Model abstraction of nondeterministic finite state automata in supervisor synthesis

Systems Engineering Group
Department of Mechanical Engineering
Eindhoven University of Technology
PO Box 513
5600 MB Eindhoven
The Netherlands
http://se.wtb.tue.nl/

SE Report: Nr. 2008-03

Model Abstraction of Nondeterministic Finite State Automata in Supervisor Synthesis

Rong Su, Jan H. van Schuppen and Jacobus E. Rooda[1] [2]

ISSN: 1872-1567

SE Report: Nr. 2008-03
Eindhoven, June 2008
SE Reports are available via http://se.wtb.tue.nl/sereports

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 to apply it to supervisor synthesis, in which plant models are nondeterministic but specifications and supervisors are deterministic. We show that a nonblocking state-controllable state-observable (or state-normal) supervisor for an abstraction of a plant under a specification is guaranteed to be a nonblocking state-controllable state-observable (or state-normal) supervisor of the original plant under the same specification. The reverse statement is also true, if every observable event is contained in the abstraction and the plant model is marking aware with respect to the alphabet of the abstraction.

1 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 [1, 2]. Since then there has been a large volume of literature under the same paradigm. In the RW paradigm the main difficulty of supervisor synthesis for a large system is to achieve the nonblockingness. The reason is that 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 [3] the authors propose the concept of *modularity*, which is then extended to the concept of *local modularity* in [4]. 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 [7, 8], 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 system size reaches 10^{21} in the AIP example [8]. 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 [10] the authors present an interesting approach, which is aimed to synthesize a state-feedback supervisor. By introducing the concept of *state tree structures*, the authors propose to represent product states in binary decision diagrams (BDDs), upon which the power of symbolic computation (as manifested by the manipulation of BDDs) is fully utilized. It has been shown in [10] that a system with 10^{24} states can be accomodated. Nevertheless, this approach is essentially a centralized approach. No matter how efficient the symbolic computational technique is, such efficiency can never completely overcome the complexity issue in an industrial system that usually consists of hundreds or thousands of components. Besides, the proposed approach does not deal with partial observation.

In this paper we will discuss how to synthesize a supervisor by using an appropriate abstraction of a system. 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 [11] abstraction is used in the modular and hierarchical supervisor synthesis; it is also used in [12] for testing the nonblocking property, and in [13] for decentralized control. Nevertheless, their approaches are language-based, which use natural projections. To make sure nonblocking information will not be incorrectly masked out by abstraction, natural projections have to possess the *observer* property [6], which may not always hold by a natural projection. Although a natural projection can always be modified to become an observer (with respect to a specific language) [14], 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, thus different from those language-based abstraction techniques. There have been several research works on automaton abstraction, e.g. [15] [24] [25] [26] [27]. [15] aims to achieve weak bisimilarity between an automaton and its abstraction. [24] [25] [26] [27] first use silence events to replace internal events, then apply rewriting rules to ensure that appropriate equivalence relations, e.g. conflict equivalence in [24] [27], supervision equivalence in [25] and synthesis equivalence in [26], hold between automata before and after rewriting. Our approach does not use silence events, and its primary goal 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 G is marking aware, then it is automatically true that S is nonconflicting with G implies S is nonconflicting with the abstraction - at this point, our approach is close to achieving conflict equivalence, but with a procedure much simpler than those rewriting rules and no silence events are needed. Our second contribution is to utilize the automaton-based abstraction in supervisor synthesis and provide conditions under which the existence of a nonblocking supervisor for an abstraction of a plant guarantees the existence of a nonblocking supervisor for the original plant, and vice versa. Since abstraction usually results in nondeterministic automata, we consider supervisor synthesis with a nondeterministic plant model but with a deterministic specification. The supervisor is required to be deterministic as well. There have been a large volume of work on supervisor synthesis for nondeterministic systems, but most do not use abstraction, e.g. [23] [18] [5] [19] [20] [21]. Although [15] [24] [25] [26] [27] utilize abstraction in synthesis, their abstraction techniques are different from ours. Our third contribution is to introduce the concept of *state normality*, which allows computing a supremal nonblocking supervisor for a nondeterministic system, which is important for synthesis, especially considering that nondeterministic systems always behave like systems under partial observation.

This paper is organized as follows. In Section II we introduce an abstraction technique over nondeterministic automata. Then in Section III we describe how to apply it to synthesize a deterministic supervisor for a nondeterministic plant model under a deterministic specification. After an illustrative example in Section IV, conclusions are stated in Section V. Long proofs are presented in the Appendix.

2 Automaton Abstraction and Relevant Properties

In this section we follow the notations used in [16]. We first briefly review concepts of languages, natural projection, synchronous product and automaton product, then introduce the concept of automaton abstraction. After that, we present properties of abstraction which will be used in supervisor synthesis.

2.1 Concepts of Languages, Automaton Product and Abstraction

Let Σ be a finite alphabet, and Σ^* 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 $L, L' \subseteq \Sigma^*$, $LL' := \{ss' \in \Sigma^* | s \in L \wedge s' \in L'\}$.

Let $\Sigma' \subseteq \Sigma$. A map $P : \Sigma^* \to \Sigma'^*$ is 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 $L \subseteq \Sigma^*$, $P(L) := \{P(s) \subseteq \Sigma'^* | s \in L\}$. For any two languages L, $L' \subseteq \Sigma^*$, we can show that $P(LL') = P(L)P(L')$. 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:

$$L_1 || L_2 := P_1^{-1}(L_1) \cap P_2^{-1}(L_2) = \{s \in (\Sigma_1 \cup \Sigma_2)^* | P_1(s) \in L_1 \wedge P_2(s) \in L_2\}$$

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. It has been shown [16] that $||$ is commutative and associative. Next, we introduce automaton product and abstraction.

Given a nondeterministic finite-state automaton $G = (X, \Sigma, \xi, x_0, X_m)$, 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 for the marker state set. As usual, we extend the domain of ξ from $X \times \Sigma$ to $X \times \Sigma^*$, where $\xi(x, s\sigma) := \{x' \in X | (\exists x'' \in \xi(x, s))\, x' \in \xi(x'', \sigma)\}$. We bring in a new event symbol τ. An automaton $G = (X, \Sigma \cup \{\tau\}, \xi, x_0, X_m)$ is *standardized* if

$$x_0 \notin X_m \wedge (\forall x \in X)\, [\xi(x, \tau) \neq \varnothing \iff x = x_0] \wedge (\forall x \in X - \{x_0\})(\forall \sigma \in \Sigma)\, x_0 \notin \xi(x, \sigma)$$

A standardized automaton is nothing but an automaton, whose initial state x_0 only has outgoing transitions with the same label τ, and no incoming transitions. For an ordinary automaton $G = (X, \Sigma, \xi, x_0, X_m)$ we can convert it into a standardized automaton by simply: (1) extend the alphabet to $\Sigma \cup \{\tau\}$; (2) add a new state x_0'; (3) define a new transition map ξ' such that $\xi'(x_0', \tau) = \{x_0\}$ and for any $(x, \sigma) \in X \times \Sigma$ we have $\xi'(x, \sigma) = \xi(x, \sigma)$. The resultant automaton $G' = (X \cup \{x_0'\}, \Sigma \cup \{\tau\}, \xi', x_0', X_m)$ is a standardized automaton. From now on, unless specified explicitly, we assume that each alphabet Σ contains τ, and $\phi(\Sigma)$ is the collection of all standardized finite state automata, whose alphabet is Σ. The role of τ will be explained shortly. We now introduce automaton product.

Given two nondeterministic automata $G_i = (X_i, \Sigma_i, \xi_i, x_{0,i}, X_{m,i}) \in \phi(\Sigma_i)$ $(i = 1, 2)$, the *product* of G_1 and G_2, written as $G_1 \times G_2$, is an automaton in $\phi(\Sigma_1 \cup \Sigma_2)$ such that

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

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,

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

Clearly, $\times$ is commutative and associative. By a slight abuse of notations, from now on we use $G_1 \times G_2$ to denote its reachability part. $\xi_1 \times \xi_2$ is extended to $X_1 \times X_2 \times (\Sigma_1 \cup \Sigma_2)^* \to 2^{X_1 \times X_2}$.

We can easily see that product of two standardized automata is still a standardized automaton. Next, we introduce automaton abstraction, which requires the following concept

of marking weak bisimilarity.

Definition 2.1. 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,

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

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 bisimulation relation is the same as weak bisimulation relation described in [17], except for the special treatment on marker states. We now introduce abstraction.

Definition 2.2. 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 $G/\approx_{\Sigma',G} := (Y, \Sigma', \eta, y_0, Y_m)$ where

1. $Y := X/\approx_{\Sigma',G} := \{< x > := \{x' \in X | (x, x') \in \approx_{\Sigma',G}\} | x \in X\}$
2. $y_0 :=< x_0 >$
3. $Y_m := \{y \in Y | y \cap X_m \neq \varnothing\}$
4. $\eta : Y \times \Sigma' \to 2^Y$, where for any $(y, \sigma) \in Y \times \Sigma'$,
$$\eta(y, \sigma) := \{y' \in Y | (\exists x \in y)(\exists u, u' \in (\Sigma - \Sigma')^*)\, \xi(x, u\sigma u') \cap y' \neq \varnothing\}$$

□

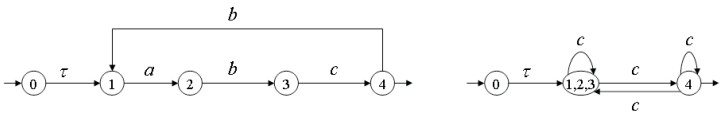
We can easily check that, if G is standardized, then $G/\approx_{\Sigma',G}$ is also standardized. The time complexity of computing $G/\approx_{\Sigma',G}$ is mainly resulted from computing $X/\approx_{\Sigma',G}$, which can be estimated as follows. We first define a new automaton $G'' = (X, \Sigma', \xi'', x_0, X_m)$, where for any $x, x' \in X$ and $\sigma \in \Sigma$, $x' \in \xi''(x, \sigma)$ if there exist $u, u' \in (\Sigma - \Sigma')^*$ such that $x' \in \xi(x, u\sigma u')$. Then we compute $X/\approx_{\Sigma',G''}$, and we can show that the result 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 is the number of transitions in G. Based on a result shown in [22], the time complexity of computing $X/\approx_{\Sigma',G''}$ 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. 2.1. If we consider this extra condition, then the overall complexity is $O(n(n-1) + mn^2 \log n)$, because we need to check at most $n(n-1)$ pairs of states.

From now on, when G is clear from the context, we simply use $\approx_{\Sigma'}$ to denote $\approx_{\Sigma',G}$, and use $< x >_{\Sigma'}$ for an element of $X/\approx_{\Sigma',G}$. If Σ' is also clear from the context, then we simply use $< x >$ for $< x >_{\Sigma'}$.

As an illustration, suppose a standardized automaton $G \in \phi(\Sigma)$ is depicted in Figure 1, where the alphabet $\Sigma = \{\tau, a, b, c\}$. We take $\Sigma' = \{\tau, c\}$. Then we have

$$X/\approx_{\Sigma'} = \{< 0 >= \{0\}, < 1 >= \{1, 2, 3\}, < 4 >= \{4\}\}$$

The abstraction $G/\approx_{\Sigma'}$ is depicted in Figure 1. Next, we present properties of automaton abstraction.



G $\qquad\qquad$ $G/\approx_{\Sigma'}$

Figure 1: Example 1: A Standardized Automaton G and Automaton Abstraction $G/\approx_{\Sigma'}$

2.2 Properties of Automaton Abstraction

We first define a map $B : \phi(\Sigma) \to 2^{\Sigma^*}$ with

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

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 $B(G) = \varnothing$. Similarly, define another map $N : \phi(\Sigma) \to 2^{\Sigma^*}$ with

$$(\forall G \in \phi(\Sigma))\, N(G) := \{s \in \Sigma^* | \xi(x_0, s) \cap X_m \neq \varnothing\}$$

We call $N(G)$ the *nonblocking set* of G, which is simply the language recognized by G. It is possible that $B(G) \cap \overline{N(G)} \neq \varnothing$, due to nondeterminism.

Definition 2.3. 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 $\{\sigma \in \Sigma | (\exists x \in X - X_m)(\exists x' \in X_m)\, x' \in \xi(x, \sigma)\} \subseteq \Sigma'$. We have the following result, which will be extensively used in the rest of this paper.

Proposition 2.4. Given $G \in \phi(\Sigma)$, let $\Sigma' \subseteq \Sigma$, 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 $P(B(G)) = B(G/\approx_{\Sigma'})$. □

As an illustration of Prop. 2.4, Figure 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 $P(B(G)) = \{\tau b\}$

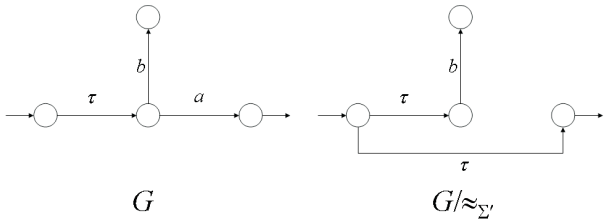


Figure 2: Example 2: G and $G/\approx_{\Sigma'}$

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. 2.4.

We now explain why we introduce the event τ and standardized automata. If we don not use it, then we may not always have $P(B(G)) \subseteq B(G/\approx_{\Sigma'})$ and $P(N(G)) \subseteq N(G/\approx_{\Sigma'})$ (let alone $P(N(G)) = N(G/\approx_{\Sigma'})$), which may cause supervisor synthesis based on a reduced model $G/\approx_{\Sigma'}$ to fail, in the sense that a nonblocking supervisor (whose precise definition will be given later) of $G/\approx_{\Sigma'}$ may not be a nonblocking one for G. This will be clear when we introduce supervisor synthesis.

Next, we want to answer the question whether $G \times S$ for some automaton S is nonblocking if and only if $(G/\approx_{\Sigma'}) \times S$ is nonblocking. Given an automaton $G = (X, \Sigma, \xi, x_0, X_m)$, for each $x \in X$, let

$$N_G(x) := \{s \in \Sigma^* | \xi(x, s) \cap X_m \neq \varnothing\}$$

We now introduce the following concept, which will be extensively used in this paper.

Definition 2.5. 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 any $s \in \overline{N(G_1)}$,

$$(\forall x_1 \in \xi_1(x_{1,0}, s))(\exists x_2 \in \xi_2(x_{2,0}, s))\, N_{G_2}(x_2) \subseteq N_{G_1}(x_1) \wedge [x_1 \in X_{1,m} \iff x_2 \in X_{2,m}]$$

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

Def. 2.5 says that, if G_1 is nonblocking preserving with respect to G_2 then their individual nonblocking parts are equal, but G_2's blocking behavior may be larger. If blocking behaviors are also equal, then G_1 and G_2 are nonblocking equivalent. We now present a few results.

Proposition 2.6. $(\forall G_1, G_2 \in \phi(\Sigma))(\forall G_3 \in \phi(\Sigma'))\, G_1 \sqsubseteq G_2 \Rightarrow G_1 \times G_3 \sqsubseteq G_2 \times G_3$ □

Corollary 2.7. $(\forall G_1, G_2 \in \phi(\Sigma))(\forall G_3 \in \phi(\Sigma'))\, G_1 \cong G_2 \Rightarrow G_1 \times G_3 \cong G_2 \times G_3$. □

Proof: Since $G_1 \cong G_2$, by Def. 2.5 we have $G_1 \sqsubseteq G_2$ and $G_2 \sqsubseteq G_1$. Then by Prop. 2.6 we get $G_1 \times G_3 \sqsubseteq G_2 \times G_3$ and $G_2 \times G_3 \sqsubseteq G_1 \times G_3$, namely $G_1 \times G_3 \cong G_2 \times G_3$. ■

Prop. 2.6 and Cor. 2.7 say nonblocking preserving and equivalence are invariant under automaton product.

Proposition 2.8. 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 G_i $(i = 1, 2)$ is marking aware with respect to $\Sigma_i \cap \Sigma'$, then
$$(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$$

□

Proposition 2.8 is about the distribution of automaton abstraction over automaton product. As an illustration we present a simple example. Suppose we have $\Sigma_1 = \{\tau, a, \mu\}$ and $\Sigma_2 = \{\tau, b, c, \mu\}$. Let $G_1 \in \phi(\Sigma_1)$ and $G_2 \in \phi(\Sigma_2)$ be shown in Figure 3. Suppose

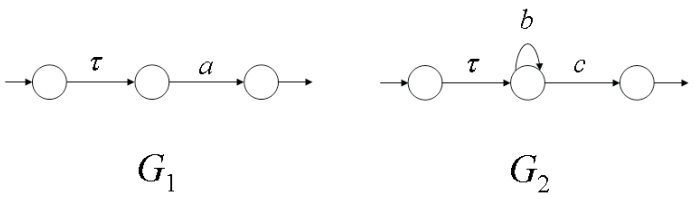


Figure 3: Example 3: G_1 and G_2

we pick $\Sigma' = \{\tau, a, b, \mu\} \supseteq \Sigma_1 \cap \Sigma_2$. The results of $G_1 \times G_2$ and $(G_1 \times G_2)/\approx_{\Sigma'}$ are depicted in Figure 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

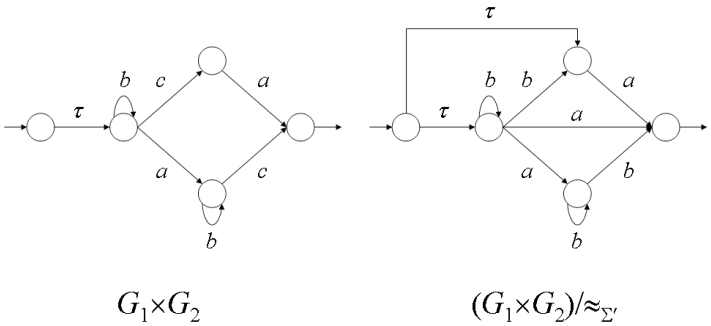


Figure 4: Example 3: $G_1 \times G_2$ and $(G_1 \times G_2)/\approx_{\Sigma'}$

Figure 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 we can check that $(G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'}) \sqsubseteq (G_1 \times G_2)/\approx_{\Sigma'}$ does not hold. If we set $\Sigma' = \{\tau, a, c\}$, then G_i $(i = 1, 2)$ is marking aware with respect to $\Sigma_i \cap \Sigma'$. We can check that, indeed $(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$, as predicted by Prop. 2.8.

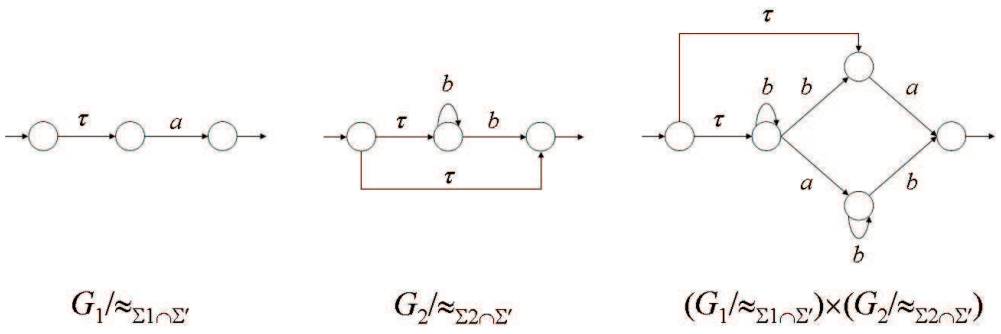


Figure 5: Example 3: $G_1/\approx_{\Sigma_1\cap\Sigma'}$, $G_2/\approx_{\Sigma_2\cap\Sigma'}$ and $(G_1/\approx_{\Sigma_1\cap\Sigma'})\times(G_2/\approx_{\Sigma_2\cap\Sigma'})$

Theorem 2.9. Given Σ and $\Sigma' \subseteq \Sigma$, 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. G is marking aware w.r.t. $\Sigma' \Rightarrow [B((G/\approx_{\Sigma'})\times S) = \varnothing \iff B(G\times S) = \varnothing]$ □

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

$$\begin{aligned}
& B((G/\approx_{\Sigma'})\times S) = \varnothing \\
\iff\ & B((G/\approx_{\Sigma'})\times (S/\approx_{\Sigma'})) = \varnothing \text{ because } S/\approx_{\Sigma'} \cong S \text{ and by Cor. 2.7} \\
\Rightarrow\ & B((G\times S)/\approx_{\Sigma'}) = \varnothing \text{ because by Prop. 2.8, } (G\times S)/\approx_{\Sigma'} \sqsubseteq (G/\approx_{\Sigma'})\times(S/\approx_{\Sigma'}) \\
\Rightarrow\ & P(B(G\times S)) = \varnothing \text{ by Prop. 2.4} \\
\iff\ & B(G\times S) = \varnothing
\end{aligned}$$

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. 2.8, we have

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

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. 2.4 we get that

$$P(B(G\times S)) = B((G\times S)/\approx_{\Sigma'}) \tag{2}$$

Thus we have

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

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

What Theorem 2.9 says can be interpreted informally 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. But to make the opposite implication true, we need to impose the marking awareness condition. We will see how to use this result in supervisor synthesis shortly. But before that, we need to address some computational issue: if G is very large, e.g. $G = G_1 \times \cdots \times G_n$ for some very large number $n \in \mathbb{N}$, where $G_i \in \phi(\Sigma_i)$ for $i = 1, 2, \cdots, n$, how to compute

$G/\approx_{\Sigma'}$? To overcome this difficulty, we propose the following algorithm.

Suppose $I = \{1, 2, \cdots, n\}$ for some $n \in \mathbb{N}$. For any $J \subseteq I$, let $\Sigma_J := \cup_{j\in J}\Sigma_j$. Let $\Sigma' \subseteq \cup_{i\in I}\Sigma_i$.

Sequential Abstraction over Product: (SAP)
(1) Input of SAP: a collection of automata $\{G_i | i \in I\}$.
(2) For $k = 1, 2, \cdots, n$, we perform the following computation.

- Set $J_k := \{1, 2, \cdots, k\}$, $T_k := \Sigma_{J_k} \cap (\Sigma_{I-J_k} \cup \Sigma')$.
- If $k = 1$ then $W_1 := G_1/\approx_{T_1}$
- If $k > 1$ then $W_k := (W_{k-1} \times G_k)/\approx_{T_k}$

(3) Output of SAP: W_n. □

To show that SAP fulfils our expectation, we first need some preparations.

Proposition 2.10. $(\forall\Sigma' \subseteq \Sigma)(\forall G_1, G_2 \in \phi(\Sigma))\, G_1 \sqsubseteq G_2 \Rightarrow G_1/\approx_{\Sigma'} \sqsubseteq G_2/\approx_{\Sigma'}$. □

Corollary 2.11. $(\forall\Sigma' \subseteq \Sigma)(\forall G_1, G_2 \in \phi(\Sigma))\, G_1 \cong G_2 \Rightarrow G_1/\approx_{\Sigma'} \cong G_2/\approx_{\Sigma'}$. □

Proof: Use Prop. 2.10 and Def. 2.5, the corollary follows. ■

Prop. 2.10 and Cor. 2.11 are about nonblocking preserving and equivalence being invariant under abstraction.

Proposition 2.12. $(\forall\Sigma'' \subseteq \Sigma' \subseteq \Sigma)(\forall G \in \phi(\Sigma))\, G/\approx_{\Sigma''} \cong (G/\approx_{\Sigma'})/\approx_{\Sigma''}$. □

Prop. 2.12 is about the chain rule of automaton abstraction, which says an automaton abstraction can be replaced by a sequence of automaton abstractions, and the results are nonblocking equivalent to each other. we now introduce the main result about SAP.

Theorem 2.13. Suppose W_n is computed by SAP. Then $(\times_{i\in I} G_i)/\approx_{\Sigma'} \sqsubseteq W_n$. □

Proof: We use induction to show that

$$(\forall k : 1 \le k \le n)\, (\times_{j\in J_k} G_j)/\approx_{T_k} \sqsubseteq W_k \tag{3}$$

It is clear that $G_1/\approx_{T_1} \sqsubseteq W_1$. Suppose Equation (3) is true for $k \le l \in \mathbb{N}$. Then we need to show that it also holds for $k = l + 1$. By the procedure,

$$\begin{aligned}
(\times_{j\in J_{l+1}} G_j)/\approx_{T_{l+1}} &\cong ((\times_{j\in J_{l+1}} G_j)/\approx_{T_l\cup\Sigma_{l+1}})/\approx_{T_{l+1}} \quad \text{by Prop. 2.12} \\
&\sqsubseteq (((\times_{j\in J_l} G_j)/\approx_{T_l}) \times G_{l+1})/\approx_{T_{l+1}} \\
&\quad \text{because } \Sigma_{l+1} \cap \Sigma_{J_l} \subseteq T_l \cup \Sigma_{l+1} \text{ and Prop. 2.8 and Prop. 2.10} \\
&\sqsubseteq (W_l \times G_{l+1})/\approx_{T_{l+1}} \\
&\quad \text{by the induction hypothesis and Prop. 2.6 and Prop. 2.10} \\
&= W_{l+1}
\end{aligned}$$

Therefore Equation (3) holds for all k, particularly $k = n$. The proposition follows. ■

SAP allows us to obtain an abstraction of the entire system $G = \times_{i \in I} G_i$ in a sequential way. Thus, we can avoid computing G explicitly, which may be prohibitively large for systems of industrial size. Next, we discuss how to synthesize a supervisor based on automaton abstractions.

3 Supervisor Synthesis over Nondeterministic Finite-State Automata

As described in the introduction section, it has been a large volume of work on supervisor synthesis for nondeterministic systems. In this section we will mainly discuss how to apply automaton abstraction in such synthesis. We assume that the plant model $G \in \phi(\Sigma)$ is nondeterministic, but the requirement $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma$ is deterministic. Here H need not necessarily be standardized. Out goal is to synthesize a deterministic supervisor $S \in \phi(\Sigma')$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$. We first introduce some concepts.

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

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

Thus, $E_G(x)$ is simply the set of all events allowable at x in G. We now bring 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 $\tau \in \Sigma_{uc}$. Let $L(G) := \{s \in \Sigma^* | \xi(x_0, s) \neq \varnothing\}$.

Definition 3.1. Let $G = (X, \Sigma, \xi, x_0, X_m)$, $\Sigma' \subseteq \Sigma$, $A = (Y, \Sigma', \eta, y_0, Y_m) \in \phi(\Sigma')$ and $P : \Sigma^* \to \Sigma'^*$ be the natural projection. A is *state-controllable* with respect to G and Σ_{uc} if

$$(\forall s \in L(G \times A))(\forall x \in \xi(x_0, s))(\forall y \in \eta(y_0, P(s)))\, E_G(x) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_A(y)$$

□

The concept of state controllability is slightly different from the one used in the literature, e.g. [23], because of the involvement of Σ'. We can check that, A is state controllable implies that $L(G \times A)\Sigma_{uc} \cap L(G \times A) \subseteq L(G \times A)$. 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 and Σ_{uo} is the set of unobservable events. Let $P_o : \Sigma^* \to \Sigma_o^*$ be the natural projection.

Definition 3.2. Let $G = (X, \Sigma, \xi, x_0, X_m) \in \phi(\Sigma)$, $\Sigma' \subseteq \Sigma$ and $A = (Y, \Sigma', \eta, y_0, Y_m) \in \phi(\Sigma')$. A is *state-observable* with respect to G and P_o if for any $s, s' \in L(G \times A)$ with $P_o(s) = P_o(s')$, we have

$$(\forall (x, y) \in \xi \times \eta((x_0, y_0), s))(\forall (x', y') \in \xi \times \eta((x_0, y_0), s'))\, E_{G \times A}(x, y) \cap E_G(x') \cap \Sigma' \subseteq E_A(y')$$

□

State observability defined in Def. 3.2 is similar to the one defined in [27], except that in [27] the authors consider A to be a sub-automaton of G and only the silence event is unobservable. Def. 3.2 says that, 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. $P(s) = P(s')$), 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

$$(\forall s, s' \in L(G\times A))(\forall \sigma \in \Sigma)\, P_o(s) = P_o(s') \wedge s\sigma \in L(G\times A) \wedge s'\sigma \in L(G) \Rightarrow s'\sigma \in L(G\times A)$$

Thus, state observability implies language observability. But the reverse 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 3.3. Let $G = (X, \Sigma, \xi, x_0, X_m) \in \phi(\Sigma)$, $\Sigma' \subseteq \Sigma$, $A = (Y, \Sigma', \eta, y_0, Y_m) \in \phi(\Sigma')$ and $P : \Sigma^* \to \Sigma'^*$ be the natural projection. A is *state-normal* with respect to G and P_o if for any $s \in L(G \times A)$ and $s' \in \overline{P_o^{-1}(P_o(s))} \cap L(G \times A)$, we have

$$(\forall (x, y) \in \xi\times\eta((x_0, y_0), s'))(\forall s'' \in \Sigma^*)\, P_o(s's'') = P_o(s) \Rightarrow [\xi(x, s'') \neq \varnothing \Rightarrow \eta(y, P(s'')) \neq \varnothing]$$

□

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 language normal with respect to $L(G)$ and P_o. The reverse statement is not true unless both A and G are deterministic. Furthermore, we can check that state normality implies state observability. But the reverse statement is not true. We now introduce the concept of supervisors.

Definition 3.4. Given $G = (X, \Sigma, \xi, x_0, X_m) \in \phi(\Sigma)$ and $\Sigma' \subseteq \Sigma$, a deterministic finite-state automaton $S = (Y, \Sigma', \eta, y_0, Y_m) \in \phi(\Sigma')$ is a *nonblocking state-controllable state-observable (or state-normal) supervisor* of G with respect to a specification $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma'$ under P_o, if the following hold:

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) with respect to G and P_o □

The first condition of Def. 3.4 says that the closed-loop behavior (CLB) satisfies the specification H and the second one says CLB must be nonblocking. The third and fourth ones are self-explanatory.

Proposition 3.5. Given $G \in \phi(\Sigma)$ and $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$, there exists a nonblocking state-controllable state-observable (or state-normal) supervisor $S \in \phi(\Sigma')$ of G with respect to H if and only if there exists an automaton $A \in \phi(\Sigma')$ such that

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

Prop. 3.5 is about the existence of a nonblocking state-controllable state-observable (or state-normal) supervisor, whose proof indicates that such a supervisor is simply a (canonical) recognizer of an automaton A such that those four conditions hold. In the language-based framework 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 3.6. Given $G \in \phi(\Sigma)$ and $\Delta \subseteq \Sigma' \subseteq \Sigma$, let $S_i \in \phi(\Sigma')$ $(i = 1, 2)$ be a nonblocking state-controllable state-normal supervisor of G w.r.t. $H \in \phi(\Delta)$. Let $S \in \phi(\Sigma')$ be a deterministic automaton such that $N(S) = N(S_1) \cup N(S_2)$ and $L(S) = \overline{N(S)}$. Then S is a nonblocking state-controllable state-normal supervisor of G w.r.t. H. □

Prop. 3.6 says that the 'union' of two nonblocking state-controllable state-normal (NSCSN) supervisors is still a NSCSN supervisor. We define a set

$$\mathcal{CN}(G, H) := \{S \in \phi(\Sigma') | S \text{ is a NSCSN supervisor of } G \text{ w.r.t. } H \wedge L(S) \subseteq L(G)\}$$

From Prop. 3.6 we can derive that $\mathcal{CN}(G, H)$ has a unique element $\hat{S}$ such that for any $S \in \mathcal{CN}(G, H)$, we have $N(S) \subseteq N(\hat{S})$. We call $\hat{S}$ the *supremal nonblocking state-controllable state-normal supervisor* of G with respect to H. In reality we will be interested in such a supremal NSCSN supervisor, which can be computed from $G \times H$. In this paper we will not discuss this issue. Instead, we will only focus on two questions: (1) under what conditions is a nonblocking supervisor for an abstract plant model $G/\approx_{\Sigma'}$ with $\Sigma' \subseteq \Sigma$ also a nonblocking supervisor for the original plant model G? (2) under what conditions is a nonblocking supervisor $S \in \phi(\Sigma')$ for G also a nonblocking supervisor for the abstract model $G/\approx_{\Sigma'}$? To answer these questions, we need the following results.

Lemma 3.7. Given $G \in \phi(\Sigma)$ and $\Sigma' \subseteq \Sigma$, let $S \in \phi(\Sigma')$. 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 3.8. Given $G \in \phi(\Sigma)$ and $\Sigma' \subseteq \Sigma$, let $S \in \phi(\Sigma')$ 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 3.9. Given $G \in \phi(\Sigma)$ and $\Sigma' \subseteq \Sigma$, let $S \in \phi(\Sigma')$ 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. □

Theorem 3.10. Given $G \in \phi(\Sigma)$ and a deterministic automaton $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$, if there exists a nonblocking state-controllable state-observable (or state-normal) supervisor $S \in \phi(\Sigma')$ for $G/\approx_{\Sigma'}$ with respect to H, then S is also a nonblocking state-controllable state-observable (or state-normal) supervisor for G with respect to H. □

Proof: Since S is a nonblocking state-controllable state-observable (or state-normal) supervisor of $G/\approx_{\Sigma'}$ with respect to H, by Def. 3.4,

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 3.7, S is state-controllable with respect to G and Σ_{uc}. By Lemma 3.8, S is state observable with respect to G and P_o, or by Lemma 3.9, S is state-normal with respect to G and P_o. Since

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

By Theorem 2.9 we get that $B(G \times S) = \varnothing$. Finally, we show that $N(G \times S) \subseteq N(G \times H)$ as follows.

$$\begin{aligned}
& N((G/\approx_{\Sigma'}) \times S) \subseteq N((G/\approx_{\Sigma'}) \times H) \text{ by (1)} \\
\Rightarrow\ & N(G/\approx_{\Sigma'})||N(S) \subseteq N(G/\approx_{\Sigma'})||N(H) \\
\Rightarrow\ & N(G)||N(G/\approx_{\Sigma'})||N(S) \subseteq N(G)||N(G/\approx_{\Sigma'})||N(H) \\
\Rightarrow\ & N(G)||P(N(G))||N(S)) \subseteq N(G)||P(N(G))||N(H) \text{ by Prop. 2.4} \\
\Rightarrow\ & N(G)||N(S)) \subseteq N(G)||N(H) \text{ because } N(G) = N(G)||P(N(G)) \\
\Rightarrow\ & N(G \times S) \subseteq N(G \times H)
\end{aligned}$$

Therefore, the theorem is true. ■

Theorem 3.10 says that a nonblocking supervisor S for $G/\approx_{\Sigma'}$ is also a nonblocking supervisor for G.

Theorem 3.11. Given $G \in \phi(\Sigma)$ and a deterministic automaton $H \in \phi(\Delta)$ with $\Delta \subseteq \Sigma' \subseteq \Sigma$, suppose G is marking aware with respect to Σ' and $\Sigma_o \subseteq \Sigma'$. Then a nonblocking state-controllable state-observable (or state-normal) supervisor $S \in \phi(\Sigma')$ for G with respect to H and $\Sigma_o \subseteq \Sigma'$ is also a nonblocking state-controllable state-observable (or state-normal) supervisor for $G/\approx_{\Sigma'}$ with respect to H. □

Proof: Since S is a nonblocking state-controllable state-observable (or state-normal) supervisor of G with respect to H, by Def. 3.4,

1. $N(G \times S) \subseteq N((G/\approx_{\Sigma'}) \times H)$
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) with respect to G and P_o

By Lemma 3.7, S is state-controllable with respect to $G/\approx_{\Sigma'}$ and $\Sigma_{uc} \cap \Sigma'$. By Lemma 3.8, S is state-observable with respect to $G/\approx_{\Sigma'}$ and P'_o, or by Lemma 3.9, 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 2.9 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.

$$
\begin{aligned}
N((G/\approx_{\Sigma'}) \times S) &= N((G/\approx_{\Sigma'}) \times (S/\approx_{\Sigma'})) \text{ because } S/\approx_{\Sigma'} \cong S \text{ and by Cor. 2.7} \\
&= N((G \times S)/\approx_{\Sigma'}) \text{ by Prop. 2.8} \\
&= P(N(G \times S)) \text{ by Prop. 2.4} \\
&\subseteq P(N(G \times H)) \text{ By (1)} \\
&= N((G \times H)/\approx_{\Sigma'}) \text{ by Prop. 2.4} \\
&= N((G/\approx_{\Sigma'}) \times (H/\approx_{\Sigma'})) \text{ by Prop. 2.8} \\
&= N((G/\approx_{\Sigma'}) \times H) \text{ because } H/\approx_{\Sigma'} \cong H \text{ and by Cor. 2.7}
\end{aligned}
$$

Therefore, the theorem is true. ■

Theorem 3.11 says that, 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 $G/\approx_{\Sigma'}$.

4 Example

As an illustration we present the following example. Suppose we have two machines, which are functionally identical, except for individual event labels. The system is depicted in Figure 6. Each machine G_i $(i = 1, 2)$ has the following standard operations: (1)

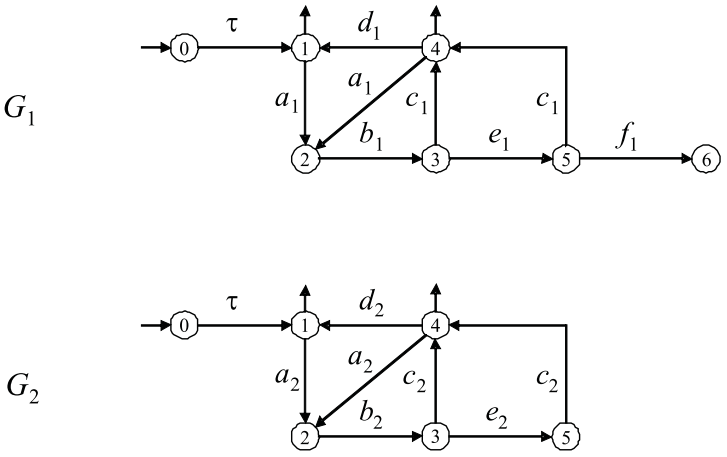


Figure 6: Example 4: A Simple Processing Unit

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 the observable alphabet $\Sigma_{i,o} = \Sigma_i$, namely every event is observable (for the purpose of simplicity). There is one specification $H \in \phi(\Delta)$ with $\Delta = \{e_1, e_2\}$, depicted in Figure 7, saying that if a work

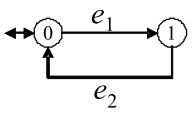


Figure 7: Example 4: The Specification $H \in \phi(\Delta)$

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 deterministic nonblocking state-controllable state-normal supervisor that enforces 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 motivation is that, since $\Delta \subseteq \Sigma'$, the abstraction $(G_1 \times G_2)/\approx_{\Sigma'}$ can capture 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. 2.8,

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

The results of $G_1/\approx_{\Sigma_1 \cap \Sigma'}$ and $G_2/\approx_{\Sigma_2 \cap \Sigma'}$ are depicted in Figure 8. The product of two

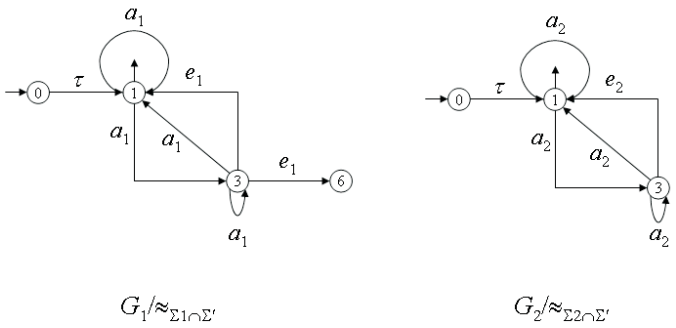


Figure 8: Example 4: The Abstractions $G_1/\approx_{\Sigma_1 \cap \Sigma'}$ and $G_2/\approx_{\Sigma_2 \cap \Sigma'}$

abstractions $G' := (G_1/\approx_{\Sigma_1 \cap \Sigma'}) \times (G_2/\approx_{\Sigma_2 \cap \Sigma'})$ is depicted in Figure 9, We now use G' and H to synthesize a supervisor. The product $G' \times H$ is depicted in Figure 9. Clearly, the transitions e_1 between states $(2,0)$ and $(3,1)$, and between states $(5,0)$ and $(4,1)$ 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 between states $(2,0)$ and $(1,1)$, and between states $(5,0)$ and $(6,1)$ must be disabled as well because, otherwise, the remaining automaton is not state-normal (and state-observable). After removing transitions e_1 at states $(2,0)$ and $(5,0)$ in Figure 9, the remaining reachable part A is depicted in Figure 10, which is nonblocking, state-controllable, state-normal (and state-observable). By Prop. 3.5 we get that, the canonical recognizer S of the marked behavior $N(A)$, depicted in Figure 11, is a nonblocking state-controllable and state-normal supervisor of G' with respect to 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 state-controllable state-normal supervisor of $G_1 \times G_2$ with respect to the specification H, as predicted by Theorem 3.10. We can

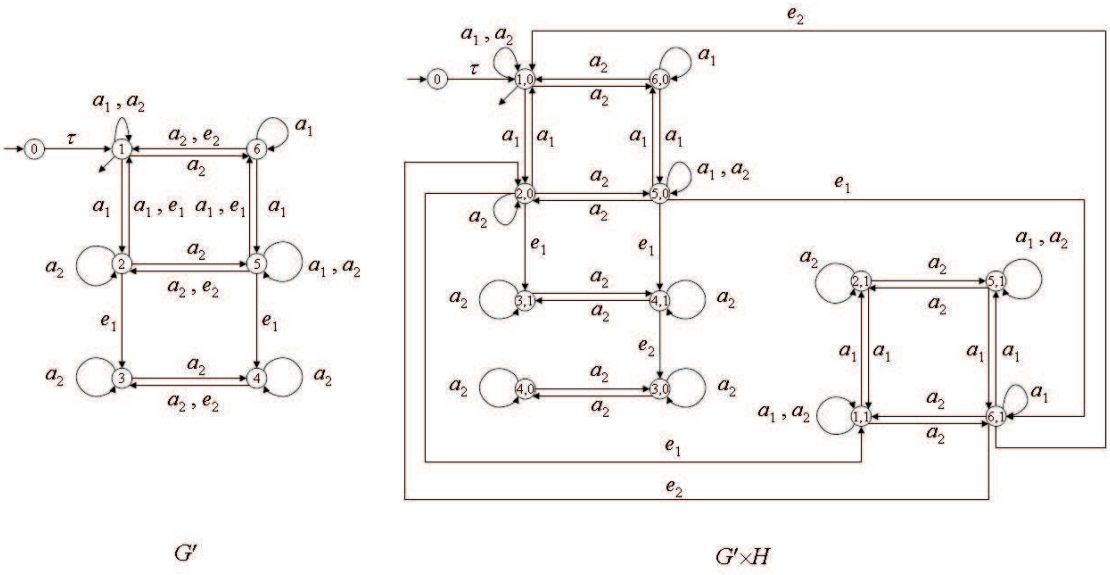


Figure 9: Example 4: The Products $G' = (G_1/\approx_{\Sigma_1\cap\Sigma'}) \times (G_2/\approx_{\Sigma_2\cap\Sigma'})$ and $G' \times H$

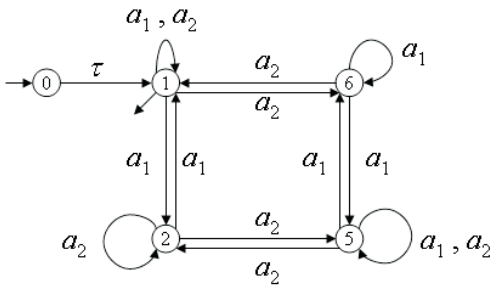


Figure 10: Example 4: Nonblocking, State-Controllable, State-Observable Automaton A

verify that the maximum number of states of any intermediate automata 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.

5 Conclusions

In this paper we first present a new abstraction technique and provide some properties. Then we apply this technique in supervisor synthesis. We consider the problem of synthesizing a deterministic supervisor for a nondeterministic plant model and a deterministic specification. After introducing the concepts of state controllability, state observability and state normality, we show that a nonblocking state-controllable state-observable (or state-normal) supervisor of an abstraction $G/\approx_{\Sigma'}$ under a specification H is also a nonblocking state-controllable state-observable (or state-normal) supervisor of the original

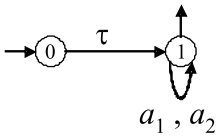


Figure 11: Example 4: The Supervisor $S \in \phi(\Sigma')$

plant G under the same specification. The reverse statement is also true, if all observable events are contained in Σ' and the plant G is marking aware with respect to Σ'. The concept of marking awareness is used to prevent extra blocking behaviors created from abstraction. Evidence shows that we may replace it by a more relaxed condition, which will be addressed in our future papers. In this paper we present a sufficient and necessary condition for the existence of a nonblocking supervisor and show that there exists a supremal nonblocking state-controllable and state-normal supervisor for a plant G and a specification H. The concrete procedure to compute such a supremal supervisor is not provided, owing to the main focus of this paper and the page limit as well. It will be addressed in our future papers.

Appendix

1. Proof of Prop. 2.4: 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 $< x > \in \xi'(< x_0 >, P(t))$. Because

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

we have that

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

Thus, $s = P(t) \in B(G/\approx_{\Sigma'}$.
To show that $P(N(G)) \subseteq N(G/\approx_{\Sigma'})$, 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, $\xi(x_0, t) \cap X_m \neq \varnothing$ implies $\xi'(< x_0 >, P(t)) \cap (X_m/\approx_{\Sigma'}) \neq \varnothing$. 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

$$\xi'(< x_0 >, s) \cap (X_m/\approx_{\Sigma'}) \neq \varnothing$$

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 $s \in B(G/\approx_{\Sigma'})$, we have

$$(\exists < x > \in \xi'(< x_0 >, s))(\forall s' \in \Sigma'^*)\, \xi'(< x >, s') \cap (X_m/\approx_{\Sigma'}) = \varnothing$$

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 $\xi'(< x >, \epsilon) \cap (X_m/\approx_{\Sigma'}) \neq \varnothing$. 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, which means $t \in B(G)$. Thus, $s = P(t) \in P(B(G))$. ■

2. Proof of Prop. 2.8: 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 $P : (\Sigma_1 \cup \Sigma_2)^* \to \Sigma'^*$, $P_i : \Sigma_i^* \to \hat{\Sigma}_i^*$, $\hat{P}_i : \Sigma'^* \to \hat{\Sigma}_i^*$ and $Q_i : (\Sigma_1 \cup \Sigma_2)^* \to \Sigma_i^*$ be natural projections, ξ' the transition map of $(G_1 \times G_2)/\approx_{\Sigma'}$ and ξ_i' be the transition map of $G_i/\approx_{\hat{\Sigma}_i}$ $(i = 1, 2)$.
First, we have the following,

$$\begin{aligned} N((G_1 \times G_2)/\approx_{\Sigma'}) &= P(N(G_1 \times G_2)) \text{ by Prop. 2.4} \\ &= P(N(G_1)||N(G_2)) \\ &= P_1(N(G_1))||P_2(N(G_2)) \text{ because } \Sigma_1 \cap \Sigma_2 \subseteq \Sigma' \\ &= N(G_1/\approx_{\hat{\Sigma}_1})||N(G_2/\approx_{\hat{\Sigma}_2}) \text{ by Prop. 2.4} \\ &= N((G_1/\approx_{\hat{\Sigma}_1}) \times (G_2/\approx_{\hat{\Sigma}_2})) \end{aligned}$$

Next, we show that $B((G_1 \times G_2)/\approx_{\Sigma'}) \subseteq B((G_1/\approx_{\hat{\Sigma}_1}) \times (G_2/\approx_{\hat{\Sigma}_2}))$. 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

$$< (x_1, x_2) >_{\Sigma'} \in \xi'(< (x_{1,0}, x_{2,0}) >_{\Sigma'}, s) \wedge (\forall s' \in \Sigma'^*)\, \xi'(< (x_1, x_2) >_{\Sigma'}, s') \cap (X_{1,m} \times X_{2,m})/\approx_{\Sigma'} = \varnothing$$

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

$$(x_1, x_2) \in \xi_1 \times \xi_2((x_{1,0}, x_{2,0}), t) \wedge (\forall t' \in \Sigma^*)\, \xi_1 \times \xi_2((x_1, x_2), t') \cap (X_{1,m} \times X_{2,m}) \neq \varnothing \Rightarrow t' \in ((\Sigma_1 \cup \Sigma_2) - \Sigma')^*$$

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

$$(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \in \xi_1' \times \xi_2'((< x_{1,0} >_{\hat{\Sigma}_1}, < x_{2,0} >_{\hat{\Sigma}_1}), s)$$

We claim that $(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2})$ 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

$$\xi_1' \times \xi_2'((< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}), s') \cap (X_{1,m}/\approx_{\hat{\Sigma}_1}) \times (X_{2,m}/\approx_{\hat{\Sigma}_2}) \neq \varnothing$$

Since $(x_1, x_2) \notin X_{1,m} \times X_{2,m}$, $(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \notin (X_{1,m}/\approx_{\hat{\Sigma}_1}) \times (X_{2,m}/\approx_{\hat{\Sigma}_2})$. 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

$$(\forall t' \in \Sigma^*)\, \xi_1 \times \xi_2((x_1, x_2), t') \cap (X_{1,m} \times X_{2,m}) \neq \varnothing \Rightarrow t' \in ((\Sigma_1 \cup \Sigma_2) - \Sigma')^*$$

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

$$< (x_1, x_2) >_{\Sigma'} \in \xi'(< (x_{1,0}, x_{2,0}) >_{\Sigma'}, s)$$

we have

$$(\exists t \in \Sigma^*)\, P_(t) = s \wedge (x_1, x_2) \in \xi((x_{1,0}, x_{2,0}), t)$$

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 expression:

$$(< x_{1,0} >_{\hat{\Sigma}_1}, < x_{2,0} >_{\hat{\Sigma}_2}) \in \xi_1' \times \xi_2'((< x_{1,0} >_{\hat{\Sigma}_1}, < x_{2,0} >_{\hat{\Sigma}_2}), \epsilon)$$

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

$$(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \in \xi_1' \times \xi_2'((< x_{1,0} >_{\hat{\Sigma}_1}, < x_{2,0} >_{\hat{\Sigma}_2}), s)$$

Thus, in either case we have

$$(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \in \xi_1' \times \xi_2'((< x_{1,0} >_{\hat{\Sigma}_1}, < x_{2,0} >_{\hat{\Sigma}_2}), s)$$

We now show that

$$N_{(G_1/\approx_{\hat{\Sigma}_1}) \times (G_2/\approx_{\hat{\Sigma}_2})}(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \subseteq N_{(G_1 \times G_2)/\approx_{\Sigma'}}(< (x_1, x_2) >_{\Sigma'})$$

Let $s' \in N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2})$. If $s' = \epsilon$, then

$$(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \in (X_{1,m}/\approx_{\hat{\Sigma}_1}) \times (X_{2,m}/\approx_{\hat{\Sigma}_2})$$

from which we get $(x_1, x_2) \in X_{1,m} \times X_{2,m}$. Thus, $< (x_1, x_2) >_{\Sigma'} \in (X_{1,m} \times X_{2,m})/\approx_{\Sigma'}$, namely $\epsilon \in N_{(G_1\times G_2)/\approx_{\Sigma'}}(< (x_1, x_2) >_{\Sigma'})$. 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 abstraction, we get $\xi'(< (x_1, x_2) >_{\Sigma'}, s') \cap (X_{1,m} \times X_{2,m})/\approx_{\Sigma'} \neq \varnothing$. Thus, $s' \in N_{(G_1\times G_2)/\approx_{\Sigma'}}(< (x_1, x_2) >_{\Sigma'})$. In either case, we have

$$N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \subseteq N_{(G_1\times G_2)/\approx_{\Sigma'}}(< (x_1, x_2) >_{\Sigma'})$$

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

$$B((G_1/\approx_{\hat{\Sigma}_1}) \times (G_2/\approx_{\hat{\Sigma}_2})) = B((G_1 \times G_2)/\approx_{\Sigma'})$$

we only need to prove one direction ($\subseteq$), because the other direction ($\supseteq$) has been proved above. 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

$$(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \in \xi'_1 \times \xi'_2(< (< x_{1,0} >_{\hat{\Sigma}_1}, < x_{2,0} >_{\hat{\Sigma}_2}), s) \tag{4}$$

and

$$(\forall s' \in \Sigma'^*)\, \xi'_1 \times \xi'_2((< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}), s') \cap ((X_{1,m}/\approx_{\hat{\Sigma}_1}) \times (X_{2,m}/\approx_{\hat{\Sigma}_2})) = \varnothing \tag{5}$$

From Expression (4) we get that

$$(\exists t \in (\Sigma_1 \cup \Sigma_2)^*)\, P(t) = s \wedge (x_1, x_2) \in \xi_1 \times \xi_2((x_{1,0}, x_{2,0}), t) \tag{6}$$

From Expression (5) 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 (6) and the fact that $\Sigma_1 \cap \Sigma_2 \subseteq \Sigma'$ we have

$$< (x_1, x_2) >_{\Sigma'} \in \xi'(< (x_{1,0}, x_{2,0}) >_{\Sigma'}, s)$$

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

$$\xi'(< (x_1, x_2) >_{\Sigma'}, s') \cap (X_{1,m} \times X_{2,m})/\approx_{\Sigma'} \neq \varnothing)$$

Since $(x_1, x_2) \notin X_{1,m} \times X_{2,m}$, we get that $< (x_1, x_2) >_{\Sigma'} \notin (X_{1,m} \times X_{2,m})/\approx_{\Sigma'}$. 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

$$\xi'_1 \times \xi'_2((< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}), s') \cap ((X_{1,m}/\approx_{\hat{\Sigma}_1}) \times (X_{2,m}/\approx_{\hat{\Sigma}_2})) \neq \varnothing$$

which contradicts Expression (5). Thus, the claim is true, namely $s \in B((G_1 \times G_2)/\approx_{\Sigma'})$.
Let $s \in \overline{N((G_1/\approx_{\hat{\Sigma}_1}) \times (G_2/\approx_{\hat{\Sigma}_2}))}$. For any $(x_1, x_2) \in X_1 \times X_2$ with

$$(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \in \xi'_1 \times \xi'_2((< x_{1,0} >_{\hat{\Sigma}_1}, < x_{2,0} >_{\hat{\Sigma}_2}), s)$$

we have

$$(\exists t \in \Sigma^*)\, P_(t) = s \wedge (x_1, x_2) \in \xi((x_{1,0}, x_{2,0}), t)$$

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 expression:

$$< (x_{1,0}, x_{2,0}) >_{\Sigma'} \in \xi'(< (x_{1,0}, x_{2,0}) >_{\Sigma'}, \epsilon)$$

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

$$< (x_1, x_2) >_{\Sigma'} \in \xi'(< (x_{1,0}, x_{2,0}) >_{\Sigma'}, s)$$

Thus, in either case we have

$$< (x_1, x_2) >_{\Sigma'} \in \xi'(< (x_{1,0}, x_{2,0}) >_{\Sigma'}, s)$$

We now show that

$$N_{(G_1\times G_2)/\approx_{\Sigma'}}(< (x_1,x_2) >_{\Sigma'}) \subseteq N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2})$$

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

$$< (x_1,x_2) >_{\Sigma'} \in (X_{1,m} \times X_{2,m})/\approx_{\Sigma'}$$

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

$$(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}) \in (X_{1,m}/\approx_{\hat{\Sigma}_1}) \times (X_{2,m}/\approx_{\hat{\Sigma}_2})$$

namely $\epsilon \in N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2})$. If $s' \neq \epsilon$, then

$$< (x_1,x_2) >_{\Sigma'} \notin (X_{1,m} \times X_{2,m})/\approx_{\Sigma'}$$

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 automaton abstraction, we get that

$$\xi_1' \times \xi_2'((< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}), s') \cap (X_{1,m}/\approx_{\hat{\Sigma}_1}) \times (X_{2,m}/\approx_{\hat{\Sigma}_2}) \neq \varnothing$$

Thus, $s' \in N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2})$. 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

$$(\exists < \hat{x}_2 >_{\hat{\Sigma}_2} \in X_{2,m}/\approx_{\hat{\Sigma}_2}) < \hat{x}_2 >_{\hat{\Sigma}_2} \in \xi_2'(< x_2 >_{\hat{\Sigma}_2}), \hat{P}_2(s'))$$

Thus,

$$(< x_1 >_{\hat{\Sigma}_1}, < \hat{x}_2 >_{\hat{\Sigma}_2}) \in \xi_1' \times \xi_2'((< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2}), s')$$

which means $s' \in N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2})$. 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

$$N_{(G_1\times G_2)/\approx_{\Sigma'}}(< (x_1,x_2) >_{\Sigma'}) \subseteq N_{(G_1/\approx_{\hat{\Sigma}_1})\times(G_2/\approx_{\hat{\Sigma}_2})}(< x_1 >_{\hat{\Sigma}_1}, < x_2 >_{\hat{\Sigma}_2})$$

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

3. Proof of Prop. 2.6: 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 $P : (\Sigma \cup \Sigma')^* \to \Sigma^*$ and $P' : (\Sigma \cup \Sigma')^* \to \Sigma'^*$ 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

$$N_{G_2\times G_3}(x_2,x_3) = N_{G_2}(x_2)||N_{G_3}(x_3) \subseteq N_{G_1}(x_1)||N_{G_3}(x_3) = N_{G_1\times G_3}(x_1,x_3) = \varnothing$$

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, follow the argument of 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 $N_{G_2\times G_3}(x_2,x_3) \subseteq N_{G_1\times G_3}(x_1,x_3)$. ■

4. Proof of Prop. 2.10: Let $G_i = (X_i, \Sigma, \xi_i, x_{i,0}, X_{i,m})$, where $i = 1, 2$, and $P : \Sigma^* \to \Sigma'^*$ be the natural projection. Since $G_1 \sqsubseteq G_2$, by Prop. 2.4 we have

$$N(G_1/\approx_{\Sigma'}) = P(N(G_1)) = P(N(G_2)) = N(G_2/\approx_{\Sigma'})$$

To show $B(G_1/\approx_{\Sigma'}) \subseteq B(G_2/\approx_{\Sigma'})$, let ξ_i' $(i = 1, 2)$ be the transition map of $G_i/\approx_{\Sigma'}$. For any $s \in B(G_1/\approx_{\Sigma'})$, there exists $x_1 \in X_1$ such that

$$< x_1 > \in \xi_1'(< x_{1,0} >, s) \wedge (\forall s' \in \Sigma'^*)\, \xi_1'(< x_1 >, s') \cap X_{1,m}/\approx_{\Sigma'} = \varnothing$$

which means there exists $t \in \Sigma^*$ such that

$$P(t) = s \wedge x_1 \in \xi_1(x_{1,0}, t) \wedge (\forall t' \in \Sigma^*)\, \xi_1(x_1, t') \cap X_{1,m} \neq \varnothing \Rightarrow t' \in (\Sigma - \Sigma')^*$$

Thus, $N_{G_1}(x_1) \subseteq (\Sigma - \Sigma')^*$. There are two cases. Case 1: x_1 is a blocking state of G_1. Then $t \in B(G_1)$, which means $s = P(t) \in P(B(G_1))$. Since $G_1 \sqsubseteq G_2$, we have $B(G_1) \subseteq B(G_2)$. Thus, by Prop. 2.4, we have $s \in P(B(G_2)) \subseteq B(G_2/\approx_{\Sigma'})$. Case 2: x_1 is a nonblocking state. Thus $t \in N(G_1)$. Clearly $x_1 \notin X_{1,m}$. Since $G_1 \sqsubseteq G_2$, there exists $x_2 \in X_2$ such that

$$x_2 \in \xi_2(x_{2,0}, t) \wedge N_{G_1}(x_1) = N_{G_2}(x_2) \wedge [x_1 \in X_{1,m} \iff x_2 \in X_{2,m}]$$

Since $N_{G_1}(x_1) \subseteq (\Sigma - \Sigma')^*$, we have

$$(\forall t' \in \Sigma^*)\, \xi_2(x_2, t') \cap X_{1,m} \neq \varnothing \Rightarrow t' \in (\Sigma - \Sigma')^*$$

Since $x_1 \notin X_{1,m}$, we have $x_2 \notin X_{2,m}$. Thus, by the definition of automaton abstraction,

$$< x_2 > \in \xi_2'(< x_{2,0} >, P(t)) \wedge (\forall s' \in \Sigma'^*)\, \xi_2'(< x_2 >, s') \cap X_{2,m}/\approx_{\Sigma'} = \varnothing$$

which means $s = P(t) \in B(G_2/\approx_{\Sigma'})$. Thus, in either case we have $B(G_1/\approx_{\Sigma'}) \subseteq B(G_2/\approx_{\Sigma'})$.
Finally, for each $s \in \overline{N(G_1/\approx_{\Sigma'})}$, there exists $x_1 \in X_1$ such that

$$< x_1 > \in \xi_1'(< x_{1,0} >, s) \wedge (\exists s' \in \Sigma'^*)\, \xi_1'(< x_1 >, s') \cap X_{1,m}/\approx_{\Sigma'} \neq \varnothing$$

which means there exists $t \in \Sigma^*$ with $P(t) = s$ such that

$$x_1 \in \xi_1'(x_{1,0}, t) \wedge (\exists t' \in \Sigma'^*)\, P(t') = s' \wedge \xi_1(x_1, t') \cap X_{1,m} \neq \varnothing$$

Clearly, $t \in \overline{N(G_1)}$. Thus, by $G_1 \sqsubseteq G_2$, we have

$$(\exists x_2 \in \xi_2(x_{2,0}, t))\, N_{G_1}(x_1) \supseteq N_{G_2}(x_2) \wedge [x_1 \in X_{1,m} \iff x_2 \in X_{2,m}]$$

Since G_2 is standardized, $< x_2 > \in \xi_2'(< x_{2,0} >, s)$. For any $s' \in N_{G_2/\approx_{\Sigma'}}(< x_2 >)$, we have $\xi_2'(< x_2 >, s') \cap X_{2,m}/\approx_{\Sigma'} \neq \varnothing$. If $s' = \epsilon$, then $x_2 \in X_{2,m}$, which means $x_1 \in X_{1,m}$, namely $\epsilon \in N_{G_1/\approx_{\Sigma'}}(< x_1 >)$. If $s' \neq \epsilon$, then there exists $t' \in \Sigma^*$ with $P(t') = s'$ such that $\xi_2(x_2, t') \cap X_{2,m} \neq \varnothing$, which means $t' \in N_{G_2}(x_2) \subseteq N_{G_1}(x_1)$. Thus,

$$\xi_1'(< x_1 >, s') \cap X_{1,m}/\approx_{\Sigma'} \neq \varnothing$$

namely $s' \in N_{G_1/\approx_{\Sigma'}}(< x_1 >)$. Thus, $N_{G_2/\approx_{\Sigma'}}(x_2) \subseteq N_{G_1/\approx_{\Sigma'}}(x_1)$. ■

5. Proof of Prop. 2.12: Let ξ'' the transition map of $G/\approx_{\Sigma''}$, and ξ''' the transition map of $(G/\approx_{\Sigma'})/\approx_{\Sigma''}$. Let $P_{12} : \Sigma^* \to \Sigma'^*$, $P_{13} : \Sigma^* \to \Sigma''^*$ and $P_{23} : \Sigma'^* \to \Sigma''^*$ be natural projections. We first show that $G/\approx_{\Sigma''} \sqsubseteq (G/\approx_{\Sigma'})/\approx_{\Sigma''}$.
By Prop. 2.4 we have

$$N(G/\approx_{\Sigma''}) = P_{13}(N(G)) = P_{23}(P_{12}(N(G))) = P_{23}(N(G/\approx_{\Sigma'})) = N((G/\approx_{\Sigma'})/\approx_{\Sigma''})$$

We now show $B(G/\approx_{\Sigma''}) \subseteq B((G/\approx_{\Sigma'})/\approx_{\Sigma''})$. Let $s \in B(G/\approx_{\Sigma''})$. There exists $x \in X$ such that

$$< x >_{\Sigma''} \in \xi''(< x_0 >_{\Sigma''}, s) \wedge (\forall s' \in \Sigma''^*)\, \xi''(< x >_{\Sigma''}, s') \cap X_m/\approx_{\Sigma''} = \varnothing$$

Thus, there exists $t \in \Sigma^*$ with $P_{13}(t) = s$ such that

$$x \in \xi(x_0, t) \wedge (\forall t' \in \Sigma^*)\, \xi(x, t') \cap X_m \neq \varnothing \Rightarrow t' \in (\Sigma - \Sigma'')^* \tag{7}$$

We have two cases to consider. Case 1: x is a blocking state. Then clearly $t \in B(G)$. By Prop. 2.4, $P_{13}(t) = s = P_{23}(P_{12}(t)) \in P_{23}(P_{12}(B(G))) \subseteq B((G/\approx_{\Sigma'})/\approx_{\Sigma''})$. Case 2: x is a nonblocking state. Clearly $x \notin X_m$, which means $<< x >_{\Sigma'}>_{\Sigma''} \notin (X_m/\approx_{\Sigma'})/\approx_{\Sigma''}$. Thus, from Expression (7) and the definition of automaton abstraction, we get that

$$<< x >_{\Sigma'}>_{\Sigma''} \in \xi'''(<< x_0 >_{\Sigma'}>_{\Sigma''}, s) \wedge (\forall s' \in \Sigma''^*)\, \xi'''(<< x >_{\Sigma'}>_{\Sigma''}, s') \cap (X_m/\approx_{\Sigma'})/\approx_{\Sigma''} = \varnothing$$

Thus, $s \in B((G/\approx_{\Sigma'})/\approx_{\Sigma''})$. In either case we have $B(G/\approx_{\Sigma''}) \subseteq B((G/\approx_{\Sigma'})/\approx_{\Sigma''})$. Let $s \in \overline{N(G/\approx_{\Sigma''})}$. For any $x \in X$ with $< x >_{\Sigma''} \in \xi''(< x_0 >_{\Sigma''}, s)$, we have that

$$(\exists t \in \Sigma^*)\, P_{13}(t) = s \wedge x \in \xi(x_0, t)$$

Since G is standardize, if $s = \epsilon$, then $t = \epsilon$, which means $x = x_0$. Clearly, we have the following expression: $<< x_0 >_{\Sigma'}>_{\Sigma''} \in \xi'''(<< x_0 >_{\Sigma'}>_{\Sigma''}, \epsilon)$. If $s \neq \epsilon$, then by the definition of automaton abstraction and the assumption that $\Sigma'' \subseteq \Sigma' \subseteq \Sigma$, we get that $<< x >_{\Sigma'}>_{\Sigma''} \in \xi'''(<< x_0 >_{\Sigma'}>_{\Sigma''}, s)$. Thus, in either case we have $<< x >_{\Sigma'}>_{\Sigma''} \in \xi'''(<< x_0 >_{\Sigma'}>_{\Sigma''}, s)$. We now show that

$$N_{(G/\approx_{\Sigma'})/\approx_{\Sigma''}}(<< x >_{\Sigma'}>_{\Sigma''}) \subseteq N_{G/\approx_{\Sigma''}}(< x >_{\Sigma''})$$

Let $s' \in N_{(G/\approx_{\Sigma'})/\approx_{\Sigma''}}(<< x >_{\Sigma'}>_{\Sigma''})$. If $s' = \epsilon$, then $<< x >_{\Sigma'}>_{\Sigma''} \in (X_m/\approx_{\Sigma'})/\approx_{\Sigma''}$, which means $x \in X_m$. Thus, $< x >_{\Sigma''} \in X_m/\approx_{\Sigma''}$, namely $\epsilon \in N_{G/\approx_{\Sigma''}}(< x >_{\Sigma''})$. If $s' \neq \epsilon$. Then there exists $t' \in \Sigma^*$ with $P_{13}(t') = s'$ such that $\xi(x, t') \cap X_m \neq \varnothing$. Since $P_{13}(t') \neq \epsilon$, by the definition of automaton abstraction,

$$\xi''(< x >_{\Sigma''}, s') \cap X_m/\approx_{\Sigma''} \neq \varnothing$$

Thus, $s' \in N_{G/\approx_{\Sigma''}}(< x >_{\Sigma''})$. In either case, we get

$$N_{(G/\approx_{\Sigma'})/\approx_{\Sigma''}}(<< x >_{\Sigma'}>_{\Sigma''}) \subseteq N_{G/\approx_{\Sigma''}}(< x >_{\Sigma''})$$

Next, we show that $(G/\approx_{\Sigma'})/\approx_{\Sigma''} \sqsubseteq G/\approx_{\Sigma''}$. We first show

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

Let $s \in B((G/\approx_{\Sigma'})/\approx_{\Sigma''})$. There exists $x \in X$ such that

$$<< x >_{\Sigma'}>_{\Sigma''} \in \xi'''(<< x_0 >_{\Sigma'}>_{\Sigma''}, s) \wedge (\forall s' \in \Sigma''^*)\, \xi'''(<< x >_{\Sigma'}>_{\Sigma''}, s') \cap (X_m/\approx_{\Sigma'})/\approx_{\Sigma''} = \varnothing$$

Thus, there exists $t \in \Sigma^*$ with $P_{13}(t) = s$ such that

$$x \in \xi(x_0, t) \wedge (\forall t' \in \Sigma^*)\, \xi(x, t') \cap X_m \neq \varnothing \Rightarrow t' \in (\Sigma - \Sigma'')^* \tag{8}$$

We have two cases to consider. Case 1: x is a blocking state. Then clearly $t \in B(G)$. By Prop. 2.4, $P_{13}(t) = s \in P_{13}(B(G)) \subseteq B(G/\approx_{\Sigma''})$. Case 2: x is a nonblocking state. Clearly $x \notin X_m$, which means $< x >_{\Sigma''} \notin X_m/\approx_{\Sigma''}$. Thus, from Expression (8) we get that

$$< x >_{\Sigma''} \in \xi''(< x_0 >_{\Sigma''}, s) \wedge (\forall s' \in \Sigma''^*)\, \xi''(< x >_{\Sigma''}, s') \cap X_m/\approx_{\Sigma''} = \varnothing$$

Thus, $s \in B(G/\approx_{\Sigma''})$. In either case we have $B((G/\approx_{\Sigma'})/\approx_{\Sigma''}) \subseteq B(G/\approx_{\Sigma''})$. Let $s \in \overline{N((G/\approx_{\Sigma'})/\approx_{\Sigma''})}$. For any $x \in X$ with $<< x >_{\Sigma'}>_{\Sigma''} \in \xi'''(<< x_0 >_{\Sigma'}>_{\Sigma''}, s)$, we have

$$(\exists t \in \Sigma^*)\, P_{13}(t) = s \wedge x \in \xi(x_0, t)$$

Since G is standardize, if $s = \epsilon$, then $t = \epsilon$, which means $x = x_0$. Clearly, we have the following expression: $< x_0 >_{\Sigma''} \in \xi''(< x_0 >_{\Sigma''}, \epsilon)$. If $s \neq \epsilon$, then by the definition of automaton abstraction and the assumption that $\Sigma'' \subseteq \Sigma' \subseteq \Sigma$, we get that

$$< x >_{\Sigma''} \in \xi''(< x_0 >_{\Sigma''}, s)$$

Thus, in either case we have $< x >_{\Sigma''} \in \xi''(< x_0 >_{\Sigma''}, s)$. We now show that

$$N_{G/\approx_{\Sigma''}}(< x >_{\Sigma''}) \subseteq N_{(G/\approx_{\Sigma'})/\approx_{\Sigma''}}(<< x >_{\Sigma'}>_{\Sigma''})$$

Let $s' \in N_{G/\approx_{\Sigma''}}(< x >_{\Sigma''})$. If $s' = \epsilon$, then $< x >_{\Sigma''} \in X_m/\approx_{\Sigma''}$, which means $x \in X_m$. Thus, $<< x >_{\Sigma'}>_{\Sigma''} \in (X_m/\approx_{\Sigma'})/\approx_{\Sigma''}$, namely $\epsilon \in N_{(G/\approx_{\Sigma'})/\approx_{\Sigma''}}(<< x >_{\Sigma'}>_{\Sigma''})$. If

$s' \neq \epsilon$. Then there exists $t' \in \Sigma^*$ with $P_{13}(t') = s'$ such that $\xi(x, t') \cap X_m \neq \varnothing$. Since $P_{13}(t') \neq \epsilon$, by the definition of automaton abstraction, we get that

$$\xi'''(<< x >_{\Sigma'} >_{\Sigma''}, s') \cap (X_m / \approx_{\Sigma'}) / \approx_{\Sigma''} \neq \varnothing$$

Thus, $s' \in N_{(G/\approx_{\Sigma'})/\approx_{\Sigma''}}(<< x >_{\Sigma'} >_{\Sigma''})$. In either case, we have

$$N_{G/\approx_{\Sigma''}}(< x >_{\Sigma''}) \subseteq N_{(G/\approx_{\Sigma'})/\approx_{\Sigma''}}(<< x >_{\Sigma'} >_{\Sigma''})$$

The proposition follows. ■

6. Proof of Prop. 3.5: The ONLY IF part is obvious. So we only need to show the IF part. Let S be a (canonical) recognizer of $N(A)$, i.e. $N(S) = N(A)$ and $L(S) = \overline{N(A)} = L(A)$ (because $B(G \times A) = \varnothing$). Then we have

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

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 $s \in B(G \times S)$ such that

$$(\exists (x, y) \in \xi \times \eta((x_0, y_0), s))(\forall s' \in \Sigma^*)\, \xi \times \eta((x, y), s') \cap (X_m \times Y_m) = \varnothing$$

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

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

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

$$E_{G \times S}(x, y) \cap E_G(x') \cap \Sigma' \nsubseteq E_S(y')$$

Since S is deterministic, we have that

$$(\exists \sigma \in \Sigma')\, s\sigma \in L(G) \wedge s'\sigma \in L(G) \wedge P(s)\sigma \in L(S) \wedge P(s')\sigma \notin L(S)$$

Since $L(S) = L(A)$, 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

$$E_{G \times A}(x, z) \cap E_G(x') \cap \Sigma' \nsubseteq E_A(z')$$

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

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

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,

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

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

7. Proof of Prop. 3.6: Since $N(G \times S_i) \subseteq N(G \times H)$ for $i = 1, 2$, we have

$$N(G \times S) = N(G)||N(S) = N(G)||(N(S_1) \cup N(S_2)) = N(G \times S_1) \cup N(G \times S_2) \subseteq N(G \times H)$$

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 exists $s \in B(G \times S)$ such that

$$(\exists (x, y) \in \xi \times \eta((x_0, y_0), s))(\forall s' \in \Sigma^*)\, \xi \times \eta((x, y), s') \cap (X_m \times Y_m) = \varnothing$$

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 $B(G \times S_1) = \varnothing$, we get that

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

Thus, $\xi(x, s') \cap X_m \neq \varnothing$ and $P(ss') \in N(S_1) \subseteq N(S)$. Since S is deterministic, we get $\eta(y, P(s')) \cap Y_m \neq \varnothing$. 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 $P(s) \in L(S) = \overline{N(S_1) \cup N(S_2)}$, without loss of generality, suppose $P(s) \in \overline{N(S_1)} = L(S_1)$ (because $B(G \times S_1) = \varnothing$). Then

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

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

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

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 $P(s) \in L(S) = L(S_1) \cup L(S_2)$. 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 $\eta_1(y_1, P(s'')) = \varnothing$, 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 $(x', y_1) \in \xi \times \eta_1((x_0, y_{1,0}), \hat{s})$, $\xi(x', \hat{s}'s'') \neq \varnothing$ but $\eta_1(y_1, \hat{s}'s'') = \varnothing$, which still contradicts the fact that S_1 is state-normal with respect to G and P_o. Thus,

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

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

8. Proof of Lemma 3.7: 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

$$(\forall s \in L(G \times S))(\forall x \in \xi(x_0, s))(\forall y \in \eta(y_0, P(s)))\, E_G(x) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_S(y) \quad (9)$$

where $P : \Sigma^* \to \Sigma'^*$ is the natural projection, and there exists $t \in L((G/\approx_{\Sigma'}) \times S)$ such that

$$(\exists < x > \in \xi'(< x_0 >, t))(\exists y \in \eta(y_0, t))\, E_{G/\approx_{\Sigma'}}(< x >) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y) \quad (10)$$

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

$$E_{G/\approx_{\Sigma'}}(< x >) = \{\sigma \in \Sigma' | (\exists u \in (\Sigma - \Sigma'^*))(\exists x' \in \xi(x, u))\, \sigma \in E_G(x')\}$$

Thus, $E_{G/\approx_{\Sigma'}}(< x >) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y)$ 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 (10) 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

$$(\exists su \in L(G \times S))(\exists x' \in \xi(x_0, su))(\exists y \in \eta(y_0, P(s)))\, E_G(x') \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y))$$

which contradicts expression (9). 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 any $t \in L((G/\approx_{\Sigma'}) \times S)$,

$$(\forall < x > \in \xi'(< x_0 >, t))(\forall y \in \eta(y_0, t))\, E_{G/\approx_{\Sigma'}}(< x >) \cap \Sigma_{uc} \cap \Sigma' \subseteq E_S(y) \quad (11)$$

and

$$(\exists s \in L(G \times S))(\exists x \in \xi(x_0, s))(\exists y \in \eta(y_0, P(s)))\, E_G(x) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y) \quad (12)$$

Since G is standardized, from expression (12) we get that $< x > \in \xi'(< x_0 >, P(s))$. Since

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

we get that

$$(\exists < x > \in \xi'(< x_0 >, P(s)))(\exists y \in \eta(y_0, P(s)))\, E_{G/\approx_{\Sigma'}}(< x >) \cap \Sigma_{uc} \cap \Sigma' \not\subseteq E_S(y)$$

which contradicts expression (11). Thus, the ONLY IF part must be true. ■

9. Proof of Lemma 3.8: (1) Let S is state observable with respect to $G/\approx_{\Sigma'}$ and P'_o. Thus, for any $s, s' \in L((G/\approx_{\Sigma'}) \times S)$ with $P'_o(s) = P'_o(s')$, and for any $(< x >, y) \in \xi' \times \eta((< x_0 >, y_0), s)$ and $(< x' >, y') \in \xi' \times \eta((< x_0 >, y_0), s')$, we have

$$E_{(G/\approx_{\Sigma'}) \times S}(< x >, y) \cap E_{G/\approx_{\Sigma'}}(< x' >) \cap \Sigma' \subseteq E_S(y') \quad (13)$$

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 $P_o(t) = P_o(t')$,

$$(\exists (x, y) \in \xi \times \eta((x_0, y_0), t))(\exists (x', y') \in \xi \times \eta((x_0, y_0), t'))\, E_{G \times S}(x, y) \cap E_G(x') \cap \Sigma' \not\subseteq E_S(y')$$

Since G is standardized, we get $< x > \in \xi'(< x_0 >, P(t))$ and $< x' > \in \xi'(< x_0 >, P(t'))$. We also have $y \in \eta(y_0, P(t))$ and $y' \in \eta(y_0, P(t'))$. Thus, $(< x >, y) \in \xi' \times \eta((< x_0 >, y_0), P(t))$ and $(< x' >, y) \in \xi' \times \eta((< x_0 >, y_0), P(t'))$. We also have that

$$E_{G \times S}(x, y) \cap E_G(x') \cap \Sigma' \not\subseteq E_S(y') \Rightarrow E_{(G/\approx_{\Sigma'}) \times S}(< x >, y) \cap E_{G/\approx_{\Sigma'}}(< x' >) \cap \Sigma' \not\subseteq E_S(y')$$

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

$$(< x >, y) \in \xi' \times \eta((< x_0 >, y_0), P(t)) \text{ and } (< x' >, y) \in \xi' \times \eta((< x_0 >, y_0), P(t'))$$

such that

$$E_{(G/\approx_{\Sigma'})\times S}(< x >, y) \cap E_{G/\approx_{\Sigma'}}(< x' >) \cap \Sigma' \nsubseteq E_S(y')$$

which contradicts expression (13). Therefore, (1) is true.
(2) Suppose $\Sigma_o \subseteq \Sigma'$. Let S be state observable w.r.t. G and P_o. Thus, for any $t, t' \in L(G \times S)$ with $P_o(t) = P_o(t')$,

$$(\forall (x,y) \in \xi \times \eta((x_0,y_0),t))(\forall (x',y') \in \xi \times \eta((x_0,y_0),t'))\, E_{G\times S}(x,y) \cap E_G(x') \cap \Sigma' \subseteq E_S(y') \tag{14}$$

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 $(< x >, y) \in \xi' \times \eta((< x_0 >, y_0), s)$ and $(< x' >, y') \in \xi' \times \eta((< x_0 >, y_0), s')$,

$$E_{(G/\approx_{\Sigma'})\times S}(< x >, y) \cap E_{G/\approx_{\Sigma'}}(< x' >) \cap \Sigma' \nsubseteq E_S(y') \tag{15}$$

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

$$E_{(G/\approx_{\Sigma'})\times S}(< x >, y) = \{\sigma \in \Sigma' | (\exists u \in (\Sigma - \Sigma')^*)(\exists \hat{x} \in \xi(x,u))\, \sigma \in E_{G\times S}(\hat{x}, y)\}$$

and

$$E_{G/\approx_{\Sigma'}}(< x' >) = \{\sigma \in \Sigma' | (\exists u' \in (\Sigma - \Sigma')^*)(\exists \hat{x}' \in \xi(x',u))\, \sigma \in E_G(\hat{x}')\}$$

Thus, from expression (15), there exist $u, u' \in (\Sigma - \Sigma')^*$ and $\hat{x} \in \xi(x,u)$ and $\hat{x}' \in \xi(x',u')$ such that

$$E_{G\times S}(\hat{x}, y) \cap E_G(\hat{x}') \cap \Sigma' \nsubseteq E_S(y')$$

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 $P_o(tu) = P_o(t'u')$,

$$(\exists (\hat{x}, y) \in \xi \times \eta((x_0,y_0), tu))(\exists (\hat{x}', y') \in \xi \times \eta((x_0,y_0), t'u'))\, E_{G\times S}(\hat{x}, y) \cap E_G(\hat{x}') \cap \Sigma' \nsubseteq E_S(y')$$

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

10. Proof of Lemma 3.9: (1) Let S be state normal w.r.t. $G/\approx_{\Sigma'}$ and P'_o. Then for any $s \in L((G/\approx_{\Sigma'}) \times S)$, $s' \in \overline{P'^{-1}_o(P'_o(s))} \cap L((G/\approx_{\Sigma'}) \times S)$, we have, for each $(< x >, y) \in \xi' \times \eta((< x_0 >, y_0), s')$ and $s'' \in \Sigma'^*$,

$$P'_o(s's'') = P'_o(s) \Rightarrow [\xi'(< x >, s'') \neq \varnothing \Rightarrow \eta(y, s'') \neq \varnothing] \tag{16}$$

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 $t'' \in \Sigma^*$,

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

Let $P''_o : \Sigma_o^* \to (\Sigma_o \cap \Sigma')^*$ be the naturel projection. Since G is standardized, we have

$$P(t) \in L((G/\approx_{\Sigma'}) \times S)$$

and from $(x, y) \in \xi \times \eta((x_0, y_0), t')$ we get $(< x >, y) \in \xi' \times \eta((< x_0 >, y_0), P(t'))$. Since $t' \in \overline{P_o^{-1}(P_o(t))} \cap L(G \times S)$, we get that

$$P_o(t') \leq P_o(t) \wedge P(t) \in L((G/\approx_{\Sigma'}) \times S)$$

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

$$P_o'(P(t't'')) = P_o''(P_o(t't'')) = P_o''(P_o(t)) = P_o'(P(t))$$

Since $\xi(x, t'') \neq \varnothing$, if $P(t'') = \epsilon$ we have $\xi'(< x >, \epsilon) \neq \varnothing$; if $P(t'') \neq \epsilon$ we have $\xi(< x >, P(t'')) \neq \varnothing$. 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 $(< x >, y) \in \xi' \times \eta((< x_0 >, y_0), s')$ and $s'' = p(t'') \in \Sigma'^*$,

$$P_o'(s's'') = P_o'(s) \wedge \xi'(< x >, s'') \neq \varnothing \wedge \eta(y, s'') = \varnothing$$

which contradicts expression (16). 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 $t'' \in \Sigma^*$,

$$P_o(t't'') = P_o(t) \Rightarrow [\xi(x, t'') \neq \varnothing \Rightarrow \eta(y, P(t'')) \neq \varnothing] \tag{17}$$

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

$$P_o'(s's'') = P_o'(s) \wedge \xi'(< x >, s'') \neq \varnothing \wedge \eta(y, s'') = \varnothing$$

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 $\xi'(< x >, s'') \neq \varnothing$, there exists $t'' \in \Sigma^*$ such that $\xi(x, t'') \neq \varnothing$. From $P_o'(s's'') = P_o'(s)$ we have $P_o(t't'') = P_o'(\overline{P(t't'')}) = P_o'(s's'') = P_o'(s) = P_o'(P(t)) = P_o(t)$. 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

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

which contradicts expression (17). Thus, (2) is true. ■

Bibliography

[1] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event systems. *SIAM J. Control and Optimization*, 25(1):206–230, 1987.

[2] W.M. Wonham and P.J. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Control and Optimization*, 25(3):637–659, 1987.

[3] W.M. Wonham and P.J. Ramadge. Modular supervisory control of discrete event systems. *Maths. of Control, Signals & Systems*, 1(1):13–30, 1988.

[4] M.H. de Queiroz and J.E.R. Cury. Modular supervisory control of composed systems. In *Proc. of American Control Conference*, pages 4051–4055, Chicago, USA, June 2000.

[5] R. Kumar and M. A. Shayman. Centralized and decentralized supervisory control of nondeterministic systems under partial observation. *SIAM Journal of Control and Optimization*, 35(2):363–383, 1997.

[6] K.C. Wong and W.M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*, 6(3):241–273, 1996.

[7] R.J. Leduc, B. A. Brandin, M. Lawford and W.M. Wonham. Hierarchical interface-based supervisory control-part I: serial case. *IEEE Trans. Automatic Control*, 50(9):1322–1335, 2005.

[8] R.J. Leduc, M. Lawford and W.M. Wonham. Hierarchical interface-based supervisory control-part II: parallel case. *IEEE Trans. Automatic Control*, 50(9):1336–1348, 2005.

[9] R.J. Leduc and P. Dai. Synthesis method for hierarchical interface-based supervisory control. In *Proc. 26th American Control Conference*, pages 4260–4267, New York City, USA, July, 2007

[10] C. Ma and W.M. Wonham. Nonblocking supervisory control of state tree structures. *IEEE Trans. Automatic Control*, 51(5):782–793, 2006.

[11] L. Feng and W.M. Wonham. Computationally efficient supervisor design: abstraction and modularity. In *Proc. 8th International Workshop on Discrete Event Systems (WODES06)*, pages 3–8, 2006.

[12] P.N. Pena, J.E.R. Cury and S. Lafortune. Testing modularity of local supervisors: an approach based on abstractions. In *Proc. 8th International Workshop on Discrete Event Systems (WODES06)*, pages 107–112, 2006.

[13] K. Schmidt, H. Marchand and B. Gaudin. Modular and decentralized supervisory control of concurrent discrete event systems using reduced system models. In *Proc. 8th International Workshop on Discrete Event Systems (WODES06)*, pages 149–154, 2006.

[14] K.C. Wong and W.M. Wonham. On the computation of observers in discrete-event systems. *Discrete Event Dynamic Systems*, 14(1):55-107, 2004.

[15] R. Su and J.G. Thistle. A distributed supervisor synthesis approach based on weak bisimulation. In *Proc. 8th International Workshop on Discrete Event Systems (WODES06)*, pages 64–69, 2006.

[16] W. M. Wonham. *Supervisory Control of Discrete-Event Systems*. Systems Control Group, Dept. of ECE, University of Toronto. URL: www.control.utoronto.ca/DES, July 1, 2007.

[17] R. Milner. Operational and algebraic semantics of concurrent processes. *Handbook of theoretical computer science (vol. B): formal models and semantics*, pp. 1201-1242, MIT Press, 1990

[18] A. Overkamp. Supervisory control using failure semantics and partial specifications. *IEEE Trans. Automatic Control*, 42(4):498-510, 1997.

[19] M. Heymann and F. Lin. Discrete event control of nondeterministic systems. *IEEE Trans. Automatic Control*, 43(1):3-17, 1998.

[20] C. Zhou, R. Kumar and S. Jiang. Control of nondeterministic discrete event systems for bisimulation equivalence. *IEEE Trans. Automatic Control*, 51(5):754-765, 2006.

[21] C. Zhou and R. Kumar. A small model theorem for bisimilarity control under partial observation. *IEEE Trans. Automation Science and Engineering*, 4(1):93-97, 2007.

[22] J.C. Fernandez. An implementation of an efficient algorithm for bisimulation equivalence. *Science of Computer Programming*, 13(2-3): 219-236, 1990

[23] M. Fabian and B. Lennartson. On non-deterministic supervisory control. In *Proc. 35th IEEE Conference on Decision and Control*, pages 2213–2218, 1996.

[24] H. Flordal and R. Malik. Modular nonblocking verification using conflict equivalence. In *Proc. 8th International Workshop on Discrete Event Systems (WODES06)*, pages 100–106, 2006.

[25] H. Flordal, R. Malik, M. Fabian and K. Akesson. Compositional synthesis of maximally permissive supervisors using supervisor equivalence. In *Discrete Event Dynamic Systems*, 17(4):475-504, 2007.

[26] R. Malik and H. Flordal. Yet another approach to compositional synthesis of discrete event systems. In *Proc. 9th International Workshop on Discrete Event Systems (WODES08)*, pages 16–21, 2008.

[27] R.C. Hill, D.M. Tilbury and S. Lafortune. Modular supervisory control with equivalence-based conflict resolution. In *Proc. 2008 American Control Conference (ACC08)*, pages 491–498, 2008.