \times S) = \varnothing.
\end{aligned}$$

Thus, if G is marking aware with respect to Σ', then $B((G/\approx_{\Sigma'}) \times S) = \varnothing \iff B(G \times S) = \varnothing$. ■

Theorem 1 can be interpreted as follows: if the abstraction of G is 'nonconflicting' with S, i.e. $B((G/\approx_{\Sigma'}) \times S) = \varnothing$, then G is 'nonconflicting' with S. The inverse implication is also true if we impose the marking awareness condition. Next, we discuss the usage of abstraction in synthesis.

III. Automaton Abstraction in Supervisor Synthesis

In this section we first introduce concepts of a supervisor synthesis problem, which is to compute a deterministic nonblocking state-controllable, state-observable (or state-normal) supervisor of a nondeterministic plant under a deterministic specification. Then we achieve our main objective of this paper: to establish a connection between the existence of a nonblocking supervisor of a plant and the existence of a nonblocking supervisor of an abstraction of the plant, generated by the proposed abstraction technique.

A. Concepts of a Supervisor Synthesis Problem

Given $G = (X, \Sigma, \xi, x_0, X_m)$, for each $x \in X$ let

$$E_G : X \rightarrow 2^{\Sigma} : x \mapsto E_G(x) := \{\sigma \in \Sigma | \xi(x, \sigma) \neq \varnothing\}.$$

Thus, $E_G(x)$ is simply the set of all events allowable at x in G. We now bring in the concept of *state controllability*. Let $\Sigma =$

$\Sigma_c \dot{\cup} \Sigma_{uc}$, where Σ_c is the set of controllable events, Σ_{uc} is the set of uncontrollable events and $\tau \in \Sigma_{uc}$.

Definition 6: Given $G = (X, \Sigma, \xi, x_0, X_m)$ and $\Sigma' \subseteq \Sigma$, let $A = (Y, \Sigma', \eta, y_0, Y_m)$ and $P : \Sigma^* \to \Sigma'^*$ be the natural projection. A is *state-controllable* with respect to G and Σ_{uc} if for all $s \in L(G \times A)$, $x \in \xi(x_0, s)$ and $y \in \eta(y_0, P(s))$, we have $E_G(x) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_A(y)$. □

The concept of state controllability is slightly different from the one used in the literature, e.g., [1], because of the involvement of Σ'. We can check that A is state controllable implies that $L(G \times A)\Sigma_{uc} \cap L(G) \subseteq L(G \times A)$. This can be briefly shown as follows. Let $s \in L(G \times A)$ and $s\sigma \in L(G)$ with $\sigma \in \Sigma_{uc}$. There must exist $x \in X$ and $y \in Y$ such that $x \in \xi(x_0, s)$, $y \in \eta(y_0, P(s))$ and $\xi(x, \sigma) \neq \varnothing$. Therefore, $\sigma \in E_G(x) \cap \Sigma_{uc}$. There are two cases: (1) $\sigma \in \Sigma'$. Then since A is state-controllable, by Def. 6 we have $\sigma \in E_A(y)$, which means $\sigma \in E_{G\times A}(x, y)$. Thus, $s\sigma \in L(G \times A)$; (2) $\sigma \in \Sigma - \Sigma'$. Then $\xi(x, \sigma) \neq \varnothing$ implies that $\xi \times \eta((x, y), \sigma) \neq \varnothing$. Therefore, we have $s\sigma \in L(G \times A)$. In either case we have $s\sigma \in L(G \times A)$. Thus, it is always true that state controllability implies language controllability described in the RW paradigm. But the reverse statement is not true unless both A and G are deterministic. We now introduce the concept of *state observability*. Let $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, where Σ_o is the set of observable events, Σ_{uo} is the set of unobservable events and $\tau \in \Sigma_{uo}$. Let $P_o : \Sigma^* \to \Sigma_o^*$ be the natural projection.

Definition 7: Given $G = (X, \Sigma, \xi, x_0, X_m)$ and $\Sigma' \subseteq \Sigma$, let $A = (Y, \Sigma', \eta, y_0, Y_m)$. A is *state-observable* with respect to G and P_o if for all $s, s' \in L(G \times A)$ with $P_o(s) = P_o(s')$, and for all $(x, y) \in \xi \times \eta((x_0, y_0), s)$ and $(x', y') \in \xi \times \eta((x_0, y_0), s')$, we have $E_{G\times A}(x, y) \cap E_G(x') \cap \Sigma' \subseteq E_A(y')$. □

State observability defined in Def. 7 is more general than the one defined in [7], as the authors in [7] consider A to be a subautomaton of G and only one event is unobservable. By Def. 7, if A is state observable then for any two states (x, y) and (x', y') in $G \times A$ reachable by two strings s and s' having the same projected image (i.e. $P_o(s) = P_o(s')$), any event σ allowed at (x, y) and x' must be allowed at y' as well. We can check that, if A is state-observable then for all $s, s' \in L(G \times A)$ with $P_o(s) = P_o(s')$ and $\sigma \in \Sigma$

$$s\sigma \in L(G \times A) \wedge s'\sigma \in L(G) \Rightarrow s'\sigma \in L(G \times A).$$

This can be briefly shown as follows. Let $s, s' \in L(G \times A)$ with $P_o(s) = P_o(s')$ and $\sigma \in \Sigma$, $s\sigma \in L(G \times A)$ and $s'\sigma \in L(G)$. There must exist $x, x' \in X$ and $y, y' \in Y$ such that $(x, y) \in \xi \times \eta((x_0, y_0), s)$, $(x', y') \in \xi \times \eta((x_0, y_0), s')$, $\xi \times \eta((x, y), \sigma) \neq \varnothing$, $\xi(x', \sigma) \neq \varnothing$. Clearly, $\sigma \in E_{G\times A}(x, y)$ and $\sigma \in E_G(x')$. There are two cases: (1) $\sigma \in \Sigma'$. Then since A is state-observable, by Def. 7 we have $\sigma \in E_{G\times A}(x, y) \cap E_G(x') \cap \Sigma' \subseteq E_A(y')$. Thus, $\sigma \in E_{G\times A}(x', y')$, which means $s'\sigma \in L(G \times A)$; (2) $\sigma \in \Sigma - \Sigma'$. Then $\xi(x', \sigma) \neq \varnothing$ implies that $\xi \times \eta((x', y'), \sigma) \neq \varnothing$, which means $s'\sigma \in L(G \times A)$. In either case, we have $s'\sigma \in L(G \times A)$. Thus, state observability implies observability defined in [11]. But the inverse statement is not always true unless both A and G are deterministic. Notice that, if $\Sigma_o = \Sigma$, namely every event is observable, A may still not be state-observable, owing to nondeterminism. In many applications we are interested in an even stronger observability property called *state normality* which is defined as follows.

Definition 8: Given $G = (X, \Sigma, \xi, x_0, X_m)$ and $\Sigma' \subseteq \Sigma$, let $A = (Y, \Sigma', \eta, y_0, Y_m)$ and $P : \Sigma^* \to \Sigma'^*$ be the natural projection. A is *state-normal* with respect to G and P_o if for all $s \in L(G \times A)$, $s' \in \overline{P_o^{-1}(P_o(s))} \cap L(G \times A)$ and for all $(x, y) \in \xi \times \eta((x_0, y_0), s')$ and $s'' \in \Sigma^*$, if $P_o(s's'') = P_o(s)$ and $\xi(x, s'') \neq \varnothing$, then $\eta(y, P(s'')) \neq \varnothing$. □

We can check that, if A is state-normal with respect to G and P_o, then $L(G) \cap P_o^{-1}(P_o(L(G \times A))) \subseteq L(G \times A)$, which means $L(G \times A)$ is normal with respect to $L(G)$ and P_o as defined in [11]. This can be briefly shown as follows. Let $s \in L(G) \cap P_o^{-1}(P_o(L(G \times A)))$. Then $s \in L(G)$ and furthermore, there exists $s' \in L(G \times A)$ such that $P_o(s) = P_o(s')$. Since $\epsilon \in L(G \times A)$ and $\epsilon \leq s$, there must exist $t, t' \in \Sigma^*$ with $s = tt'$ such that $t \in L(G \times A)$. Since $P_o(s) = P_o(s')$, we have $t \in \overline{P_o^{-1}(P_o(s'))} \cap L(G \times A)$. Clearly, there exist $(x, y) \in X \times Y$ such that $(x, y) \in \xi \times \eta((x_0, y_0), t)$ and $\xi(x, t') \neq \varnothing$. Since A is state-normal, by Def. 8, we have $\eta(y, P(t')) \neq \varnothing$, which means $\xi \times \eta((x, y), t') \neq \varnothing$. Thus, $s = tt' \in L(G \times A)$. The inverse statement is not true unless both A and G are deterministic. Furthermore, we can check that state normality implies state observability. But the inverse statement is not true.

Definition 9: Given $G \in \phi(\Sigma)$ and $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$, an automaton $S \in \phi(\Sigma')$ is a *nonblocking supervisor* of G under H, if S is deterministic and the following conditions hold:

1) $N(G \times S) \subseteq N(G \times H)$;
2) $B(G \times S) = \varnothing$;
3) S is state-controllable w.r.t. G and Σ_{uc};
4) S is state-observable (or state-normal) w.r.t. G and P_o.

□

The first condition of Def. 9 indicates that $G \times S$, which represents the closed-loop system in the sense that G is supervised by S, complies with the specification H in terms of language inclusion. Because of this condition we only consider H to be deterministic. The use of a nondeterministic specification is described in, e.g. [15], where the goal is to achieve a closed-loop system $G \times S$ that *reduces* the requirement H in terms of failure semantics. Because this paper is about the usage of abstraction in synthesis, which may or may not be applicable to cases with nondeterministic specifications, we decide to use deterministic specifications. For practical applications, it is not necessary that $\tau \in \Delta$. The second condition indicates $G \times S$ is nonblocking. The third and fourth ones are self-explanatory. Later we will use the term "nonblocking state-normal supervisor," when we want to emphasize that S is state-normal with respect to G and P_o. The following result provides a sufficient and necessary condition for the existence of a nonblocking supervisor.

Theorem 2: Given $G \in \phi(\Sigma)$ and $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$, there exists a nonblocking supervisor $S \in \phi(\Sigma')$ of G under H if and only if there exists $A \in \phi(\Sigma')$ with $L(A) = \overline{N(A)}$ such that

1) $N(G \times A) \subseteq N(G \times H)$;
2) $B(G \times A) = \varnothing$;
3) A is state-controllable w.r.t. G and Σ_{uc};
4) A is state-observable (or state-normal) w.r.t. G and P_o.

□

The proof of Theorem 2 indicates that a nonblocking supervisor is simply a recognizer of an automaton A which satisfies those four conditions. In [11], [18] we know that controllability and normality are closed under language union. The following result shows that state controllability and state normality bear a similar feature.

Proposition 4: Given $G \in \phi(\Sigma)$ and $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$, let $S_i \in \phi(\Sigma')$ $(i = 1, 2)$ be a nonblocking state-normal supervisor of G under H and $L(S_i) = \overline{N(S_i)}$. Let $S \in \phi(\Sigma')$ be a deterministic automaton with $N(S) = N(S_1) \cup N(S_2)$ and $L(S) = \overline{N(S)}$. Then S is a nonblocking state-normal supervisor of G w.r.t. H. □

By Prop. 4 the 'union' of two nonblocking state-normal (NSN) supervisors is still a NSN supervisor. We define a set

$$\mathcal{CN}(G, H, \Sigma') := \{S \in \phi(\Sigma') | \, S \text{ is a NSN supervisor of } G \text{ under } H \, \wedge L(S) \subseteq \overline{N(G/\approx_{\Sigma'})}\Big\}.$$

If $L(S) \subseteq \overline{N(G/\approx_{\Sigma'})}$, then S is a nonblocking supervisor of G under H implies that $L(S) = \overline{N(S)}$ because $B(G \times S) = \varnothing$. From Prop. 4 we can derive that $\mathcal{CN}(G, H, \Sigma')$ has a unique element $\hat{S}$ such that for any $S \in \mathcal{CN}(G, H, \Sigma')$, we have $N(S) \subseteq N(\hat{S})$. We call $\hat{S}$ the *supremal nonblocking state-normal supervisor* of G under H with respect to Σ'. In practice we are interested in such a supremal NSN supervisor because it is least restrictive and computable by a procedure proposed in [21]. The reason why we introduce the concept of state-normality is because of the existence of the supremal NSN supervisors, which allows for formal synthesis. Next, we describe how to use the proposed abstraction technique in supervisor synthesis.

B. Abstraction in Nonblocking Supervisor Synthesis

Our main objective is to answer the following two questions: (1) under what conditions is a nonblocking supervisor for an abstraction $G/\approx_{\Sigma'}$ also a nonblocking supervisor for G? (2) under what conditions is a nonblocking supervisor $S \in \phi(\Sigma')$ for G also a nonblocking supervisor for $G/\approx_{\Sigma'}$? To this end we need the following lemmas.

Lemma 1: Let $\Sigma' \subseteq \Sigma$, $G \in \phi(\Sigma)$ and $S \in \phi(\Sigma')$. Then S is state-controllable with respect to $G/\approx_{\Sigma'}$ and $\Sigma_{uc} \cap \Sigma'$ if and only if S is state-controllable with respect to G and Σ_{uc}. □

Lemma 2: Let $\Sigma' \subseteq \Sigma$, $G \in \phi(\Sigma)$, $S \in \phi(\Sigma')$ and $P'_o : \Sigma'^* \to (\Sigma' \cap \Sigma_o)^*$ be the natural projection. Then (1) If S is state-observable w.r.t. $G/\approx_{\Sigma'}$ and P'_o then S is state-observable w.r.t. G and P_o. (2) If $\Sigma_o \subseteq \Sigma'$ and S is state-observable w.r.t. G and P_o, then S is state-observable w.r.t. $G/\approx_{\Sigma'}$ and P_o. □

Lemma 3: Let $\Sigma' \subseteq \Sigma$, $G \in \phi(\Sigma)$, $S \in \phi(\Sigma')$ and $P'_o : \Sigma'^* \to (\Sigma' \cap \Sigma_o)^*$ be the natural projection. Then (1) If S is state-normal w.r.t. $G/\approx_{\Sigma'}$ and P'_o, then S is state-normal w.r.t. G and P_o. (2) If $\Sigma_o \subseteq \Sigma'$ and S is state-normal w.r.t. G and P_o, then S is state-normal w.r.t. $G/\approx_{\Sigma'}$ and P'_o. □

Based on Lemmas 1–3 we present the following result, which answers the first question raised above.

Theorem 3: Given $G \in \phi(\Sigma)$ and $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$, if there exists a nonblocking supervisor $S \in \phi(\Sigma')$ of $G/\approx_{\Sigma'}$ under H, then S is also a nonblocking supervisor of G under H. □

Proof: Since S is a nonblocking supervisor of $G/\approx_{\Sigma'}$ under H, by Def. 9,

1) $N((G/\approx_{\Sigma'}) \times S) \subseteq N((G/\approx_{\Sigma'}) \times H)$;
2) $B((G/\approx_{\Sigma'}) \times S) = \varnothing$;
3) S is state-controllable w.r.t. $G/\approx_{\Sigma'}$ and $\Sigma_{uc} \cap \Sigma'$;
4) S is state-observable (or state-normal) w.r.t. $G/\approx_{\Sigma'}$ and $P'_o : \Sigma'^* \to (\Sigma_o \cap \Sigma')^*$.

By Lemma 1, S is state-controllable with respect to G and Σ_{uc}. By Lemma 2, S is state observable with respect to G and P_o, or by Lemma 3, S is state-normal with respect to G and P_o. Since $B((G/\approx_{\Sigma'}) \times S) = \varnothing$, by Theorem 1 we get that $B(G \times S) = \varnothing$. Finally, we show that $N(G \times S) \subseteq N(G \times H)$ as follows:

$$\begin{aligned}
& N\left((G/\approx_{\Sigma'}) \times S\right) \subseteq N\left((G/\approx_{\Sigma'}) \times H\right) \text{ by (1)} \\
\Rightarrow\ & N(G/\approx_{\Sigma'}) \| N(S) \subseteq N(G/\approx_{\Sigma'}) \| N(H) \\
\Rightarrow\ & N(G) \| N(G/\approx_{\Sigma'}) \| N(S) \subseteq N(G) \| N(G/\approx_{\Sigma'}) \| N(H) \\
\Rightarrow\ & N(G) \| P\left(N(G)\right) \| N(S) \subseteq N(G) \| P\left(N(G)\right) \| N(H) \\
\Rightarrow\ & N(G) \| N(S) \subseteq N(G) \| N(H) \\
\Rightarrow\ & N(G \times S) \subseteq N(G \times H).
\end{aligned}$$

Therefore, the theorem is true. ■

By Theorem 3 a nonblocking supervisor S for $G/\approx_{\Sigma'}$ is also a nonblocking supervisor of G. Therefore, the first question has been answered. To answer the second question raised above, we present another result as follows.

Theorem 4: Given $G \in \phi(\Sigma)$ and $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$, suppose G is marking aware w.r.t. Σ' and $\Sigma_o \subseteq \Sigma'$. Then a nonblocking supervisor $S \in \phi(\Sigma')$ of G under H is also a nonblocking supervisor of $G/\approx_{\Sigma'}$ under H. □

Proof: Since S is a nonblocking supervisor of G under H, by Def. 9,

1) $N(G \times S) \subseteq N(G \times H)$;
2) $B(G \times S) = \varnothing$;
3) S is state-controllable with respect to G and Σ_{uc};
4) S is state-observable (or state-normal) w.r.t. G and P_o.

By Lemma 1, S is state-controllable with respect to $G/\approx_{\Sigma'}$ and $\Sigma_{uc} \cap \Sigma'$. By Lemma 2, S is state-observable with respect to $G/\approx_{\Sigma'}$ and P'_o, or by Lemma 3, S is state-normal with respect to $G/\approx_{\Sigma'}$ and P'_o. Since $B(G \times S) = \varnothing$ and G is marking aware with respect to Σ', by Theorem 1 we get that $B((G/\approx_{\Sigma'}) \times S) = \varnothing$. Finally, we show that $N((G/\approx_{\Sigma'}) \times S) \subseteq N((G/\approx_{\Sigma'}) \times H)$ as follows:

$$\begin{aligned}
& N\left((G/\approx_{\Sigma'}) \times S\right) \\
& \quad = N\left((G/\approx_{\Sigma'}) \times (S/\approx_{\Sigma'})\right) \text{ by } S/\approx_{\Sigma'} \cong S \text{ and Cor. 1} \\
& \quad = N\left((G \times S)/\approx_{\Sigma'}\right) \text{ by Prop3} \\
& \quad = P\left(N(G \times S)\right) \text{ by Prop. 1} \\
& \quad \subseteq P\left(N(G \times H)\right) \text{ By (1)} \\
& \quad = N\left((G \times H)/\approx_{\Sigma'}\right) \text{ by Prop. 1} \\
& \quad = N\left((G/\approx_{\Sigma'}) \times (H/\approx_{\Sigma'})\right) \text{ by Prop. 3} \\
& \quad = N\left((G/\approx_{\Sigma'}) \times H\right) \text{ by } H/\approx_{\Sigma'} \cong H \text{ and Cor. 1.}
\end{aligned}$$

Therefore, the theorem is true. ■

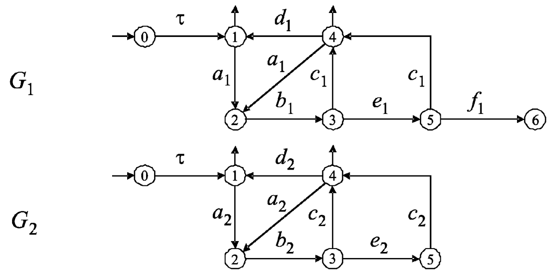


Fig. 6. Example 4: a simple processing unit.

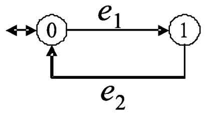
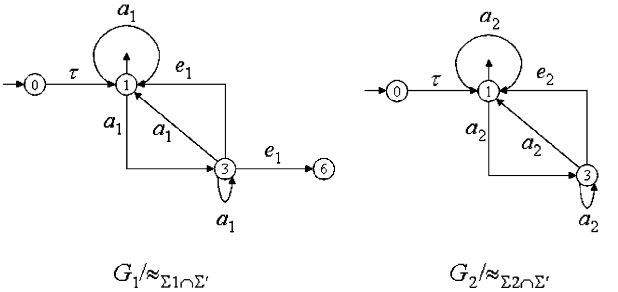


Fig. 7. Example 4: the specification $H \in \phi(\Delta)$.



$G_1/\approx_{\Sigma 1\cap\Sigma'}$ $G_2/\approx_{\Sigma 2\cap\Sigma'}$

Fig. 8. Example 4: abstractions $G_1/\approx_{\Sigma_1\cap\Sigma'}$ and $G_2/\approx_{\Sigma_2\cap\Sigma'}$.

By Theorem 4, if G is marking aware with respect to Σ' and $\Sigma_o \subseteq \Sigma'$, then a nonblocking supervisor of G is also a nonblocking supervisor of $G/\approx_{\Sigma'}$, which means, under conditions of Theorem 4, we have

$$\mathcal{CN}(G,H,\Sigma') \subseteq \mathcal{CN}(G/\approx_{\Sigma'},H,\Sigma').$$

On the other hand, by Theorem 3 we have

$$\mathcal{CN}(G/\approx_{\Sigma'},H,\Sigma') \subseteq \mathcal{CN}(G,H,\Sigma').$$

Thus, if G is marking aware with respect to Σ' and $\Sigma_o \subseteq \Sigma'$, we have $\mathcal{CN}(G,H,\Sigma') = \mathcal{CN}(G/\approx_{\Sigma'},H,\Sigma')$, which means the supremal nonblocking state-normal supervisor of $G/\approx_{\Sigma'}$ under H is also the supremal nonblocking state-normal supervisor of G under H, whose alphabet is Σ'. When the supervisor alphabet Σ' is not specified a priori, it is an open question whether there exists a minimal Σ' such that the supremal nonblocking state-normal supervisor of the corresponding abstraction can also achieve the maximal permissiveness for the original plant. Next, we use a simple example to illustrate the relevant concepts and the process of using abstraction in synthesis.

IV. Example

Suppose we have models of two machines, which are part of one processing unit and functionally identical, except for individual event labels. The system is depicted in Fig. 6. Each machine G_i ($i = 1, 2$) has the following standard operations: 1) fetching a work piece (a_i); 2) preprocessing (b_i); 3) postprocessing (c_i); 4) polishing (e_i); 5) packaging (d_i). After preprocessing b_i, there are two choices: to be postprocessed directly (c_i) or to be polished first (e_i) before postprocessing. The latter gives a product with better quality. The negative aspect is that polishing may cause the machine G_1 to fail (f_1). If failure does happen, G_1 will stop automatically and wait for repair. Among each alphabet Σ_i, the controllable alphabet is $\Sigma_{i,c} = \{a_i, e_i\}$, and for the purpose of simplicity the observable alphabet $\Sigma_{i,o} = \Sigma_i - \{\tau\}$, namely every event except for τ is observable. There is one specification $H \in \phi(\Delta)$ with $\Delta = \{e_1, e_2\}$, depicted in Fig. 7, indicating that if a work piece is polished in $G_1(e_1)$, then a work piece must be polished in G_2 afterwards (e_2). We now start to synthesize a nonblocking supervisor for $G_1 \times G_2$ that complies with the specification H.

First, we create an appropriate abstraction of $G_1 \times G_2$. We pick $\Sigma' = \{\tau, a_1, a_2, e_1, e_2\}$. The rationality is that, since $\Delta \subseteq \Sigma'$, the abstraction $(G_1 \times G_2)/\approx_{\Sigma'}$ can capture constraints imposed by the specification H; and since all controllable events are in Σ', the abstraction $(G_1 \times G_2)/\approx_{\Sigma'}$ also contains all means of control available to $G_1 \times G_2$ itself. Since $\Sigma_1 \cap \Sigma_2 = \{\tau\} \subseteq \Sigma'$, by Prop. 3

$$(G_1 \times G_2)/\approx_{\Sigma'} \sqsubseteq (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'}).$$

The results of $G_1/\approx_{\Sigma_1\cap\Sigma'}$ and $G_2/\approx_{\Sigma_2\cap\Sigma'}$ are depicted in Fig. 8. The product of two abstractions $G' := (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$ is depicted in Fig. 9, We now use G' and H to synthesize a supervisor. The product $G' \times H$ is depicted in Fig. 9. Clearly, the transitions e_1 from state (2,0) to state (3,1), and from (5,0) to (4,1) in $G' \times H$ must be disabled. Otherwise, blocking states (3,1) and (4,1) will be reached. Once these two transitions are disabled, transitions e_1 from (2,0) to (1,1), and from (5,0) to (6,1) must be disabled as well because, otherwise, the remaining automaton is neither state-normal nor state-observable. After removing transitions e_1 at states (2,0) and (5,0) in Fig. 9, the remaining reachable part A is depicted in Fig. 10, which is nonblocking, state-controllable, state-normal (and state-observable). By Theorem 2 we get that a recognizer S of the marked behavior $N(A)$, depicted in Fig. 11, is a nonblocking supervisor of G' under H. We can see that S does not allow events e_1 and e_2 to happen. It is not difficult to check that S is a nonblocking supervisor of $G_1 \times G_2$ under H, as predicted by Theorem 3. We can verify that the maximum number of states of any intermediate automaton computed is 13, which occurs when we compute $G' \times H$. Clearly, abstractions help to reduce the computational complexity in this example because otherwise we will have to face the product $G_1 \times G_2 \times H$ directly, which has 61 states.

The abstraction technique has been applied to a semiconductor cluster tool example in [21], where the monolithic plant model has about 2.68×10^8 states and, as a contrast, the largest abstraction has only 985 states. Thus, the abstraction-based synthesis shows a significant computational advantage over centralized synthesis. It has also been applied to a cable service network example in [20], where the ratio of the sizes of state sets of abstractions obtained by using our approach and the observer-based approach is $(1.25)^n$, where n denotes the number

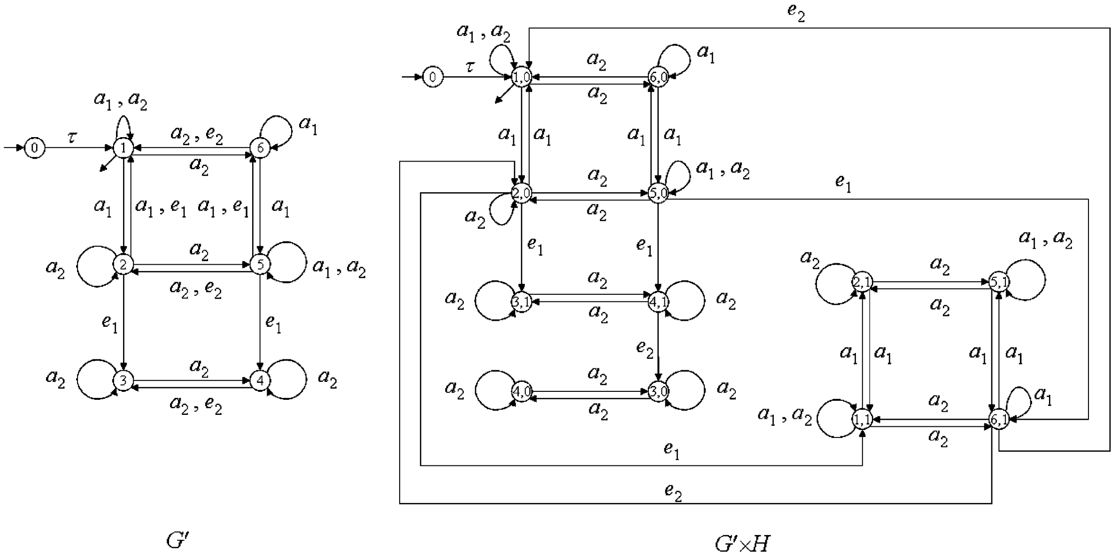


Fig. 9. Example 4: the product $G' = (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$ and the product $G' \times H$.

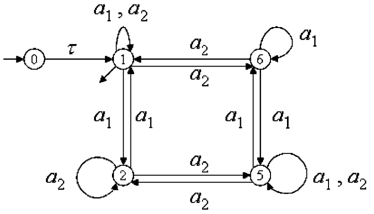


Fig. 10. Example 4: nonblocking, state-controllable, state-observable (and state-normal) automaton A.

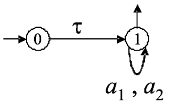


Fig. 11. Example 4: the supervisor $S \in \phi(\Sigma')$.

of residents in a community. Clearly, our abstraction approach enjoys a computational advantage over the observer-based abstraction approach. We are applying this technique to other case studies at the moment to test its efficiency compared with other automaton-based abstraction techniques in the literature.

V. Conclusion

In this paper, we first present a new technique that computes an abstraction of a nondeterministic finite-state automaton and provide some relevant properties. Then we show the usage of this technique in a synthesis problem, where supervisors and specifications are deterministic but plant models are nondeterministic. After introducing the concepts of state controllability, state observability and state normality, we show that a nonblocking supervisor of an abstraction $G/\approx_{\Sigma'}$ under a specification H is also a nonblocking supervisor of the original plant G under the same specification. The inverse statement is true, if all observable events are contained in Σ' and the plant G is marking aware with respect to Σ'. In this paper we also present a sufficient and necessary condition for the existence of a nonblocking supervisor and show that the supremal nonblocking state-normal supervisor exists for a plant G and a specification H. The concrete procedure to compute such a supremal supervisor is not provided, owing to the different objective of this paper and the page limit as well. It is addressed in another paper of the authors [21].

Although results in this paper are about standardized automata, they are applicable in a supervisor synthesis problem, where G is non-standardized in the sense that $\tau \notin \Sigma$. To do this, we first standardize G to obtain G^τ, then synthesize a standardized nonblocking supervisor S^τ based on $G^\tau/\approx_{\Sigma'}$. Since S^τ is deterministic, we can convert it to a non-standardized automaton by simply removing the τ transition and setting the target state of the τ transition as the initial state of the resultant automaton S. Since τ is uncontrollable and unobservable, we can show that S is a nonblocking supervisor of G, which is introduced in [21] for aggregative synthesis of distributed supervisors.

Appendix

1) Proof of Prop. 1: Let ξ' be the transition map of $G/\approx_{\Sigma'}$. First we show that $P(B(G)) \subseteq B(G/\approx_{\Sigma'})$. For each string $s \in P(B(G))$, there exists $t \in B(G)$ with $P(t) = s$ such that

$$(\exists x \in \xi(x_0, t))\,(\forall t' \in \Sigma^*)\xi(x, t') \cap X_m = \varnothing.$$

Since G is standardized, $P(t) \neq \epsilon$ iff $t \neq \epsilon$. Thus, we get that $\langle x\rangle \in \xi'(\langle x_0\rangle, P(t))$. Because

$$(\forall t' \in \Sigma^*)\xi(x, t') \cap X_m = \varnothing$$

we have that, $(\forall s' \in \Sigma'^*)\xi'(\langle x\rangle, s') \cap (X_m/\approx_{\Sigma'}) = \varnothing$. Thus, $s = P(t) \in B(G/\approx_{\Sigma'})$.

To show $P(N(G)) \subseteq N(G/\approx_{\Sigma'})$, let $s \in P(N(G))$. Then

$$(\exists t \in N(G))\,P(t) = s \wedge \xi(x_0, t) \cap X_m \neq \varnothing.$$

Since G is standardized, from $\xi(x_0, t) \cap X_m \neq \varnothing$ we have $\xi'(\langle x_0\rangle, P(t)) \cap (X_m/\approx_{\Sigma'}) \neq \varnothing$. Thus, $s \in N(G/\approx_{\Sigma'})$. To show $N(G/\approx_{\Sigma'}) \subseteq P(N(G))$, let $s \in N(G/\approx_{\Sigma'})$. Then we have $\xi'(\langle x_0\rangle, s) \cap (X_m/\approx_{\Sigma'}) \neq \varnothing$, which means, there exists $t \in \Sigma^*$ with $P(t) = s$ such that $\xi(x_0, t) \cap X_m \neq \varnothing$. Thus, $t \in N(G)$, namely $P(t) = s \in P(N(G))$. Therefore, we have $P(N(G)) = N(G/\approx_{\Sigma'})$.

Finally, suppose G is marking aware with respect to Σ'. To show $B(G/\approx_{\Sigma'}) = P(B(G))$, we only need to show that $B(G/\approx_{\Sigma'}) \subseteq P(B(G))$. For each string $s \in B(G/\approx_{\Sigma'})$, there exists $\langle x \rangle \in \xi'(\langle x_0 \rangle, s)$ such that

$$(\forall s' \in \Sigma'^*)\xi'(\langle x \rangle, s') \cap (X_m/\approx_{\Sigma'}) = \varnothing$$

from which we can derive that, there exists $t \in \Sigma^*$ such that $P(t) = s$ and

$$x \in \xi(x_0, t) \wedge (\forall t' \in \Sigma^*)\xi(x, t') \cap X_m \neq \varnothing \Rightarrow t' \in (\Sigma - \Sigma')^*.$$

Clearly, $x \notin X_m$, because otherwise $\xi'(\langle x \rangle, \epsilon) \cap (X_m/\approx_{\Sigma'}) \neq \varnothing$. We claim that x is a blocking state of G. Otherwise, there exists $t' \in \Sigma^*$ such that $\xi(x, t') \cap X_m \neq \varnothing$. Since G is marking aware with respect to Σ', we have that $t' \notin (\Sigma - \Sigma')^*$, which contradicts the fact that

$$(\forall t' \in \Sigma^*)\xi(x, t') \cap X_m \neq \varnothing \Rightarrow t' \in (\Sigma - \Sigma')^*.$$

Thus, the claim is true. Since x is a blocking state, we have $t \in B(G)$, namely $s = P(t) \in P(B(G))$. ■

2) Proof of Prop. 2: Let $G_i = (X_i, \Sigma_i, \xi_i, x_{i,0}, X_{i,m})$ with $i = 1, 2, 3$, where $\Sigma_1 = \Sigma_2 = \Sigma$ and $\Sigma_3 = \Sigma'$. Let $P : (\Sigma \cup \Sigma')^* \rightarrow \Sigma^*$ and $P' : (\Sigma \cup \Sigma')^* \rightarrow \Sigma'^*$ be natural projections. We first show that $N(G_1 \times G_3) = N(G_2 \times G_3)$. Clearly, we have $N(G_1 \times G_3) = N(G_1) \| N(G_3)$. Since $G_1 \sqsubseteq G_2$, we have $N(G_1) = N(G_2)$. Thus, we have $N(G_1 \times G_3) = N(G_1) \| N(G_3) = N(G_2) \| N(G_3) = N(G_2 \times G_3)$.

To show that $B(G_1 \times G_3) \subseteq B(G_2 \times G_3)$, let $s \in B(G_1 \times G_3)$. By the definition of automaton product, there exists $x_1 \in X_1$ such tat $x_1 \in \xi_1(x_{1,0}, P(s))$. There are two cases to consider. Case 1: x_1 is a blocking state. Then $P(s) \in B(G_1) \subseteq B(G_2)$. Thus, $s \in B(G_2 \times G_3)$. Case 2: x_1 is a nonblocking state. Since $G_1 \sqsubseteq G_2$, there exists $x_2 \in \xi_2(x_{2,0}, P(s))$ such that $N_{G_1}(x_1) \supseteq N_{G_2}(x_2)$. Since $s \in B(G_1 \times G_3)$, there exists $x_3 \in X_3$ such that $(x_1, x_3) \in \xi_1 \times \xi_3((x_{1,0}, x_{3,0}), s)$ and $N_{G_1 \times G_3}(x_1, x_3) = \varnothing$. We have

$$\begin{aligned} N_{G_2 \times G_3}(x_2, x_3) &= N_{G_2}(x_2) \| N_{G_3}(x_3) \\ &\subseteq N_{G_1}(x_1) \| N_{G_3}(x_3) \\ &= N_{G_1 \times G_3}(x_1, x_3) = \varnothing \end{aligned} \quad .$$

Thus, (x_2, x_3) is a blocking state of $G_2 \times G_3$, which means $s \in B(G_2 \times G_3)$. Therefore, in either case we have $B(G_1 \times G_3) \subseteq B(G_2 \times G_3)$.

Finally, from the above argument in Case 2, for any $s \in (\Sigma \cup \Sigma')^*$ and $(x_1, x_3) \in \xi_1 \times \xi_3((x_{1,0}, x_{3,0}), s)$, we have $(x_2, x_3) \in \xi_2 \times \xi_3((x_{2,0}, x_{3,0}), s)$ such that $N_{G_2 \times G_3}(x_2, x_3) \subseteq N_{G_1 \times G_3}(x_1, x_3)$. ■

3) Proof of Prop. 3: Let $G_i = (X_i, \Sigma_i, \xi_i, x_{i,0}, X_{i,m}) \in \phi(\Sigma_i)$ with $i = 1, 2$. For notation simplicity let $\hat{\Sigma}_i = \Sigma_i \cap \Sigma'$, and $P : (\Sigma_1 \cup \Sigma_2)^* \rightarrow \Sigma'^*$, $P_i : \Sigma_i^* \rightarrow \hat{\Sigma}_i^*$, $\hat{P}_i : \Sigma'^* \rightarrow \hat{\Sigma}_i^*$ and $Q_i : (\Sigma_1 \cup \Sigma_2)^* \rightarrow \Sigma_i^*$ be natural projections, ξ' for the transition map of $(G_1 \times G_2)/\approx_{\Sigma'}$ and ξ'_i for the transition map of $G_i/\approx_{\hat{\Sigma}_i}$ $(i = 1, 2)$.

First, we have the following:

$$\begin{aligned} & N\left((G_1 \times G_2)/\approx_{\Sigma'}\right) \\ &\quad = P\left(N(G_1 \times G_2)\right) \text{ by Prop. 1} \\ &\quad = P\left(N(G_1) \| N(G_2)\right) \\ &\quad = P_1\left(N(G_1)\right) \| P_2\left(N(G_2)\right) \text{ because } \Sigma_1 \cap \Sigma_2 \subseteq \Sigma' \\ &\quad = N\left(G_1/\approx_{\hat{\Sigma}_1}\right) \| N\left(G_2/\approx_{\hat{\Sigma}_2}\right) \text{ by Prop. 1} \\ &\quad = N\left(\left(G_1/\approx_{\hat{\Sigma}_1}\right) \times \left(G_2/\approx_{\hat{\Sigma}_2}\right)\right). \end{aligned}$$

Next, we show that

$$B\left((G_1 \times G_2)/\approx_{\Sigma'}\right) \subseteq B\left(\left(G_1/\approx_{\hat{\Sigma}_1}\right) \times \left(G_2/\approx_{\hat{\Sigma}_2}\right)\right).$$

Let $s \in B((G_1 \times G_2)/\approx_{\Sigma'})$. Then there exists $(x_1, x_2) \in X_1 \times X_2$ such that $\langle (x_1, x_2) \rangle_{\Sigma'} \in \xi'(\langle (x_{1,0}, x_{2,0}) \rangle_{\Sigma'}, s)$ and for all $s' \in \Sigma'^*$

$$\xi'\left(\langle (x_1, x_2) \rangle_{\Sigma'}, s'\right) \cap \left((X_{1,m} \times X_{2,m})/\approx_{\Sigma'}\right) = \varnothing$$

which means $(x_1, x_2) \notin X_{1,m} \times X_{2,m}$ and there exists $t \in (\Sigma_1 \cup \Sigma_2)^*$ with $P(t) = s$ such that $(x_1, x_2) \in \xi_1 \times \xi_2((x_{1,0}, x_{2,0}), t)$ and for all $t' \in \Sigma^*$

$$\begin{aligned} & \xi_1 \times \xi_2\left((x_1, x_2), t'\right) \cap \left(X_{1,m} \times X_{2,m}\right) \neq \varnothing \\ &\quad \Rightarrow t' \in \left((\Sigma_1 \cup \Sigma_2) - \Sigma'\right)^*. \end{aligned}$$

Since G_1 and G_2 are standardized, from $(x_1, x_2) \in \xi_1 \times \xi_2((x_{1,0}, x_{2,0}), t)$ and the fact that $\Sigma_1 \cap \Sigma_2 \subseteq \Sigma'$ we can derive that

$$\left(\langle x_1 \rangle_{\hat{\Sigma}_1}, \langle x_2 \rangle_{\hat{\Sigma}_2}\right) \in \xi'_1 \times \xi'_2\left(\left(\langle x_{1,0} \rangle_{\hat{\Sigma}_1}, \langle x_{2,0} \rangle_{\hat{\Sigma}_1}\right), s\right).$$

We claim that $(\langle x_1 \rangle_{\hat{\Sigma}_1}, \langle x_2 \rangle_{\hat{\Sigma}_2})$ is a blocking state of $(G_1/\approx_{\hat{\Sigma}_1}) \times (G_2/\approx_{\hat{\Sigma}_2})$. Otherwise, there exists $s' \in \Sigma'^*$ such that

$$\begin{aligned} & \xi'_1 \times \xi'_2\left(\left(\langle x_1 \rangle_{\hat{\Sigma}_1}, \langle x_2 \rangle_{\hat{\Sigma}_2}\right), s'\right) \cap \\ &\qquad \left(\left(X_{1,m}/\approx_{\hat{\Sigma}_1}\right) \times \left(X_{2,m}/\approx_{\hat{\Sigma}_2}\right)\right) \neq \varnothing. \end{aligned}$$

Since $(x_1, x_2) \notin X_{1,m} \times X_{2,m}$, we get that $(\langle x_1 \rangle_{\hat{\Sigma}_1}, \langle x_2 \rangle_{\hat{\Sigma}_2}) \notin (X_{1,m}/\approx_{\hat{\Sigma}_1}) \times (X_{2,m}/\approx_{\hat{\Sigma}_2})$. Thus, $s' \neq \epsilon$, which means there exists $t' \in \Sigma^*$ with $P(t') = s' \notin ((\Sigma_1 \cup \Sigma_2) - \Sigma')^*$ such that $\xi_1 \times \xi_2((x_1, x_2), t') \cap (X_{1,m} \times X_{2,m}) \neq \varnothing$—contradict the fact that for all $t' \in \Sigma^*$

$$\begin{aligned} & \xi_1 \times \xi_2\left((x_1, x_2), t'\right) \cap \left(X_{1,m} \times X_{2,m}\right) \neq \varnothing \\ &\quad \Rightarrow t' \in \left((\Sigma_1 \cup \Sigma_2) - \Sigma'\right)^*. \end{aligned}$$

From the claim we get that $s \in B((G_1/\approx_{\hat{\Sigma}_1}) \times (G_2/\approx_{\hat{\Sigma}_2}))$.

Let $s \in \overline{N((G_1 \times G_2)/\approx_{\Sigma'})}$. For any $(x_1, x_2) \in X_1 \times X_2$ with $\langle (x_1, x_2) \rangle_{\Sigma'} \in \xi'(\langle (x_{1,0}, x_{2,0}) \rangle_{\Sigma'}, s)$, there exists $t \in \Sigma^*$ such that $P(t) = s$ and $(x_1, x_2) \in \xi((x_{1,0}, x_{2,0}), t)$. Since G_1 and G_2 are standardize, if $s = \epsilon$, then $t = \epsilon$, which means $(x_1, x_2) = (x_{1,0}, x_{2,0})$. Clearly, we have the following:

$$\left(\langle x_{1,0} \rangle_{\hat{\Sigma}_1}, \langle x_{2,0} \rangle_{\hat{\Sigma}_2}\right) \in \xi'_1 \times \xi'_2\left(\left(\langle x_{1,0} \rangle_{\hat{\Sigma}_1}, \langle x_{2,0} \rangle_{\hat{\Sigma}_2}\right), \epsilon\right).$$

If $s \neq \epsilon$, then by the definition of automaton abstraction and the assumption that $\Sigma_1 \cap \Sigma_2 \subseteq \Sigma'$, we get

$$\left(\langle x_1\rangle_{\hat\Sigma_1}, \langle x_2\rangle_{\hat\Sigma_2}\right) \in \xi_1' \times \xi_2' \left(\left(\langle x_{1,0}\rangle_{\hat\Sigma_1}, \langle x_{2,0}\rangle_{\hat\Sigma_2}\right), s\right).$$

Thus, in either case we have

$$\left(\langle x_1\rangle_{\hat\Sigma_1}, \langle x_2\rangle_{\hat\Sigma_2}\right) \in \xi_1' \times \xi_2' \left(\left(\langle x_{1,0}\rangle_{\hat\Sigma_1}, \langle x_{2,0}\rangle_{\hat\Sigma_2}\right), s\right).$$

We now show that

$$N_{\left(G_1/\approx_{\hat\Sigma_1}\right)\times\left(G_2/\approx_{\hat\Sigma_2}\right)}\left(\langle x_1\rangle_{\hat\Sigma_1}, \langle x_2\rangle_{\hat\Sigma_2}\right) \subseteq N_{(G_1\times G_2)/\approx_{\Sigma'}}\left(\langle(x_1,x_2)\rangle_{\Sigma'}\right).$$

Let $s' \in N_{(G_1/\approx_{\hat\Sigma_1})\times(G_2/\approx_{\hat\Sigma_2})}(\langle x_1\rangle_{\hat\Sigma_1}, \langle x_2\rangle_{\hat\Sigma_2})$. If $s'=\epsilon$, then $(\langle x_1\rangle_{\hat\Sigma_1}, \langle x_2\rangle_{\hat\Sigma_2}) \in (X_{1,m}/\approx_{\hat\Sigma_1}) \times (X_{2,m}/\approx_{\hat\Sigma_2})$, from which we can derive that $(x_1,x_2)\in X_{1,m}\times X_{2,m}$. Thus, $\langle(x_1,x_2)\rangle_{\Sigma'} \in (X_{1,m}\times X_{2,m})/\approx_{\Sigma'}$, namely $\epsilon \in N_{(G_1\times G_2)/\approx_{\Sigma'}}(\langle(x_1,x_2)\rangle_{\Sigma'})$. If $s'\neq\epsilon$, then there exists $t'\in\Sigma^*$ with $P(t')=s'$ such that $\xi_1\times\xi_2((x_1,x_2),t')\cap(X_{1,m}\times X_{2,m})\neq\varnothing$. Since $P(t')\neq\epsilon$, by the definition of automaton abstraction, we get that $\xi'(\langle(x_1,x_2)\rangle_{\Sigma'},s')\cap(X_{1,m}\times X_{2,m})/\approx_{\Sigma'}\neq\varnothing$. Thus, $s'\in N_{(G_1\times G_2)/\approx_{\Sigma'}}(\langle(x_1,x_2)\rangle_{\Sigma'})$. In either case, we have

$$N_{\left(G_1/\approx_{\hat\Sigma_1}\right)\times\left(G_2/\approx_{\hat\Sigma_2}\right)}\left(\langle x_1\rangle_{\hat\Sigma_1}, \langle x_2\rangle_{\hat\Sigma_2}\right) \subseteq N_{(G_1\times G_2)/\approx_{\Sigma'}}\left(\langle(x_1,x_2)\rangle_{\Sigma'}\right).$$

Thus, $(G_1\times G_2)/\approx_{\Sigma'}\sqsubseteq(G_1/\approx_{\hat\Sigma_1})\times(G_2/\approx_{\hat\Sigma_2})$.

Suppose G_i ($i=1,2$) is marking aware with respect to $\Sigma_i\cap\Sigma'$. To show

$$B\left(\left(G_1/\approx_{\hat\Sigma_1}\right)\times\left(G_2/\approx_{\hat\Sigma_2}\right)\right)=B\left((G_1\times G_2)/\approx_{\Sigma'}\right)$$

we only need to prove one direction ($\subseteq$), because the other direction ($\supseteq$) has been proved. Let $s\in B((G_1/\approx_{\hat\Sigma_1})\times(G_2/\approx_{\hat\Sigma_2}))$. Then there exists $(x_1,x_2)\in X_1\times X_2$ such that

$$\left(\langle x_1\rangle_{\hat\Sigma_1}, \langle x_2\rangle_{\hat\Sigma_2}\right) \in \xi_1' \times \xi_2' \left(\left(\langle x_{1,0}\rangle_{\hat\Sigma_1}, \langle x_{2,0}\rangle_{\hat\Sigma_2}\right), s\right) \quad (3)$$

and

$$(\forall s'\in\Sigma'^*)\xi_1'\times\xi_2'\left(\left(\langle x_1\rangle_{\hat\Sigma_1},\langle xx_2\rangle_{\hat\Sigma_2}\right),s'\right)\cap W=\varnothing \quad (4)$$

where $W=((X_{1,m}/\approx_{\hat\Sigma_1})\times(X_{2,m}/\approx_{\hat\Sigma_2}))$. From Expression (3) we get that

$$(\exists t\in(\Sigma_1\cup\Sigma_2)^*)\,P(t)=s\wedge(x_1,x_2)\in\xi_1\times\xi_2\left((x_{1,0},x_{2,0}),t\right). \quad (5)$$

From Expression (4) we get that $(x_1,x_2)\notin X_{1,m}\times X_{2,m}$. Since G_1 and G_2 are standardized, from Expression (5) and the fact that $\Sigma_1\cap\Sigma_2\subseteq\Sigma'$ we have

$$\langle(x_1,x_2)\rangle_{\Sigma'}\in\xi'\left(\langle(x_{1,0},x_{2,0})\rangle_{\Sigma'},s\right).$$

We claim that $\langle(x_1,x_2)\rangle_{\Sigma'}$ is a blocking state of $(G_1\times G_2)/\approx_{\Sigma'}$. Otherwise, there exists $s'\in\Sigma'^*$ such that

$$\xi'\left(\langle(x_1,x_2)\rangle_{\Sigma'},s'\right)\cap(X_{1,m}\times X_{2,m})/\approx_{\Sigma'}\neq\varnothing.$$

Since $(x_1,x_2)\notin X_{1,m}\times X_{2,m}$, we get that $\langle(x_1,x_2)\rangle_{\Sigma'}\notin(X_{1,m}\times X_{2,m})/\approx_{\Sigma'}$. Thus, $s'\neq\epsilon$. Furthermore, since G_i ($i=1,2$) is marking aware with respect to $\Sigma_i\cap\Sigma'$, we have $\hat P_i(s')\neq\epsilon$. Thus, there exists $t'\in\Sigma^*$ with $P(t')=s'$ such that $\xi_1\times\xi_2((x_1,x_2),t')\cap(X_{1,m}\times X_{2,m})\neq\varnothing$. Since $\hat P_i(s')\neq\epsilon$ for $i=1$, 2 and $\Sigma_1\cap\Sigma_2\subseteq\Sigma'$, we have

$$\xi_1'\times\xi_2'\left(\left(\langle x_1\rangle_{\hat\Sigma_1},\langle x_2\rangle_{\hat\Sigma_2}\right),s'\right)\cap\left(\left(X_{1,m}/\approx_{\hat\Sigma_1}\right)\times\left(X_{2,m}/\approx_{\hat\Sigma_2}\right)\right)\neq\varnothing$$

which contradicts Expression (4). Thus, the claim is true, from which we have $s\in B((G_1\times G_2)/\approx_{\Sigma'})$.

Let $s\in\overline{N((G_1/\approx_{\hat\Sigma_1})\times(G_2/\approx_{\hat\Sigma_2}))}$. For all $(x_1,x_2)\in X_1\times X_2$ with $(\langle x_1\rangle_{\hat\Sigma_1},\langle x_2\rangle_{\hat\Sigma_2})\in\xi_1'\times\xi_2'((\langle x_{1,0}\rangle_{\hat\Sigma_1},\langle x_{2,0}\rangle_{\hat\Sigma_2}),s)$, there exists $t\in\Sigma^*$ such that

$$P(t)=s\wedge(x_1,x_2)\in\xi\left((x_{1,0},x_{2,0}),t\right).$$

Since G_1 and G_2 are standardized, if $s=\epsilon$, then $t=\epsilon$, which means $(x_1,x_2)=(x_{1,0},x_{2,0})$. Clearly, we have $\langle(x_{1,0},x_{2,0})\rangle_{\Sigma'}\in\xi'(\langle(x_{1,0},x_{2,0})\rangle_{\Sigma'},\epsilon)$. If $s\neq\epsilon$, then by the definition of abstraction and the assumption that $\Sigma_1\cap\Sigma_2\subseteq\Sigma'$, we get

$$\langle(x_1,x_2)\rangle_{\Sigma'}\in\xi'\left(\langle(x_{1,0},x_{2,0})\rangle_{\Sigma'},s\right).$$

Thus, in either case we have

$$\langle(x_1,x_2)\rangle_{\Sigma'}\in\xi'\left(\langle(x_{1,0},x_{2,0})\rangle_{\Sigma'},s\right).$$

We now show that

$$N_{(G_1\times G_2)/\approx_{\Sigma'}}\left(\langle(x_1,x_2)\rangle_{\Sigma'}\right)\subseteq N_{\left(G_1/\approx_{\hat\Sigma_1}\right)\times\left(G_2/\approx_{\hat\Sigma_2}\right)}\left(\langle x_1\rangle_{\hat\Sigma_1},\langle x_2\rangle_{\hat\Sigma_2}\right).$$

Let $s'\in N_{(G_1\times G_2)/\approx_{\Sigma'}}(\langle(x_1,x_2)\rangle_{\Sigma'})$. If $s'=\epsilon$, then

$$\langle(x_1,x_2)\rangle_{\Sigma'}\in(X_{1,m}\times X_{2,m})/\approx_{\Sigma'}$$

from which we can derive that $(x_1,x_2)\in X_{1,m}\times X_{2,m}$. Thus

$$\left(\langle x_1\rangle_{\hat\Sigma_1},\langle x_2\rangle_{\hat\Sigma_2}\right)\in\left(X_{1,m}/\approx_{\hat\Sigma_1}\right)\times\left(X_{2,m}/\approx_{\hat\Sigma_2}\right)$$

namely $\epsilon\in N_{(G_1/\approx_{\hat\Sigma_1})\times(G_2/\approx_{\hat\Sigma_2})}(\langle x_1\rangle_{\hat\Sigma_1},\langle x_2\rangle_{\hat\Sigma_2})$. If $s'\neq\epsilon$. Then $\langle(x_1,x_2)\rangle_{\Sigma'}\notin(X_{1,m}\times X_{2,m})/\approx_{\Sigma'}$, which means $(x_1,x_2)\notin(X_{1,m}\times X_{2,m})/\approx_{\Sigma'}$. Furthermore, there exists $t'\in\Sigma^*$ with $P(t')=s'$ such that $\xi_1\times\xi_2((x_1,x_2),t')\cap(X_{1,m}\times X_{2,m})\neq\varnothing$. We consider three cases. Case 1: $\hat P_i(s')\neq\epsilon$ ($i=1$, 2), namely $x_1\notin X_{1,m}$ and $x_2\notin X_{2,m}$. By the definition of abstraction, we have

$$\xi_1' \times \xi_2' \left(\left(\langle x_1\rangle_{\hat{\Sigma}_1}, \langle x_2\rangle_{\hat{\Sigma}_2}\right), s'\right) \cap \left(\left(X_{1,m}/\approx_{\hat{\Sigma}_1}\right) \times \left(X_{2,m}/\approx_{\hat{\Sigma}_2}\right)\right) \neq \varnothing.$$

Thus, $s' \in N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(\langle x_1\rangle_{\hat{\Sigma}_1}, \langle x_2\rangle_{\hat{\Sigma}_2})$. Case 2: $\hat{P}_1(s') = \epsilon$ and $\hat{P}_2(s') = s' \neq \epsilon$. Since G_1 is marking aware with respect to $\Sigma_1 \cap \Sigma'$, $\hat{P}_1(s') = \epsilon$ implies that $x_1 \in X_{1,m}$. Since $\hat{P}_2(s') = s' \neq \epsilon$, we have

$$\left(\exists \langle \hat{x}_2\rangle_{\hat{\Sigma}_2} \in X_{2,m}/\approx_{\hat{\Sigma}_2}\right) \langle \hat{x}_2\rangle_{\hat{\Sigma}_2} \in \xi_2'\left(\langle x_2\rangle_{\hat{\Sigma}_2}, \hat{P}_2(s')\right).$$

Thus

$$\left(\langle x_1\rangle_{\hat{\Sigma}_1}, \langle \hat{x}_2\rangle_{\hat{\Sigma}_2}\right) \in \xi_1' \times \xi_2'\left(\left(\langle x_1\rangle_{\hat{\Sigma}_1}, \langle x_2\rangle_{\hat{\Sigma}_2}\right), s'\right)$$

which means $s' \in N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(\langle x_1\rangle_{\hat{\Sigma}_1}, \langle x_2\rangle_{\hat{\Sigma}_2})$. Case 3: $\hat{P}_1(s') \neq \epsilon$ and $\hat{P}_2(s') = \epsilon$. This case is similar to Case 2. In either case, we have

$$N_{(G_1\times G_2)/\approx_{\Sigma'}}(\langle (x_1, x_2)\rangle_{\Sigma'}) \subseteq N_{\left(G_1/\approx_{\hat{\Sigma}_1}\right)\times\left(G_2/\approx_{\hat{\Sigma}_2}\right)}\left(\langle x_1\rangle_{\hat{\Sigma}_1}, \langle x_2\rangle_{\hat{\Sigma}_2}\right).$$

Thus, $(G_1/\approx_{\hat{\Sigma}_1}) \times (G_2/\approx_{\hat{\Sigma}_2}) \sqsubseteq (G_1 \times G_2)/\approx_{\Sigma'}$. ■

4) Proof of Theorem 2: The ONLY IF part is obvious. So we only need to show the IF part. Let S be a recognizer of $N(A)$, i.e. $N(S) = N(A)$ and $L(S) = \overline{N(S)}$. Then we have

$$N(G \times S) = N(G) \| N(S) = N(G) \| N(A) \subseteq N(G \times H).$$

Next, we show $B(G \times S) = \varnothing$. Let $G = (X, \Sigma, \xi, x_0, X_m)$, $A = (Z, \Sigma', \delta_i, z_0, Z_m)$ and $S = (Y, \Sigma', \eta, y_0, Y_m)$. Suppose $B(G \times S) \neq \varnothing$. Then there exists $s \in B(G \times S)$ and $(x, y) \in \xi \times \eta((x_0, y_0), s)$ such that for all $s' \in \Sigma^*$

$$\xi \times \eta\left((x, y), s'\right) \cap (X_m \times Y_m) = \varnothing.$$

Let $P : \Sigma^* \to \Sigma'^*$ be the natural projection. Since $P(s) \in L(S) = \overline{N(A)}$, there exists $z \in \delta(z_0, P(s))$. Thus, $(x, z) \in \xi \times \delta((x_0, z_0), s)$. Since $B(G \times A) = \varnothing$, we get that

$$(\exists s' \in \Sigma^*) \xi \times \delta\left((x, z), s'\right) \cap (X_m \times Z_m) \neq \varnothing.$$

Thus, $\xi(x, s') \cap X_m \neq \varnothing$ and $P(ss') \in N(A) = N(S)$. Since S is deterministic, $\eta(y, P(s')) \cap Y_m \neq \varnothing$. Therefore, $\xi \times \eta((x, y), s') \cap (X_m \times Y_m) \neq \varnothing$—contradicting the fact that (x, y) is a blocking state. Thus, $B(G \times S) = \varnothing$.

For each $s \in L(G \times S)$, let $x \in \xi(x_0, s)$ and $y \in \eta(y_0, P(s))$. Since A is state-controllable, for any $z \in \delta(z_0, P(s))$, we have $E_G(x) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_A(z)$. Since $E_S(y) = \cup_{z \in \delta(z_0, P(s))} E_A(z)$, we have

$$E_G(x) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_S(y).$$

Thus, S is state controllable with respect to G and Σ_{uc}.

Next, we show that S is state-observable w.r.t. G and P_o if A is state-observable w.r.t. G and P_o. Suppose it is not true. Then there exist $s, s' \in L(G \times S) \subseteq L(A)$ with $P_o(s) = P_o(s')$, $(x, y) \in \xi \times \eta((x_0, y_0), s)$ and $(x', y') \in \xi \times \eta((x_0, y_0), s')$ such that $E_{G\times S}(x, y) \cap E_G(x') \cap \Sigma' \not\subseteq E_S(y')$. Since S is deterministic, there exists $\sigma \in \Sigma'$ such that

$$s\sigma \in L(G) \wedge s'\sigma \in L(G) \wedge P(s)\sigma \in L(S) \wedge P(s')\sigma \notin L(S).$$

Since $L(S) = \overline{N(A)} = L(A)$, we have that there exist $s, s' \in L(A)$ with $P_o(s) = P_o(s')$ such that

$$s\sigma \in L(G) \wedge s'\sigma \in L(G) \wedge P(s)\sigma \in L(A) \wedge P(s')\sigma \notin L(A).$$

Pick $z \in \delta(z_0, P(s))$ and $z' \in \delta(z_0, P(s'))$, then $(x, z) \in \xi \times \delta((x_0, z_0), s)$ and $(x', z') \in \xi \times \delta((x_0, z_0), s')$. Furthermore, we have that $\sigma \in E_{G\times A}(x, z) \cap E_G(x') \cap \Sigma'$ but $\sigma \notin E_A(z')$, namely $E_{G\times A}(x, z) \cap E_G(x') \cap \Sigma' \not\subseteq E_A(z')$, which contradicts that A is state-observable w.r.t. G and P_o. Thus, S is state-observable w.r.t. G and P_o.

Finally, we show that S is state-normal w.r.t. G and P_o if A is state-normal w.r.t. G and P_o. Let $s \in L(G \times S)$ and $s' \in \overline{P_o^{-1}(P_o(s))} \cap L(G \times S)$. For any $(x, y) \in \xi \times \eta((x_0, y_0), s')$ and $s'' \in \Sigma^*$ with $P_o(s's'') = P_o(s)$, we need to show that

$$\xi(x, s'') \neq \varnothing \Rightarrow \eta\left(y, P(s'')\right) \neq \varnothing.$$

Suppose it is not true. Then there exist $x \in X$ and $s'' \in \Sigma^*$ such that $\xi(x, s'') \neq \varnothing$ but $\eta(y, P(s'')) = \varnothing$. Since S is deterministic, $P(s's'') \notin L(S)$. Since $s \in L(G \times S)$, we get that $P(s) \in L(S) = L(A)$. Let $\hat{s}\sigma \leq P(s's'')$ such that $\hat{s} \in L(A)$ but $\hat{s}\sigma \notin L(A)$. Such $\hat{s}\sigma$ must exists because at least $\epsilon \leq P(s's'')$ and $\epsilon \in \overline{P_o^{-1}(P_o(s))} \cap L(G \times A)$ and $P(s's'') \notin L(A)$. If $P(s') \leq P(\hat{s})$, then let $z \in \delta(z_0, P(s'))$, and we have $(x, z) \in \xi((x_0, z_0), s')$. But $\delta(z, P(s'')) = \varnothing$, which contradicts the fact that A is state-normal with respect to G and P_o. If $P(\hat{s}) \leq P(s')$ and $P(\hat{s}) \neq P(s')$, let $z \in \delta(z_0, P(\hat{s}))$. There exist $x' \in \xi(x_0, \hat{s})$ and $\hat{s}' \in \Sigma^*$ such that $\hat{s}\hat{s}' = s'$ and $x \in \xi(x', \hat{s}')$. Then we have $(x', z) \in \xi \times \delta((x_0, z_0), \hat{s})$, $\xi(x', \hat{s}'s'') \neq \varnothing$ but $\delta(z, \hat{s}'s'') = \varnothing$, which still contradicts the fact that A is state-normal with respect to G and P_o. Thus

$$\xi(x, s'') \neq \varnothing \Rightarrow \eta\left(y, P(s'')\right) \neq \varnothing$$

which means S is state-normal with respect to G and P_o. ■

5) Proof of Prop. 4: Since $N(G \times S_i) \subseteq N(G \times H)$ for $i = 1, 2$, we have

$$\begin{aligned} N(G \times S) &= N(G) \| \left(N(S_1) \cup N(S_2)\right) \\ &= N(G \times S_1) \cup N(G \times S_2) \subseteq N(G \times H). \end{aligned}$$

Next, we show $B(G \times S) = \varnothing$. Let $G = (X, \Sigma, \xi, x_0, X_m)$, $S_i = (Y_i, \Sigma', \eta_i, y_{i,0}, Y_{i,m})$ and $S = (Y, \Sigma', \eta, y_0, Y_m)$. Suppose $B(G \times S) \neq \varnothing$. Then there exist $s \in B(G \times S)$ and $(x, y) \in \xi \times \eta((x_0, y_0), s)$ such that for all $s' \in \Sigma^*$

$$\xi \times \eta\left((x, y), s'\right) \cap (X_m \times Y_m) = \varnothing.$$

Let $P : \Sigma^* \to \Sigma'^*$ be the natural projection. Then $P(s) \in L(S) = \overline{N(S_1) \cup N(S_2)}$. Thus, either $P(s) \in \overline{N(S_1)}$

or $P(s) \in \overline{N(S_2)}$. Without loss of generality, suppose $P(s) \in \overline{N(S_1)}$. Then there exists $y_1 \in \eta_1(y_{1,0}, P(s))$, namely $(x, y_1) \in \xi \times \eta_1((x_0, y_{1,0}), s)$. Since $B(G \times S_1) = \emptyset$, we have

$$(\exists s' \in \Sigma^*)\xi \times \eta_1 \left((x, y_1), s'\right) \cap (X_m \times Y_{1,m}) \neq \emptyset.$$

Thus, $\xi(x, s') \cap X_m \neq \emptyset$ and $P(ss') \in N(S_1) \subseteq N(S)$. Since S is deterministic, we get $\eta(y, P(s')) \cap Y_m \neq \emptyset$. Therefore, $\xi \times \eta((x, y), s') \cap (X_m \times Y_m) \neq \emptyset$—contradicting the fact that (x, y) is a blocking state. Thus, $B(G \times S) = \emptyset$.

For each $s \in L(G \times S)$, let $x \in \xi(x_0, s)$ and $y \in \eta(y_0, P(s))$. Since $P(s) \in L(S) = \overline{N(S_1)} \cup \overline{N(S_2)}$, without loss of generality, suppose $P(s) \in \overline{N(S_1)} = L(S_1)$. Then because S_1 is deterministic, we have

$$E_G(x) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_{S_1}\left(\eta_1\left(y_{1,0}, P(s)\right)\right) \subseteq E_S(y).$$

Thus, S is state controllable with respect to G and Σ_{uc}.

Finally, we show that S is state-normal with respect to G and P_o. Let $s \in L(G \times S)$ and $s' \in \overline{P_o^{-1}(P_o(s))} \cap L(G \times S)$. For any $(x, y) \in \xi \times \eta((x_0, y_0), s')$ and $s'' \in \Sigma^*$ with $P_o(s's'') = P_o(s)$, we need to show that

$$\xi(x, s'') \neq \emptyset \Rightarrow \eta\left(y, P(s'')\right) \neq \emptyset.$$

Suppose it is not true. Then there exist $x \in X$ and $s'' \in \Sigma^*$ such that $\xi(x, s'') \neq \emptyset$ but $\eta(y, P(s'')) = \emptyset$. Since S is deterministic, $P(s's'') \notin L(S)$. Since $s \in L(G \times S)$, we get that $P(s) \in L(S) = L(S_1) \cup L(S_2)$. Without loss of generality, suppose $P(s) \in L(S_1)$. Let $\hat{s}\sigma \leq P(s's'')$ such that $\hat{s} \in L(S_1)$ but $\hat{s}\sigma \notin L(S_1)$. Such $\hat{s}\sigma$ must exists because at least $\epsilon \leq P(s's'')$ and $\epsilon \in \overline{P_o^{-1}(P_o(s))} \cap L(G \times S_1)$ and $P(s's'') \notin L(S_1)$. If $P(s') \leq P(\hat{s})$, then let $y_1 \in \eta_1(y_{1,0}, P(s'))$, and we have $(x, y_1) \in \xi \times \eta_1((x_0, y_{1,0}), s')$. But $\eta_1(y_1, P(s'')) = \emptyset$, which contradicts the fact that S_1 is state-normal with respect to G and P_o. If $P(\hat{s}) \leq P(s')$ and $P(\hat{s}) \neq P(s')$, let $y_1 \in \eta_1(y_{1,0}, P(\hat{s}))$. There exist $x' \in \xi(x_0, \hat{s})$ and $\hat{s}' \in \Sigma^*$ such that $\hat{s}\hat{s}' = s'$ and $x \in \xi(x', \hat{s}')$. Then we have $(x', y_1) \in \xi \times \eta_1((x_0, y_{1,0}), \hat{s})$, $\xi(x', \hat{s}'s'') \neq \emptyset$ but $\eta_1(y_1, \hat{s}'s'') = \emptyset$, which still contradicts the fact that S_1 is state-normal with respect to G and P_o. Thus

$$\xi(x, s'') \neq \emptyset \Rightarrow \eta\left(y, P(s'')\right) \neq \emptyset$$

which means S is state-normal with respect to G and P_o. ■

6) Proof of Lemma 1: Let $G = (X, \Sigma, \xi, x_0, X_m)$ and $S = (Y, \Sigma', \eta, y_0, Y_m)$. We first show the IF part. Suppose it is not true. Then S is state-controllable w.r.t. G and Σ_{uc}, but it is not state-controllable w.r.t. $G/\approx_{\Sigma'}$ and $\Sigma_{uc} \cap \Sigma'$. Thus, for all $s \in L(G \times S)$, $x \in \xi(x_0, s)$ and $y \in \eta(y_0, P(s))$

$$E_G(x) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_S(y) \tag{6}$$

where $P : \Sigma^* \rightarrow \Sigma'^*$ is the natural projection, and there exist $t \in L((G/\approx_{\Sigma'}) \times S)$, $\langle x \rangle \in \xi'(\langle x_0 \rangle, t)$ and $y \in \eta(y_0, t)$ such that

$$E_{G/\approx_{\Sigma'}}(\langle x \rangle) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y) \tag{7}$$

where ξ' is the transition map of $G/\approx_{\Sigma'}$. By the definition of automaton abstraction we have

$$\begin{aligned}&E_{G/\approx_{\Sigma'}}(\langle x \rangle)\\ &\quad = \left\{\sigma \in \Sigma' \,|\, (\exists u \in (\Sigma - \Sigma'^*))\, (\exists x' \in \xi(x, u))\, \sigma \in E_G(x')\right\}.\end{aligned}$$

Thus, $E_{G/\approx_{\Sigma'}}(\langle x \rangle) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y)$ implies that

$$(\exists u \in (\Sigma - \Sigma'^*))\, (\exists x' \in \xi(x, u))\, E_G(x') \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y).$$

From expression (7) we also get that

$$(\exists s \in \Sigma^*) P(s) = t \wedge x \in \xi(x_0, s).$$

Thus, $(x, y) \in \xi \times \eta((x_0, y_0), s)$. Since $u \in (\Sigma - \Sigma')^*$, we have $P(su) = t$, from which we can get that $(x', y) \in \xi \times \eta((x_0, y_0), su)$. Thus, $su \in L(G \times S)$, which means there exist $su \in L(G \times S)$, $x' \in \xi(x_0, su)$ and $y \in \eta(y_0, P(s))$ such that

$$E_G(x') \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y))$$

which contradicts expression (6). Thus, the IF part is true.

Next, we show the ONLY IF part. Suppose it is not true. Then S is state-controllable w.r.t. $G/\approx_{\Sigma'}$ and $\Sigma_{uc} \cap \Sigma'$, but it is not state-controllable w.r.t. G and Σ_{uc}. Thus, for all $t \in L((G/\approx_{\Sigma'}) \times S)$, $\langle x \rangle \in \xi'(\langle x_0 \rangle, t)$ and $y \in \eta(y_0, t)$

$$E_{G/\approx_{\Sigma'}}(\langle x \rangle) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_S(y) \tag{8}$$

and there exist $s \in L(G \times S)$, $x \in \xi(x_0, s)$ and $y \in \eta(y_0, P(s))$ such that

$$E_G(x) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y). \tag{9}$$

Since G is standardized, from expression (9) we get that $\langle x \rangle \in \xi'(\langle x_0 \rangle, P(s))$. Since

$$E_G(x) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y) \Rightarrow E_{G/\approx_{\Sigma'}}(\langle x \rangle) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y)$$

there exist $\langle x \rangle \in \xi'(\langle x_0 \rangle, P(s))$ and $y \in \eta(y_0, P(s))$ such that $E_{G/\approx_{\Sigma'}}(\langle x \rangle) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y)$, which contradicts expression (8). Thus, the ONLY IF part is true. ■

7) Proof of Lemma 2: (1) Let $G = (X, \Sigma, \xi, x_0, X_m)$ and $S = (Y, \Sigma', \eta, y_0, Y_m)$. Suppose S is state observable with respect to $G/\approx_{\Sigma'}$ and P'_o. Thus, for all $s, s' \in L((G/\approx_{\Sigma'}) \times S)$ with $P'_o(s) = P'_o(s')$, and all $(\langle x \rangle, y) \in \xi' \times \eta((\langle x_0 \rangle, y_0), s)$ and $(\langle x' \rangle, y') \in \xi' \times \eta((\langle x_0 \rangle, y_0), s')$

$$E_{(G/\approx_{\Sigma'}) \times S}(\langle x \rangle, y) \cap E_{G/\approx_{\Sigma'}}(\langle x' \rangle) \cap \Sigma' \subseteq E_S(y'). \tag{10}$$

Assume that S is not state-observable w.r.t. G and P_o. Then there are $t, t' \in L(G \times S)$ with $P_o(t) = P_o(t')$, and $(x, y) \in \xi \times \eta((x_0, y_0), t)$ and $(x', y') \in \xi \times \eta((x_0, y_0), t')$ such that

$$E_{G \times S}(x, y) \cap E_G(x') \cap \Sigma' \not\subseteq E_S(y').$$

Since G is standardized, we get $\langle x \rangle \in \xi'(\langle x_0 \rangle, P(t))$ and $\langle x' \rangle \in \xi'(\langle x_0 \rangle, P(t'))$. We also have $y \in \eta(y_0, P(t))$ and

$y' \in \eta(y_0, P(t'))$. Thus, $(\langle x\rangle, y) \in \xi' \times \eta((\langle x_0\rangle, y_0), P(t))$ and $(\langle x'\rangle, y) \in \xi' \times \eta((\langle x_0\rangle, y_0), P(t'))$. We also have that

$$\begin{aligned} &E_{G\times S}(x,y)\cap E_G(x')\cap\Sigma' \not\subseteq E_S(y') \\ &\qquad\Rightarrow E_{(G/\approx_{\Sigma'})\times S}(\langle x\rangle,y)\cap E_{G/\approx_{\Sigma'}}(\langle x'\rangle)\cap\Sigma'\not\subseteq E_S(y'). \end{aligned}$$

Finally, since $P(t)=P(t')$, we have $P'_o(P(t)) = P'_o(P(t'))$. Thus, there exist $s=P(t)$ and $s'=P(t')$ with $P'_o(s)=P'_o(s')$, and there exist

$$\begin{aligned} (\langle x\rangle,y) &\in\xi'\times\eta\left((\langle x_0\rangle,y_0),P(t)\right) \text{ and } \\ (\langle x'\rangle,y) &\in\xi'\times\eta\left((\langle x_0\rangle,y_0),P(t')\right) \end{aligned}$$

such that $E_{(G/\approx_{\Sigma'})\times S}(\langle x\rangle,y)\cap E_{G/\approx_{\Sigma'}}(\langle x'\rangle)\cap\Sigma'\not\subseteq E_S(y')$, which contradicts expression (10). Thus, (1) is true.

(2) Suppose $\Sigma_o\subseteq\Sigma'$. Let S be state observable w.r.t. G and P_o. Thus, for all $t,t'\in L(G\times S)$ with $P_o(t)=P_o(t')$, and all $(x,y)\in\xi\times\eta((x_0,y_0),t)$ and $(x',y')\in\xi\times\eta((x_0,y_0),t')$

$$E_{G\times S}(x,y)\cap E_G(x')\cap\Sigma'\subseteq E_S(y'). \tag{11}$$

Assume that S is not state-observable w.r.t. $G/\approx_{\Sigma'}$ and P'_o. Then there exist $s,s'\in L((G/\approx_{\Sigma'})\times S)$ with $P'_o(s)=P'_o(s')$, and $(\langle x\rangle,y)\in\xi'\times\eta((\langle x_0\rangle,y_0),s)$ and $(\langle x'\rangle,y')\in\xi'\times\eta((\langle x_0\rangle,y_0),s')$

$$E_{(G/\approx_{\Sigma'})\times S}(\langle x\rangle,y)\cap E_{G/\approx_{\Sigma'}}(\langle x'\rangle)\cap\Sigma'\not\subseteq E_S(y'). \tag{12}$$

Clearly, there exist $t,t'\in\Sigma^*$ with $P(t)=s$ and $P(t')=s'$ such that $(x,y)\in\xi\times\eta((x_0,y_0),t)$ and $(x',y')\in\xi\times\eta((x_0,y_0),t')$. We also have that

$$\begin{aligned} E_{(G/\approx_{\Sigma'})\times S}(\langle x\rangle,y) = \{\sigma\in\Sigma' |\, &(\exists u\in(\Sigma-\Sigma')^*) \\ &(\exists\hat{x}\in\xi(x,u))\,\sigma\in E_{G\times S}(\hat{x},y)\} \end{aligned}$$

and

$$\begin{aligned} E_{G/\approx_{\Sigma'}}(\langle x'\rangle) = \left\{\sigma\in\Sigma' |\, \left(\exists u'\in(\Sigma-\Sigma')^*\right)\right. \\ \left.(\exists\hat{x}'\in\xi(x',u))\,\sigma\in E_G(\hat{x}')\right\}. \end{aligned}$$

Thus, from expression (12), there exist $u,u'\in(\Sigma-\Sigma')^*$ and $\hat{x}\in\xi(x,u)$ and $\hat{x}'\in\xi(x',u')$ such that

$$E_{G\times S}(\hat{x},y)\cap E_G(\hat{x}')\cap\Sigma'\not\subseteq E_S(y').$$

Since $P'_o(P(t)) = P'_o(P(t'))$ and $\Sigma_o\subseteq\Sigma'$, we have $P_o(tu)=P_o(t'u')$. Thus, there exist $tu,t'u'\in L(G\times S)$ with $P_o(tu)=P_o(t'u')$, $(\hat{x},y)\in\xi\times\eta((x_0,y_0),tu)$ and $(\hat{x}',y')\in\xi\times\eta((x_0,y_0),t'u')$ such that

$$E_{G\times S}(\hat{x},y)\cap E_G(\hat{x}')\cap\Sigma'\not\subseteq E_S(y')$$

which contradicts expression (11). Thus, (2) is true. ■

8) Proof of Lemma 3: (1) Let $G=(X,\Sigma,\xi,x_0,X_m)$ and $S=(Y,\Sigma',\eta,y_0,Y_m)$. Suppose S be state normal w.r.t. $G/\approx_{\Sigma'}$ and P'_o. Then for any $s\in L((G/\approx_{\Sigma'})\times S)$, $s'\in\overline{P_o'^{-1}(P'_o(s))}\cap L((G/\approx_{\Sigma'})\times S)$, we have, for each $(\langle x\rangle,y)\in\xi'\times\eta((\langle x_0\rangle,y_0),s')$ and $s''\in\Sigma'^*$

$$P'_o(s's'')=P'_o(s)\Rightarrow\left[\xi'\left(\langle x\rangle,s''\right)\neq\emptyset\Rightarrow\eta(y,s'')\neq\emptyset\right]. \tag{13}$$

Suppose S is not state-normal w.r.t. G and P_o. Then there exist $t\in L(G\times S)$ and $t'\in\overline{P_o^{-1}(P_o(t))}\cap L(G\times S)$ such that there exist $(x,y)\in\xi\times\eta((x_0,y_0),t')$ and $t''\in\Sigma^*$

$$P_o(t't'')=P_o(t)\wedge\xi(x,t'')\neq\emptyset\wedge\eta\left(y,P(t'')\right)=\emptyset.$$

Let $P''_o:\Sigma_o^*\rightarrow(\Sigma_o\cap\Sigma')^*$ be the naturel projection. Since G is standardized, we have $P(t)\in L((G/\approx_{\Sigma'})\times S)$. From $(x,y)\in\xi\times\eta((x_0,y_0),t')$ we can derive that $(\langle x\rangle,y)\in\xi'\times\eta((\langle x_0\rangle,y_0),P(t'))$. Since $t'\in\overline{P_o^{-1}(P_o(t))}\cap L(G\times S)$, we have $P_o(t')\leq P_o(t)$ and $P(t)\in L((G/\approx_{\Sigma'})\times S)$, which means $P'_o(P(t')) = P''_o(P_o(t'))\leq P''_o(P_o(t)) = P'_o(P(t))$. Thus, $P(t')\in\overline{P_o'^{-1}(P'_o(s))}\cap L((G/\approx_{\Sigma'})\times S)$. Since $P_o(t't'')=P_o(t)$, we have

$$P'_o\left(P(t't'')\right)=P''_o\left(P_o\left(t't''\right)\right)=P''_o\left(P_o(t)\right)=P'_o\left(P(t)\right).$$

Since $\xi(x,t'')\neq\emptyset$, if $P(t'')=\epsilon$ we have $\xi'(\langle x\rangle,\epsilon)\neq\emptyset$; if $P(t'')\neq\epsilon$ we have $\xi(\langle x\rangle,P(t''))\neq\emptyset$. Thus, there exist $s=P(t)\in L((G/\approx_{\Sigma'})\times S)$, $s'=P(t')\in\overline{P_o'^{-1}(P'_o(s))}\cap L((G/\approx_{\Sigma'})\times S)$ such that there exist $(\langle x\rangle,y)\in\xi'\times\eta((\langle x_0\rangle,y_0),s')$ and $s''=p(t'')\in\Sigma'^*$

$$P'_o(s's'')=P'_o(s)\wedge\xi'\left(\langle x\rangle,s''\right)\neq\emptyset\wedge\eta(y,s'')=\emptyset$$

which contradicts expression (13). Thus, (1) is true.

(2) Let S be state-normal w.r.t. G and P_o. Then for any $t\in L(G\times S)$, $t'\in\overline{P_o^{-1}(P_o(t))}\cap L(G\times S)$, we have, for each $(x,y)\in\xi\times\eta((x_0,y_0),t')$ and $t''\in\Sigma^*$

$$P_o(t't'')=P_o(t)\Rightarrow\left[\xi(x,t'')\neq\emptyset\Rightarrow\eta\left(y,P(t'')\right)\neq\emptyset\right]. \tag{14}$$

Suppose S is not state-normal w.r.t. $G/\approx_{\Sigma'}$ and P'_o. Then there exist $s\in L((G/\approx_{\Sigma'})\times S)$, $s'\in\overline{P_o'^{-1}(P'_o(s))}\cap L((G/\approx_{\Sigma'})\times S)$, $(\langle x\rangle,y)\in\xi'\times\eta((\langle x_0\rangle,y_0),s')$ and $s''\in\Sigma'^*$ such that

$$P'_o(s's'')=P'_o(s)\wedge\xi'\left(\langle x\rangle,s''\right)\neq\emptyset\wedge\eta(y,s'')=\emptyset.$$

Clearly, there exists $t\in L(G\times S)$ such that $P(t)=s$. There also exists $t'\in\Sigma^*$ such that $P(t')=s'$ and $(x,y)\in\xi\times\eta((x_0,y_0),t')$. Thus, $t'\in L(G\times S)$. Since $s'\in\overline{P_o'^{-1}(P'_o(s))}\cap L((G/\approx_{\Sigma'})\times S)$, we have $P'_o(s')\leq P'_o(s)$. Since $\Sigma_o\subseteq\Sigma'$, we have $P_o(t')=P'_o(P(t'))=P'_o(s')\leq P'_o(s)=P'_o(P(t))=P_o(t)$. Thus, $t'\in\overline{P_o^{-1}(P_o(t))}\cap L(G\times S)$. Since $\xi'(\langle x\rangle,s'')\neq\emptyset$, there exists $t''\in\Sigma^*$ such that $\xi(x,t'')\neq\emptyset$. From $P'_o(s's'')=P'_o(s)$ we have $P_o(t't'')=P'_o(P(t't''))=P'_o(s's'')=P'_o(s)=P'_o(P(t))=P_o(t)$. Thus, there exist $t\in L(G\times S)$, $t'\in\overline{P_o^{-1}(P_o(t))}\cap L(G\times S)$, $(x,y)\in\xi\times\eta((x_0,y_0),t')$ and $t''\in\Sigma^*$ such that

$$P_o(t't'')=P_o(t)\wedge\xi(x,t'')\neq\emptyset\wedge\eta\left(y,P(t'')\right)=\emptyset$$

which contradicts expression (14). Thus, (2) is true. ■

References

[1] M. Fabian and B. Lennartson, "On non-deterministic supervisory control," in *Proc. 35th IEEE Conf. Decision Control (CDC'96)*, 1996, pp. 2213–2218.

[2] L. Feng and W. M. Wonham, "Computationally efficient supervisor design: Abstraction and modularity," in *Proc. 8th Int. Workshop Discrete Event Syst. (WODES'06)*, 2006, pp. 3–8.

[3] J. C. Fernandez, "An implementation of an efficient algorithm for bisimulation equivalence," *Sci. Comp. Programming*, vol. 13, no. 2–3, pp. 219–236, 1990.

[4] H. Flordal and R. Malik, "Modular nonblocking verification using conflict equivalence," in *Proc. 8th Int. Workshop Discrete Event Syst. (WODES'06)*, 2006, pp. 100–106.

[5] H. Flordal, R. Malik, M. Fabian, and K. Akesson, "Compositional synthesis of maximally permissive supervisors using supervisor equivalence," *Discrete Event Dyn. Syst.*, vol. 17, no. 4, pp. 475–504, 2007.

[6] M. Heymann and F. Lin, "Discrete event control of nondeterministic systems," *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 3–17, Jan. 1998.

[7] R. C. Hill, D. M. Tilbury, and S. Lafortune, "Modular supervisory control with equivalence-based conflict resolution," in *Proc. 27th Amer. Control Conf. (ACC'08)*, 2008, pp. 491–498.

[8] R. Kumar and M. A. Shayman, "Centralized and decentralized supervisory control of nondeterministic systems under partial observation," *SIAM J. Control Optim.*, vol. 35, no. 2, pp. 363–383, 1997.

[9] R. J. Leduc and P. Dai, "Synthesis method for hierarchical interface-based supervisory control," in *Proc. 26th Amer. Control Conf. (ACC'07)*, 2007, pp. 4260–4267.

[10] R. J. Leduc, M. Lawford, and W. M. Wonham, "Hierarchical interface-based supervisory control-part II: Parallel case," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 1336–1348, Sep. 2005.

[11] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inform. Sci.*, vol. 44, no. 2, pp. 173–198, 1988.

[12] C. Ma and W. M. Wonham, "Nonblocking supervisory control of state tree structures," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 782–793, May 2006.

[13] R. Malik and H. Flordal, "Yet another approach to compositional synthesis of discrete event systems," in *Proc. 9th Int. Workshop Discrete Event Systems (WODES'08)*, 2008, pp. 16–21.

[14] R. Milner, "Operational and algebraic semantics of concurrent processes," in *Handbook of Theoretical Computer Science (vol. B): Formal Models and Semantics*. Cambridge, MA: MIT Press, 1990, pp. 1201–1242.

[15] A. Overkamp, "Supervisory control using failure semantics and partial specifications," *IEEE Trans. Autom. Control*, vol. 42, no. 4, pp. 498–510, Apr. 1997.

[16] P. N. Pena, J. E. R. Cury, and S. Lafortune, "Testing modularity of local supervisors: An approach based on abstractions," in *Proc. 8th Int. Workshop Discrete Event Syst. (WODES'06)*, 2006, pp. 107–112.

[17] M. H. de Queiroz and J. E. R. Cury, "Modular supervisory control of composed systems," in *Proc. 19th Amer. Control Conf. (ACC'00)*, 2000, pp. 4051–4055.

[18] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event systems," *SIAM J. Control Optim.* , vol. 25, no. 1, pp. 206–230, 1987.

[19] K. Schmidt, H. Marchand, and B. Gaudin, "Modular and decentralized supervisory control of concurrent discrete event systems using reduced system models," in *Proc. 8th Int. Workshop Discrete Event Syst. (WODES'06)*, 2006, pp. 149–154.

[20] R. Su, J. H. van Schuppen, and J. E. Rooda, "Synthesize nonblocking distributed supervisors with coordinators," in *Proc. 17th Mediterranean Conf. Control Autom. (MED'09)*, 2009, pp. 1108–1113.

[21] R. Su, J. H. van Schuppen, and J. E. Rooda, "Aggregative synthesis of distributed supervisors based on automaton abstraction," *IEEE Trans. Autom. Control*, vol. 55, no. 7, pp. 1627–1640, Jul. 2010.

[22] R. Su and J. G. Thistle, "A distributed supervisor synthesis approach based on weak bisimulation," in *Proc. 8th Int. Workshop Discrete Event Syst. (WODES06)*, 2006, pp. 64–69.

[23] K. C. Wong and W. M. Wonham, "Hierarchical control of discrete-event systems," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 6, no. 3, pp. 241–273, 1996.

[24] K. C. Wong and W. M. Wonham, "On the computation of observers in discrete-event systems," *Discrete Event Dyn. Syst.*, vol. 14, no. 1, pp. 55–107, 2004.

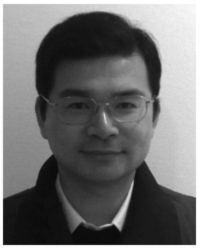
[25] W. M. Wonham, Supervisory Control of Discrete-Event Systems, Systems Control Group Dept. ECE, Univ. Toronto, Toronto, ON, Canada, Tech. Rep., Jul. 2007 [Online]. Available: www.control.utoronto.ca/DES

[26] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM J. Control Optim.*, vol. 25, no. 3, pp. 637–659, 1987.

[27] W. M. Wonham and P. J. Ramadge, "Modular supervisory control of discrete event systems," *Maths. Control, Signals Syst.*, vol. 1, no. 1, pp. 13–30, 1988.

[28] C. Zhou and R. Kumar, "A small model theorem for bisimilarity control under partial observation," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 93–97, Jan. 2007.

[29] C. Zhou, R. Kumar, and S. Jiang, "Control of nondeterministic discrete event systems for bisimulation equivalence," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 754–765, May 2006.



Rong Su received the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2000 and 2004, respectively.

He is currently affiliated with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include modeling, fault diagnosis and supervisory control of discrete-event dynamic systems.

Dr. Su has been a member of IFAC technical committee on discrete event and hybrid systems (TC 1.3) since 2005.

Jan H. van Schuppen (M'73) is affiliated with Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands, and as Full Professor with the Department of Mathematics, Delft University of Technology (part time), Delft, The Netherlands. He is Editor-in-Chief of *Mathematics of Control, Signals, and Systems* and was Department Editor of *Discrete Event Dynamic Systems*. His research interests include control of hybrid systems and of discrete-event systems, stochastic control, realization, and system identification. In applied research his interests include engineering problems of control of motorway traffic, of communication networks, and control and system theory for the life sciences.

Dr. van Schuppen was Associate Editor-at-Large of the IEEE TRANSACTIONS AUTOMATIC CONTROL.

Jacobus E. Rooda (M'90) received the M.Sc. degree from Wageningen University of Agriculture Engineering, Wageningen, The Netherlands and the Ph.D. degree from Twente University, Enschede, The Netherlands.

Since 1985, he has been a Professor of (manufacturing) systems engineering with the Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands. His research fields of interest are modeling and analysis of manufacturing systems. His interest is especially in control of (high-tech) manufacturing lines and in supervisory control of high-tech (manufacturing) machines.