博士学位論文

Maximally Permissive Similarity Control of Nondeterministic Discrete Event Systems

李　京倫

2022年 6月

大阪大学大学院工学研究科

Contents

List of Figures

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my thesis advisor, Prof. Shigemasa Takai at Division of Electrical, Electronic and Infocommunications Engineering, Graduate School of Engineering, Osaka University for providing me with valuable guidance in every stage of this research. It has been a great privilege and joy to study under his guidance and supervision.

My hearty appreciations go to Prof. Toshiyuki Miyamoto at Department of Information Systems, Graduate School and Faculty of Information Science and Technology, Osaka Institute of Technology, Assoc. Prof. Naoki Hayashi, at Division of Mathematical Science for Social Systems, Graduate School of Engineering Science, Osaka University, Assoc. Prof. Takeshi Hatanaka at Department of Systems and Control Engineering, School of Engineering, Tokyo Institute of Technology, and Asst. Prof. Kazumune Hashimoto at Division of Electrical, Electronic and Infocommunications Engineering, Graduate School of Engineering, Osaka University for their suggestions, feedback, and encouragement during my study.

I would like to express my sincere gratitude to Prof. Tomoo Ushio at Division of Electrical, Electronic and Infocommunications Engineering, Graduate School of Engineering, Osaka University for his careful review of this thesis. I also wish to express my cordial thanks to Prof. Tsuyoshi Funaki at Division of Electrical, Electronic and Infocommunications Engineering, Graduate School of Engineering, Osaka University for participating my dissertation committee.

I shall extend my thanks to the students from Takai laboratory, who participated in this study with great cooperation.

Last but not least, I would like to thank my family and all my friends for their encouragement and support.

Abstract

Over the past few decades, the control theory of discrete event systems has been successfully applied to provide formal treatments for many engineering systems. The basic supervisory control of discrete event systems aims to synthesize a maximally permissive supervisor (optimal controller) so that the system under control satisfies the given safety and nonblockingness (liveness) specification.

In the analysis and control of discrete event systems at a certain level of abstraction, nondeterminism may be introduced in the system and specification models when some lower-level details are omitted. In the nondeterministic setting, a notion of simulation relation is stronger than the conventional language inclusion relation since it specifies the universal constraints on the branching behavior in addition to the sequential behavior represented by a language.

This dissertation studies the supervisory control problem of enforcing similarity for nondeterministic discrete event systems so that the supervised system is simulated by the specification which denotes the safety requirement. First, we study the similarity control problem for partially observed discrete event systems, which extends an existing research studied for the full observation case. We develop a method to synthesize a maximally permissive supervisor that enforces similarity under partial observation. Then, we consider the control problem of both enforcing similarity and imposing nonblockingness. To solve the nonblocking similarity control problem, we provide an algorithm that computes a maximally permissive nonblocking supervisor from a possibly blocking one, which works for both fully and partially observed systems. An existence condition for a solution to the nonblocking similarity control problem is also derived. Finally, the modular control architecture is applied in the similarity control problem in two cases, where the supervisor synthesis suffers from exponential state space explosion: (i) the system is given as monolithic and the specification consists of a set

of elementary specifications, and (ii) the composite system consists of multiple subsystems, each of which is assigned a corresponding local subspecification. We identify the conditions under which the maximal permissiveness can be achieved by the modular control.

Chapter 1

Introduction

A cyber-physical system (CPS), which combines computation and communication entities with the physical world, is gathering attention in today's technological society. The growing demands for CPSs on safety, stability, privacy, and security motivate the development and utilization of formal techniques. Discrete Event Systems (DESs), which serve as an important class of the CPSs, are inherently event-driven systems whose discrete state evolution depends on the occurrence of asynchronous discrete events over time. For example, automated manufacturing systems, traffic systems, database systems, as well as computer and communication networks can be modeled as DESs.

To solve a logical control problem of DESs, Ramadge and Wonham introduced supervisory control in [1], so that a controller named supervisor can be synthesized from the models of the system and specification. The correctness of the obtained supervisor is guaranteed since the supervised system is proved to satisfy the specification. As verification and validation of the system are reduced, supervisor synthesis is an efficient and principled approach to handle DESs. Supervisory control theory has been widely studied in recent decades, as to be introduced later.

Supervisory control of DESs was implemented by utilizing the industrial technologies of Programmable Logic Controllers (PLCs) and Sequential Function Charts (SFCs) in [2, 3], and was successfully applied in the testbed assembly process of the Atelier Interétablissement Productique (AIP) [4] and flexible manufacturing [5]. In the fields of computer science and software engineering, supervisory control theory was applied for deadlock avoidance [6], concurrency control [7], and automated service composition [8]. Other applications of supervisory control theory span varieties of domains including the design of the directory assistance services of a call center [9], the control of theme park vehicles [10], a patient support system for a magnetic resonance image (MRI) scanner [11], chemical process control [12], the design for wafer logistics in lithography machines [13], advanced driver assistance systems [14], the control of a waterway lock and a bridge [15], and the control of a road tunnel [16].

1.1 Literature Review: Supervisory Control Theory

Supervisory control initiated by Ramadge and Wonham aims to achieve the given specification by disabling inappropriate events by a supervisor [1]. Under the framework of supervisory control formulated in [1], where both the system and the specification are represented as deterministic automata, a nonblocking supervisor is synthesized for the given system and the specification to achieve the equivalence of the generated or marked language of the supervised system and that of the specification. If a supervisor that exactly achieves the specification language does not exist, a nonblocking supervisor is synthesized for the supremal controllable and $L_m(G)$-closed subspecification (sublanguage of the specification language) [17].

Supervisory control under partial observation was initially studied in [18, 19]. It was shown that there exists a nonblocking supervisor for the given specification that works under partial observation if and only if the specification language is controllable, observable, and $L_m(G)$-closed. If these conditions are not satisfied, a problem of synthesizing a supervisor that achieves a subspecification (or equivalently, computing an achievable subspecification) arises. Since the observability property is not preserved under union in general, the optimal nonblocking supervisor that achieves the supremal subspecification does not necessarily exist under partial observation. Varieties of works have been done to synthesize a supervisor for partially observed DESs. In [18, 19], a normality property stronger than observability was introduced, which is preserved under union. The supremal controllable, normal, and $L_m(G)$-closed sublanguage can be computed using the results in [20–22]. Without imposing nonblockingness, researches [23–26] obtained solutions that are locally maximal, which means that there does not exist another solution that is strictly larger. Solutions possibly larger than the supremal controllable, normal, and $L_m(G)$-closed one were obtained in studies [27, 28], while their (local) maximal permissiveness is not guaranteed. The problem of synthesizing a maximally permissive nonblocking supervisor for partially observed systems was solved in [29]. A supervisor that achieves a maximal sublanguage that is controllable, observable, and $L_m(G)$-closed was synthesized based on a game structure named bipartite transition system. Nondeterministic control was introduced in the control of deterministic DESs in [30], and then a more generalized nondeterministic supervisor that allows silent transitions was provided in [31].

As the system and the specification usually consist of multiple components, a main challenge arises in the monolithic supervisor synthesis due to the fact that, if a system or specification model is given as the synchronous composition of multiple components, then its state number grows exponentially in the number of the components in the worst case (also known as exponential state space explosion) [32]. Given the system and a modular specification being composed of a set of elementary specifications, the control problem is modularly solved by a set of elementary supervisors which are respectively synthesized for these subspecifications [33]. Such elementary supervisors act as global supervisors. Then, the modular control was extended to the case where the system also consists of multiple smaller subsystems [34–37], in which each of the subsystems is controlled by the corresponding local supervisor. Thanks to the local synthesis, the space and time complexity of the modular control scheme is lower than that of the monolithic control scheme in general. Moreover, modular control also has an advantage that the partial changes of the system and the specification can be handled by only recomputing those relevant local supervisors. It was proved in [34] that, if any locally uncontrollable event is not shared by other subsystems, then the language equivalence of the system supervised by a maximally permissive monolithic supervisor and the one supervised by a set of local supervisors which are respectively maximally permissive with respect to the corresponding subsystems and subspecifications is achieved. On the other hand, blockingness may arise in the system under modular control since modular supervisors, regardless of whether they are global or local, are unaware of their global interaction [38]. A nonconflicting property that ensures the nonblockingness when multiple supervisors are combined was widely studied; see, for example, [39–41].

Nondeterminism of transitions may arise in the system and specification models because of unmodeled dynamics, abstraction, and lack of information. The control of nondeterministic systems has been studied for varieties of specifications: language [42–44], process [45], trajectory [46], μ-calculus [47, 48], CTL* temporal logic [49], and modal logic [50].

The notion of bisimulation equivalence introduced by [51, 52] has been widely applied to control theory as well as computer science. The bisimulation equivalence relation is stronger than the language equivalence relation since bisimulation equivalence specifies the exact branching behavior in addition to the sequential behavior captured by language equivalence. Hence, it is commonly used in the control of DESs where both the system and specification are given as nondeterministic.

Given the system and the specification represented as nondeterministic automata, a bisimilarity control problem is a problem of synthesizing a bisimilarity-enforcing supervisor such that the supervised system is bisimilar (bisimulation equivalent) to the specification. This problem was formulated under full and partial observation in [53] and [54], respectively, where a small model theorem is provided to show that a bisimilarity-enforcing supervisor exists if and only if such a supervisor over the power set of the Cartesian product of the states of the system and specification exists. An exhaustive search for verifying the existence condition (and supervisor synthesis) is doubly exponential in the numbers of states of the system and the specification [53]. In [55] and [56], new necessary and sufficient conditions for the existence of a supervisor that enforces bisimilarity that works under full and partial observation were derived, respectively, both of which can be verified in single exponential time with respect to the size of system and specification. See also [57] for a sufficient but not necessary condition for the existence of a bisimilarity-enforcing supervisor that can be polynomially verified. The bisimilarity control problems under specialized situations were studied in [58–61]. It was assumed that a supervisor can observe both the current state and event occurrence of the system in [58]. In [59], [60], and [61], the supervisor, the specification, and both of them were restricted to be deterministic, respectively.

If the above-mentioned existence condition for a bisimilarity-enforcing supervisor is not satisfied, then the bisimilarity control problem is not solvable for the given system and specification. A possible approach is to relax the requirement of enforcing bisimilarity. One may consider a similarity control problem which only requires us to synthesize a similarity-enforcing supervisor so that the supervised system is simulated by the specification. It is known that the simulation relation from the supervised system to the specification is stronger than the inclusion relations of their generated and marked languages.

The similarity control problem was studied in [62–65] without addressing the nonblocking issue. In [62], a range similarity control problem was studied, where a supervisor is synthesized so that the supervised system simulates the lower bound specification and is simulated by the upper bound specification. The permissiveness of the synthesized supervisor in [62] highly depends on how permissive the lower bound specification is. In [63], a maximally permissive similarity-enforcing supervisor was synthesized under state and event observation, where the assumption of state observation is restrictive in general. The similarity control problem that does not require state observation was studied in [64, 65], while the event occurrence is assumed to be fully observable. In particular, a maximal solution to the similarity control problem without state observation was proposed in [65]. The afore-mentioned maximally permissive similarity-enforcing supervisor is maximal in a global sense, which means that the system supervised by a maximally permissive supervisor simulates the system under the control of any other solution to the similarity control problem.

The nonblocking similarity control problem is a problem of synthesizing a nonblocking similarity-enforcing supervisor. Due to nondeterminism, imposing nonblockingness complicates the similarity control problem. To solve the problem, the assumption that the current state is observable was imposed in [66, 67]. Under such a restrictive assumption, a maximally permissive nonblocking similarity-enforcing supervisor is synthesized in [66, 67].

The similarity control problem for composite DESs with and without state observation was studied in [68] and [69], respectively. Here, the composite system and the entire specification consist of multiple nondeterministic components, and each subsystem is related to a corresponding local subspecification. Under the same assumption as [34] that any locally uncontrollable event is not shared by other subsystems, the validity of the modular control approach was proved in [68] and [69] by showing the bisimulation equivalence of the composite system controlled by a monolithic supervisor and the one controlled by a set of local supervisors. Without the restrictive requirement of state observation, [69] showed the equivalence of the local supervisors synthesized in the way of [64] and their monolithic counterpart. The maximal permissiveness of the modular control, however, was not guaranteed since the similarity-enforcing supervisor proposed by [64] is not necessarily maximally permissive.

1.2 Contributions and Organization of This Thesis

This dissertation studies several similarity control problems and provides maximal solutions to them. First, we consider the similarity control problem subject to partial observation. Here, the existing researches [64, 65] cannot deal with the presence of unobservable events. A supervisor that correctly estimates the current state of the partially observed nondeterministic system is synthesized, and it is proved to be a maximal solution to the similarity control problem. Second, we tackle the nonblocking similarity control problem, which remains open when the current state of the system cannot be observed. We develop an algorithm that can be used for verifying whether the problem is solvable and for computing a maximal solution to it. Third, we apply the modular control architecture to the similarity control problem in two cases, where the supervisor synthesis suffers from exponential state space explosion: (i) the system is given as monolithic and the specification consists of a set of elementary specifications, and (ii) the composite system consists of multiple subsystems, each of which is assigned a corresponding local subspecification. In both cases, we do not impose state observation and provide maximal modular solutions to the similarity control problems. In case (ii), the results are obtained under assumptions that are weaker than the one used in [68, 69]. Table 1.1 compares this study with previous work.

Table 1.1: Comparison of this dissertation and previous work

<table>
<tr><th colspan="2"></th><th>State and event observation</th><th>Full event observation</th><th>Partial event observation</th></tr>
<tr><td rowspan="2">Monolithic control</td><td>Does not impose non-blockingness</td><td>[63]</td><td>[64, 65]</td><td>Chapter 3</td></tr>
<tr><td>Imposes non-blockingness</td><td>[66, 67]</td><td colspan="2">Chapter 4</td></tr>
<tr><td rowspan="2">Modular control</td><td>Modular Specification</td><td>None</td><td colspan="2">Chapter 5</td></tr>
<tr><td>Composite system</td><td>[68]</td><td>[69], Chapter 6</td><td>None</td></tr>
</table>

This dissertation is organized as follows.

Chapter 2 reviews some basic notions that are needed in this dissertation. We mainly introduce the definitions of Nondeterministic Finite Automata (NFAs), the notions of simulation and bisimulation relations, and the control framework of nondeterministic supervisors. Besides, some related notions and properties, as well as discussions on nondeterministic supervisory control are presented.

Chapter 3 handles the control problem of enforcing similarity for the nondeterministic system and specification where only a part of the events are observable. This problem requires us to synthesize an admissible supervisor so that the supervised system is simulated by the specification. We begin the chapter with an example that motivates the enforcement of similarity. Then, we provide a necessary and sufficient condition for the existence of such a similarity-enforcing supervisor. When this condition holds, such a supervisor that is (globally) maximally permissive is synthesized. Finally, we provide a technique that removes a certain set of redundant states from the synthesized supervisor without reducing its permissiveness.

Chapter 4 solves the nonblocking similarity control problem, where both enforcing similarity and imposing nonblockingness are required. For the given nondeterministic system and specification, we synthesize a nonblocking supervisor that enforces similarity. In order to construct such a supervisor, we propose an algorithm that computes a nonblocking supervisor from a possibly blocking one by iteratively removing certain states. This algorithm is proved to be valid under both full and partial observation. Then, we identify two key properties of input supervisors, named state-unmergedness and strong maximal permissiveness, which together guarantee the maximal permissiveness of output nonblocking supervisors. The algorithm is applied to the supervisors introduced in Chapter 3, which have these two properties, to obtain a maximally permissive nonblocking supervisor. In addition, we show that a nonblocking supervisor is generated by the algorithm if and only if there exists a solution to the nonblocking similarity control problem.

In Chapter 5, we consider the modular similarity control problem for the monolithic system and the modular specification that consists of multiple elementary specifications. This problem is required to be solved modularly without constructing the entire specification. The modular control architecture is applied, where, for each elementary specification, an elementary supervisor that enforces similarity is synthesized. We show that the similarity control problem with the modular specification is modularly solvable if it is monolithically solvable. Besides, we show that the elementary supervisors together achieve maximal permissiveness with respect to the modular specification if each of them is maximally permissive with respect to the corresponding elementary specification. The results of this chapter can be applied in both full and partial observation.

In Chapter 6, we consider the modular similarity control problem for the composite system under full observation. The composite system under consideration consists of multiple local subsystems, each of which is assigned a nondeterministic local subspecification. To modularly solve the similarity control problem, we synthesize a local supervisor for each subsystem and its corresponding local subspecification, so that the entire specification is modularly achieved. We provide a condition under which the modular control is applicable, which is relaxed compared to the one used in [68, 69]. Finally, we show that the same permissiveness as a maximally permissive monolithic supervisor can be achieved by these local supervisors when the above-mentioned condition is satisfied.

We conclude this dissertation and discuss some directions of future research in Chapter 7.

Chapter 2

Nondeterministic Discrete Event Systems

In this chapter, we review several basic notions that will be used in this dissertation. We first show the definition of an NFA, which models a DES at the logical level. Then, we introduce the languages generated and marked by an NFA, and the synchronous composition of NFAs. In addition, we introduce the definitions of simulation relation and bisimulation relation. Finally, we review the nondeterministic supervisory control framework (without state observation) that is used throughout this dissertation.

2.1 System Model

An NFA, which we use as a model of a DES in this dissertation, is a 5-tuple

$$G = (X, \Sigma, \alpha, X_0, X_m),$$

where

- X is the finite set of states,
- Σ is the finite set of events,
- $\alpha : X \times \Sigma \to 2^X$ is the nondeterministic state transition function,
- $X_0 \subseteq X$ is the nonempty set of initial states, and

- $X_m \subseteq X$ is the set of marked states, which represent, for example, the completion of tasks.

Given $x, x' \in X$ and $\sigma \in \Sigma$, $x' \in \alpha(x, \sigma)$ denotes that a transition from state x to state x' is caused by the occurrence of event σ, and $\alpha(x, \sigma) = \emptyset$ means that σ is not possible at x. We say that σ is defined at x if $\alpha(x, \sigma) \neq \emptyset$. The set $\mathcal{A}(\Sigma)$ denotes the collection of all NFAs over the event set Σ.

Example 2.1 We consider an NFA $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ shown in Figure 2.1 that models a simple processing machine. The state set X, initial state set X_0, and marked state set X_m are given as $X = \{x_0, x_1, x_2, x_3\}$, $X_0 = \{x_0\}$, and $X_m = \{x_3\}$, respectively. The initial state x_0, which is identified by $\downarrow$ to a circle in Figure 2.1, represents the standby state of the machine. States x_1 and x_2 indicate that the machine is operating and out of order, respectively. The marked state x_3 means that the machine completes the operation, and it is identified by a double circle in Figure 2.1. Events σ_1, σ_2, and σ_3 in $\Sigma = \{\sigma_1, \sigma_2, \sigma_3\}$ represent the start of operation, failure recovery, and completion of operation, respectively. At state x_0, event σ_1 causes a transition to x_1 or x_2 nondeterministically. The transition function $\alpha : X \times \Sigma \to 2^X$ is described as

$$\alpha(x, \sigma) = \begin{cases} \{x_1, x_2\}, & \text{if } x = x_0, \sigma = \sigma_1 \\ \{x_1\}, & \text{if } x = x_2, \sigma = \sigma_2 \\ \{x_3\}, & \text{if } x = x_1, \sigma = \sigma_3 \\ \emptyset, & \text{otherwise} \end{cases}$$

for each $x \in X$ and each $\sigma \in \Sigma$.

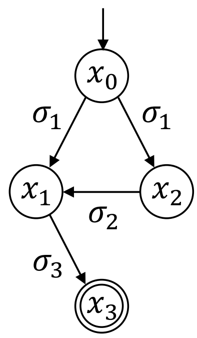


Figure 2.1: An NFA G that models a processing machine.

For an arbitrary set A, $|A|$ denotes its cardinality. A Deterministic Finite Automaton (DFA) can be considered as a special case of an NFA. Formally, for a DFA $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$, it holds that $|X_0| = 1$ and $|\alpha(x, \sigma)| \leq 1$ for each $x \in X$ and each $\sigma \in \Sigma$.

Let Σ^* be the collection of all finite event strings of elements in Σ, including the empty string ε. A subset L of Σ^* is called a language over Σ. For each $s \in \Sigma^*$, $|s|$ denotes its length, and $|\varepsilon| = 0$ holds. $tu \in \Sigma^*$ is the concatenation of t and u, where t and u are strings in Σ^*. It holds that $t\varepsilon = \varepsilon t = t$. The concatenation L_1L_2 of $L_1, L_2 \subseteq \Sigma^*$ is given as $L_1L_2 = \{s_1s_2 \in \Sigma^* \mid s_1 \in L_1 \wedge s_2 \in L_2\}$.

For $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$, we extend the transition function α to $\alpha : X \times \Sigma^* \to 2^X$ inductively as

- $\forall x \in X : \alpha(x, \varepsilon) = \{x\}$, and
- $\forall x \in X, \forall s \in \Sigma^*, \forall \sigma \in \Sigma : \alpha(x, s\sigma) = \bigcup_{x' \in \alpha(x,s)} \alpha(x', \sigma)$.

Additionally, for any $X' \subseteq X$ and any $s \in \Sigma^*$, we let

$$\alpha(X', s) = \bigcup_{x' \in X'} \alpha(x', s).$$

The language generated by $G \in \mathcal{A}(\Sigma)$ is defined as $L(G) = \{s \in \Sigma^* \mid \alpha(X_0, s) \neq \emptyset\}$, which is the set of all possible event strings of G. The marked language $L_m(G) = \{s \in L(G) \mid \alpha(X_0, s) \cap X_m \neq \emptyset\}$ is the set of event strings that lead to a marked state. We let $\mathit{Reach}(G) = \{x \in X \mid \exists s \in L(G) : x \in \alpha(X_0, s)\}$ be the set of all reachable states of G. Given an arbitrary reachable state $x \in \mathit{Reach}(G)$, the language that leads to it from an initial state is defined as $L_x(G) = \{s \in L(G) \mid x \in \alpha(X_0, s)\}$.

G is said to be nonblocking if

$$\forall x \in \mathit{Reach}(G), \exists s \in \Sigma^* : \alpha(x, s) \cap X_m \neq \emptyset, \tag{2.1}$$

which guarantees that G can reach a marked state by the execution of an event string from any reachable state. In this dissertation, an NFA might also be denoted as a 4-tuple $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ when the issue of nonblockingness is out of consideration.

Due to nondeterminism, an NFA $G \in \mathcal{A}(\Sigma)$ is not necessarily nonblocking even if it satisfies the conventional nonblocking condition $\overline{L_m(G)} = L(G)$ defined for deterministic automata, where $\overline{L_m(G)} = \{s \in \Sigma^* \mid \exists t \in \Sigma^* : st \in L_m(G)\}$ denotes the prefix-closure of $L_m(G)$. That is, condition (2.1) is stronger than the one $\overline{L_m(G)} = L(G)$ in the nondeterministic setting.

For a set of n NFAs $G_i \in \mathcal{A}(\Sigma_i)$ ($i \in \{1, \ldots, n\}$), the operation of synchronous composition is defined in the following manner.

Definition 2.1 Given $G_i = (X_i, \Sigma_i, \alpha_i, X_{i,0}, X_{i,m}) \in \mathcal{A}(\Sigma_i)$ ($i \in \{1, \ldots, n\} = I$), their synchronous composition

$$\|_{i\in I} G_i = G_1 \| \ldots \| G_n = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$$

is defined as $X = \prod_{i\in I} X_i = X_1 \times \cdots \times X_n$, $\Sigma = \bigcup_{i\in I} \Sigma_i$, $X_0 = \prod_{i\in I} X_{i,0}$, and $X_m = \prod_{i\in I} X_{i,m}$. The nondeterministic transition function $\alpha : X \times \Sigma \to 2^X$ is given as

$$\alpha((x_1, \ldots, x_n), \sigma) = \begin{cases} \prod_{i\in I} X'_i, & \text{if } \forall i \in I(\sigma) : \alpha_i(x_i, \sigma) \neq \emptyset \\ \emptyset, & \text{otherwise} \end{cases}$$

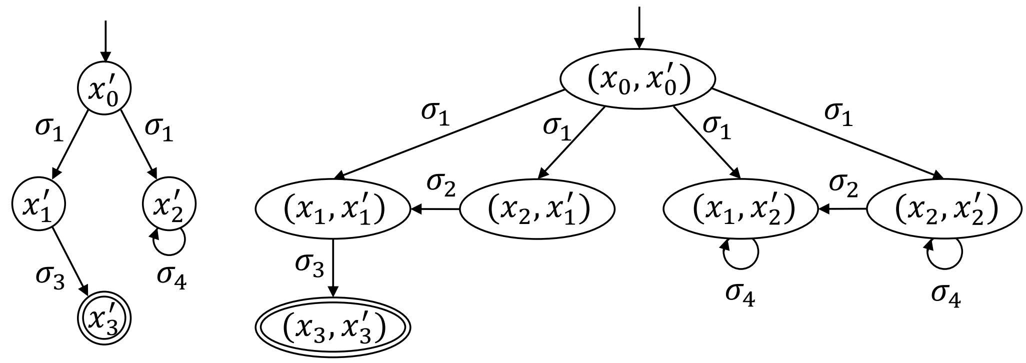
for each $(x_1, \ldots, x_n) \in X$ and each $\sigma \in \Sigma$, where $I(\sigma) = \{i \in I \mid \sigma \in \Sigma_i\}$ and

$$X'_i = \begin{cases} \alpha_i(x_i, \sigma), & \text{if } i \in I(\sigma) \\ \{x_i\}, & \text{otherwise} \end{cases}$$

for each $i \in I$.

Example 2.2 For $G \in \mathcal{A}(\Sigma)$ shown in Figure 2.1, it holds that $\mathit{Reach}(G) = X$ and $L(G) = \{\varepsilon, \sigma_1, \sigma_1\sigma_2, \sigma_1\sigma_3, \sigma_1\sigma_2\sigma_3\}$. Since the marked state x_3 is reached from the initial state x_0 after executing $\sigma_1\sigma_3$ or $\sigma_1\sigma_2\sigma_3$, we have $L_{x_3}(G) = L_m(G) = \{\sigma_1\sigma_3, \sigma_1\sigma_2\sigma_3\}$.

Consider another NFA $G' = (X', \Sigma', \alpha', X'_0, X'_m) \in \mathcal{A}(\Sigma')$ shown in Figure 2.2 (a), where $\Sigma' = \{\sigma_1, \sigma_3, \sigma_4\}$. The reachable part of the synchronous composition $G\|G'$ is shown in Figure 2.2 (b).



(a) **(b)**

Figure 2.2: (a) Automaton G' and (b) synchronous composition $G \| G'$.

2.2 Simulation Relation and Bisimulation Relation

Languages describe the sequential behavior of DESs, and they are generally used in the behavioral comparison of deterministic automata. On the other hand, nondeterminism in an automaton can capture aspects of the system behavior beyond the languages generated and marked by the automaton [32]. Instead of language inclusion and equivalence relations, we consider stronger notions of simulation relation and bisimulation relation defined in [51, 52] among nondeterministic automata.

A simulation relation is defined for NFAs as follows.

Definition 2.2 [53] For $G_i = (X_i, \Sigma, \alpha_i, X_{i,0}, X_{i,m}) \in \mathcal{A}(\Sigma)$ $(i \in \{1, 2\})$, a relation $\Phi \subseteq X_1 \times X_2$ is said to be a simulation relation from G_1 to G_2 if

- $\forall x_{1,0} \in X_{1,0}, \exists x_{2,0} \in X_{2,0} : (x_{1,0}, x_{2,0}) \in \Phi$,
- $\forall (x_1, x_2) \in \Phi, \forall \sigma \in \Sigma, \forall x_1' \in \alpha_1(x_1, \sigma), \exists x_2' \in \alpha_2(x_2, \sigma) : (x_1', x_2') \in \Phi$, and
- $\forall (x_1, x_2) \in \Phi : x_1 \in X_{1,m} \Rightarrow x_2 \in X_{2,m}$.

If there exists a simulation relation from G_1 to G_2, G_1 is said to be simulated by G_2, and we write $G_1 \sqsubseteq G_2$.

We drop the third condition

$$\forall (x_1, x_2) \in \Phi : x_1 \in X_{1,m} \Rightarrow x_2 \in X_{2,m}$$

on marked states where the issue of marking (nonblockingness) is not considered.

If $G_1, G_2 \in \mathcal{A}(\Sigma)$ simulate each other, namely, $G_1 \sqsubseteq G_2$ and $G_2 \sqsubseteq G_1$ hold, they are said to be simulation equivalent and we write as $G_1 \sim G_2$ [62].

Remark 2.1 [53, 62] For two NFAs $G_1, G_2 \in \mathcal{A}(\Sigma)$, we only have

$$G_1 \sqsubseteq G_2 \Rightarrow L(G_1) \subseteq L(G_2) \wedge L_m(G_1) \subseteq L_m(G_2),$$

and the reverse relation does not hold in general. In other words, the simulation relation is stronger than the inclusion relation of the generated and marked languages in the nondeterministic setting. Also note that $G_1 \sim G_2$ implies $L(G_1) = L(G_2)$ and $L_m(G_1) = L_m(G_2)$.

Example 2.3 We consider four automata $G_i = (X_i, \Sigma, \alpha_i, X_{i,0}, X_{i,m}) \in \mathcal{A}(\Sigma)$ ($i \in \{1, 2, 3, 4\} = I$) shown in Figures 2.3 (a), (b), (c), and (d), respectively. For each $i \in I$, it holds that $L(G_i) = \{\varepsilon, \sigma_1, \sigma_1\sigma_2, \sigma_1\sigma_3\}$ and $L_m(G_i) = \{\sigma_1\sigma_2, \sigma_1\sigma_3\}$. Since the relation $\Phi_{1,2} \subseteq X_1 \times X_2$ given as

$$\Phi_{1,2} = \{(x_{1,0}, x_{2,0}), (x_{1,1}, x_{2,1}), (x_{1,2}, x_{2,1}), (x_{1,3}, x_{2,3}), (x_{1,4}, x_{2,4})\}$$

is a simulation relations from G_1 to G_2, we have $G_1 \sqsubseteq G_2$. It also holds that $L(G_1) \subseteq L(G_2)$ and $L_m(G_1) \subseteq L_m(G_2)$. On the other hand, by the occurrence of σ_1, G_2 can reach $x_{2,1}$, where σ_2 and σ_3 are both possible. Since there does not exist a counterpart state of $x_{2,1}$ in G_1, G_2 is not simulated by G_1, even though $L(G_2) \subseteq L(G_1)$ and $L_m(G_2) \subseteq L_m(G_1)$. Hence, the simulation relation is stronger than the inclusion relation of languages in the nondeterministic setting.

Besides, it can be verified that $G_1 \sim G_4$ and $G_2 \sim G_3$.

For any relation $\Phi \subseteq X_1 \times X_2$, its inverse relation $\Phi^{-1} \subseteq X_2 \times X_1$ is defined as

$$\Phi^{-1} = \{(x_2, x_1) \in X_2 \times X_1 \mid (x_1, x_2) \in \Phi\}.$$

A bisimulation relation for two NFAs is defined as follows.

Definition 2.3 [53] For $G_i = (X_i, \Sigma, \alpha_i, X_{i,0}, X_{i,m}) \in \mathcal{A}(\Sigma)$ ($i \in \{1, 2\}$), a relation $\Phi \subseteq X_1 \times X_2$ is said to be a bisimulation relation of G_1 and G_2 if Φ is a simulation relation from G_1 to G_2 and Φ^{-1} is a simulation relation from G_2 to G_1. If there exists a bisimulation

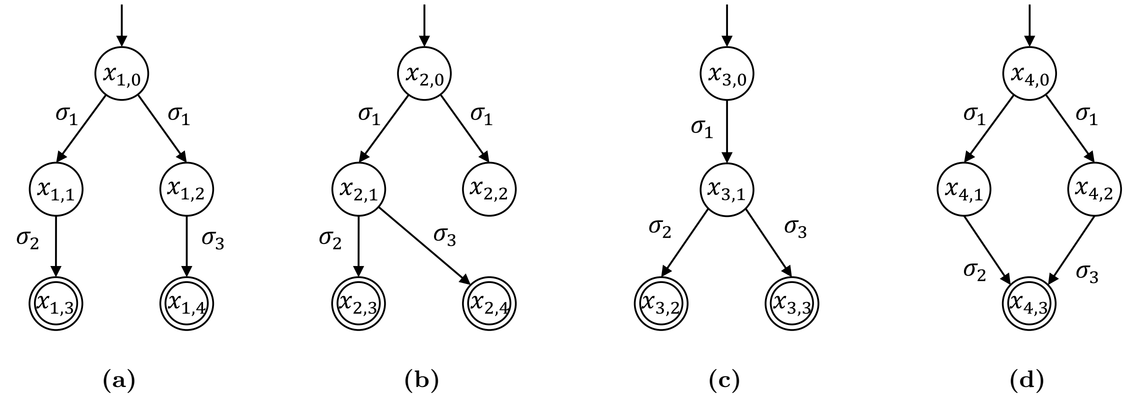


Figure 2.3: Automata (a) G_1, (b) G_2, (c) G_3, and (d) G_4.

relation of G_1 and G_2, they are said to be bisimilar (or bisimulation equivalent), and we write $G_1 \simeq G_2$.

Example 2.4 Recall $G_i = (X_i, \Sigma, \alpha_i, X_{1,0}, X_{i,m}) \in \mathcal{A}(\Sigma)$ $(i \in I)$ shown in Figures 2.3 (a), (b), (c), and (d), respectively. The relation $\Phi_{1,4} \subseteq X_1 \times X_4$ given as

$$\Phi_{1,4} = \{(x_{1,0}, x_{4,0}), (x_{1,1}, x_{4,1}), (x_{1,2}, x_{4,2}), (x_{1,3}, x_{4,3}), (x_{1,4}, x_{4,3})\}$$

and its inverse relation $\Phi_{1,4}^{-1} \subseteq X_4 \times X_1$ are simulation relations from G_1 to G_4 and G_4 to G_1, respectively. Then, $\Phi_{1,4}$ is a bisimulation relation of G_1 and G_4, which implies $G_1 \simeq G_4$. On the other hand, there does not exist a simulation relation from G_2 to G_3 whose inverse relation is a simulation relation from G_3 to G_2, $G_2 \simeq G_3$ does not hold and we only have $G_2 \sim G_3$.

Remark 2.2 The order $\sqsubseteq$ serves as a preorder for $\mathcal{A}(\Sigma)$ [62]. In other words, the order $\sqsubseteq$ on $\mathcal{A}(\Sigma)$ is

- transitive (for any $G_1, G_2, G_3 \in \mathcal{A}(\Sigma)$, if $G_1 \sqsubseteq G_2$ and $G_2 \sqsubseteq G_3$, then $G_1 \sqsubseteq G_3$),
- reflexive (for any $G \in \mathcal{A}(\Sigma)$, $G \sqsubseteq G$), but
- not antisymmetric (given $G_1, G_2 \in \mathcal{A}(\Sigma)$ such that $G_1 \sqsubseteq G_2$ and $G_2 \sqsubseteq G_1$, it is not necessarily $G_1 = G_2$).

The relation $\simeq$ on $\mathcal{A}(\Sigma)$ is also transitive and reflexive.

Moreover, we have the following propositions on the order $\sqsubseteq$, which will be used later.

Proposition 2.1 For $G_i = (X_i, \Sigma, \alpha_i, X_{i,0}, X_{i,m}) \in \mathcal{A}(\Sigma)$ ($i \in I = \{1, \ldots, n\}$), the following two statements hold for their synchronous composition $\|_{i\in I} G_i$.

1. For each $i \in I$, we have $\|_{i\in I} G_i \sqsubseteq G_i$.

2. For any $G' \in \mathcal{A}(\Sigma)$ such that $G' \sqsubseteq G_i$ holds for each $i \in I$, we have $G' \sqsubseteq \|_{i\in I} G_i$.

Proposition 2.2 For $G_i = (X_i, \Sigma, \alpha_i, X_{i,0}, X_{i,m}) \in \mathcal{A}(\Sigma)$ ($i \in I = \{1, \ldots, n\}$) and $G'_i = (X'_i, \Sigma, \alpha'_i, X'_{i,0}, X'_{i,m}) \in \mathcal{A}(\Sigma)$ ($i \in I$) such that $G_i \sqsubseteq G'_i$ holds for each $i \in I$, it holds that $\|_{i\in I} G_i \sqsubseteq \|_{i\in I} G'_i$.

Propositions 2.1 and 2.2 with $n = 2$ are proved in Theorem 2 and Lemma 1 of [62], respectively. Their extensions to the case of $n > 2$ are straightforward by the definitions of the synchronous composition and the simulation relation.

2.3 Nondeterministic Supervisory Control Framework

Given an uncontrolled DES that is modeled as an automaton, its behavior is modified in order to achieve a certain specification. A supervisor modifies the system's behavior by dynamically enabling/disabling the events of the system [1]. In this dissertation, we use $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$ to denote the system and the specification, respectively. A supervisor is also modeled as an NFA $S = (Y, \Sigma, \beta, Y_0, Y_m = Y) \in \mathcal{A}(\Sigma)$ over Σ, whose states are all marked. The system G being supervised by a supervisor S is given as their synchronous composition

$$S\|G = (Y \times X, \Sigma, \xi, Y_0 \times X_0, Y \times X_m) \in \mathcal{A}(\Sigma). \tag{2.2}$$

As S and G share the same event set Σ, we have $\xi((y, x), \sigma) = \beta(y, \sigma) \times \alpha(x, \sigma)$ for each $(y, x) \in Y \times X$ and each $\sigma \in \Sigma$, meaning that $\xi((y, x), \sigma) \neq \emptyset$ if and only if both $\beta(y, \sigma)$ and $\alpha(x, \sigma)$ are nonempty.

In general, a supervisor works under limited control and observation capabilities. The limitation on control is referred to the fact that some events cannot be disabled by a supervisor (for example, an event representing "fault"). The event set Σ is partitioned into two disjoint sets Σ_c and Σ_{uc}, which denote the set of controllable events and the set of uncontrollable events, respectively. An uncontrollable event in Σ_{uc} cannot be disabled by a supervisor.

The limitation on observation arises due to the limitation of the sensors, which are used for the observation on event occurrence and current state of the system. Similarly, Σ is also partitioned into two disjoint sets Σ_o and Σ_{uo}, which denote the set of observable events and the set of unobservable events, respectively. A supervisor observes the event occurrence through the natural projection $P : \Sigma^* \to \Sigma_o^*$ defined inductively as

- $P(\varepsilon) = \varepsilon$, and
- $\forall s \in \Sigma^*, \forall \sigma \in \Sigma : P(s\sigma) = \begin{cases} P(s)\sigma, & \text{if } \sigma \in \Sigma_o \\ P(s), & \text{if } \sigma \in \Sigma_{uo}. \end{cases}$

The inverse projection $P^{-1} : \Sigma_o^* \to 2^{\Sigma^*}$ is defined as $P^{-1}(t) = \{s \in \Sigma^* \mid P(s) = t\}$ for each $t \in \Sigma_o^*$.

Then, a notion of admissibility is imposed on a supervisor to guarantee that: (i) the occurrence of uncontrollable events in the system is not prevented by a supervisor; and (ii) a supervisor does not react to the occurrence of any unobservable event.

Definition 2.4 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$, a supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ is said to be admissible if it satisfies

- $\forall (y, x) \in \mathit{Reach}(S\|G), \forall \sigma \in \Sigma_{uc} : \alpha(x, \sigma) \neq \emptyset \Rightarrow \beta(y, \sigma) \neq \emptyset$ [64], and
- $\forall (y, x) \in \mathit{Reach}(S\|G), \forall \sigma \in \Sigma_{uo} : \beta(y, \sigma) \neq \emptyset \Rightarrow \beta(y, \sigma) = \{y\}$ [54].

When the events of the system under consideration are fully observed (namely, $\Sigma_{uo} = \emptyset$ holds and the second condition is automatically satisfied), we call an admissible supervisor Σ_{uc}-admissible for distinction.

The control mechanism of an admissible supervisor S for the system G can be explained as follows. Let $(y, x) \in \mathit{Reach}(S\|G)$ be the current state of $S\|G$. An event $\sigma \in \Sigma$ is enabled by S if $\beta(y, \sigma) \neq \emptyset$. We suppose that an enabled event σ occurs in G and causes a transition from x to $x' \in \alpha(x, \sigma)$. If $\sigma \in \Sigma_{uo}$, S does not react to σ because of the second condition of admissibility, and the resulting state of $S\|G$ is $(y, x') \in \mathit{Reach}(S\|G)$. If $\sigma \in \Sigma_o$, S observes the occurrence of σ and chooses the next state y' nondeterministically from $\beta(y, \sigma)$. The resulting state of $S\|G$ is $(y', x') \in \mathit{Reach}(S\|G)$. Since all transitions specified by $\alpha(x, \sigma)$ can occur in $S\|G$, S does not affect the nondeterminism of transitions in G. Since all the states in Y of S are marked, a state (y, x) of $S\|G$ is marked if and only if x is marked in G. A supervisor S is said to be nonblocking if $S\|G$ is nonblocking.

Remark 2.3 For practical implementation, a nondeterministic supervisor needs to nondeterministically select the control action (the initial state and the next state when an observable event is observed) online, where a "coin toss" mechanism with as many possible outcomes as the number of control action choices can be used [31].

The current state of the system is not required to be observed in this dissertation. Hereafter, we refer to full event observation and partial event observation as full observation and partial observation, respectively.

Remark 2.4 The control of nondeterministic DESs under state (and event) observation was studied in [63, 66–68], where for the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$, a supervisor that can observe the current state of G can be expressed by a pair (S, Ψ) of an automata $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ and a relation $\Psi \subseteq Y \times X$. A modified version of the synchronous composition

$$S\|_\Psi G = (Y \times X, \Sigma, \xi_\Psi, (Y_0 \times X_0) \cap \Psi, Y \times X_m) \in \mathcal{A}(\Sigma)$$

is used to model the controlled system, where the nondeterministic transition function ξ_Ψ : $(Y \times X) \times \to 2^{Y \times X}$ is given as

$$\xi_\Psi((y, x), \sigma) = (\beta(y, \sigma) \times \alpha(x, \sigma)) \cap \Psi. \tag{2.3}$$

for each $(y, x) \in Y \times X$ and each $\sigma \in \Sigma$. The initial state set $(Y_0 \times X_0) \cap \Psi$ means that, when the system G starts up with an initial state $x_0 \in X_0$, the corresponding initial state y_0 of S is not "fully" nondeterministically chosen from Y_0. Instead, y_0 is chosen from those states in Y_0 which satisfy $(y_0, x_0) \in \Psi$, based on the fact that S exactly knows that the current state of G is x_0. The same treatment is executed when a transition by an event $\sigma \in \Sigma$ is caused, as defined in (2.3). In addition, the following two conditions are imposed on (S, Ψ) to guarantee that the nondeterminism of transitions in G is not affected by (S, Ψ):

- $\forall x_0 \in X_0, \exists y_0 \in Y_0 : (x_0, y_0) \in \Psi$, and
- $\forall (y, x) \in \mathit{Reach}(S\|_\Psi G) : \beta(y, \sigma) \neq \emptyset \Rightarrow [\forall x' \in \alpha(x, \sigma), \exists y' \in \beta(y, \sigma) : (y', x') \in \xi_\Psi((y, x), \sigma)]$.

Given the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and a supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$, we also define the following notations. For any state $y \in Y$, $\Sigma_{uo}(y) = \{\sigma \in \Sigma_{uo} \mid \beta(y, \sigma) \neq \emptyset\}$ is the set of unobservable events enabled at y, and $C(y) = \{x \in X \mid (y, x) \in Reach(S \| G)\}$ is the set of states of G which are in the charge of y.

For any reachable state $y \in Reach(S)$ of an admissible supervisor $S \in \mathcal{A}(\Sigma)$, there exists an observable event string $s \in \Sigma_o^* \cap L(S)$ such that $y \in \beta(Y_0, s)$. Then, there exist states $y_0, y_1, \ldots, y_n \in Reach(S)$ with $y_0 \in Y_0$, $y_n = y$, and $y_{i+1} \in \beta(y_i, \sigma_i)$ for $i = 0, 1, \ldots, n-1$, where $s = \sigma_0 \sigma_1 \ldots \sigma_{n-1}$. For a series of state transitions $y_0 \xrightarrow{\sigma_0} y_1 \xrightarrow{\sigma_1} \ldots \xrightarrow{\sigma_{n-1}} y_n$ that finally reaches $y_n = y$, we call $\gamma_0 \sigma_0 \ldots \gamma_{n-1} \sigma_{n-1} \gamma_n \in 2^{\Sigma_{uo}}(\Sigma_o 2^{\Sigma_{uo}})^*$ a run in S that reaches y, where $\gamma_j = \Sigma_{uo}(y_j)$ $(j = 0, 1, \ldots, n)$. We let $\mathcal{R}_y(S)$ denote the set of all runs in S which reach y. Note that, when $y \in Y_0$, since $y \in \beta(Y_0, \varepsilon)$, the run $\Sigma_{uo}(y) \in 2^{\Sigma_{uo}}$ is an element of $\mathcal{R}_y(S)$. A run $r = \gamma_0 \sigma_0 \ldots \gamma_{n-1} \sigma_{n-1} \gamma_n \in \mathcal{R}_y(S)$ records not only the observable event string $\sigma_0 \sigma_1 \ldots \sigma_{n-1} \in \Sigma_o^* \cap L(S)$ that leads to y but also the control decisions $\gamma_0, \gamma_1, \ldots, \gamma_n \in 2^{\Sigma_{uo}}$ on unobservable events that S made. Here, the set of event strings that lead to y via r is given as $L(r, S) = \{s_0 \sigma_0 \ldots s_{n-1} \sigma_{n-1} s_n \in L(S) \mid \forall i \in \{0, 1, \ldots, n\} : s_i \in \gamma_i^*\} \subseteq L(S)$, which is also the set of strings enabled by S when its current state is reached via r. Note that $L(r, S)$ is uniquely determined by r. We suppose that r is shared by another supervisor $S' \in \mathcal{A}(\Sigma)$, namely, there exists a state $y' \in Reach(S')$ such that $r \in \mathcal{R}_{y'}(S')$. Then, it holds that $L(r, S) = L(r, S')$. Hence, for the sake of simplicity, we write $L(r, S)$ as $L(r)$ with a slight abuse of the notation.

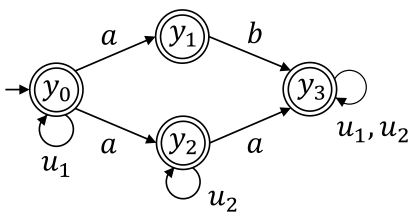


Figure 2.4: Supervisor S.

Example 2.5 Let $S \in \mathcal{A}(\Sigma)$ shown in Figure 2.4 be a supervisor for some system $G \in \mathcal{A}(\Sigma)$, where $\Sigma = \{a, b, u_1, u_2\}$ and $\Sigma_{uo} = \{u_1, u_2\}$. We consider state $y_3 \in Reach(S)$, which can be reached as $y_0 \xrightarrow{a} y_1 \xrightarrow{b} y_3$ or $y_0 \xrightarrow{a} y_2 \xrightarrow{a} y_3$. It follows that $\mathcal{R}_{y_3}(S) = \{r_1, r_2\}$, where $r_1 = \{u_1\}a\emptyset b\{u_1, u_2\}$ and $r_2 = \{u_1\}a\{u_2\}a\{u_1, u_2\}$. Then, we have $L(r_1) = \{s_1 a b s_1' \in L(S) \mid s_1 \in \{u_1\}^* \wedge s_1' \in \{u_1, u_2\}^*\}$ and $L(r_2) = \{s_2 a s_2' a s_2'' \in L(S) \mid s_2 \in \{u_1\}^* \wedge s_2' \in \{u_2\}^* \wedge s_2'' \in$

$\{u_1, u_2\}^*\}$.

In a special case where $\Sigma_{uo} = \emptyset$, namely, the system is fully observed, a run $r = \gamma_0\sigma_0\gamma_1\sigma_1 \ldots \gamma_{n-1}\sigma_{n-1}\gamma_n = \emptyset\sigma_0\emptyset\sigma_1 \ldots \emptyset\sigma_{n-1}\emptyset$ can be characterized by the string $\sigma_0\sigma_1 \ldots \sigma_{n-1}$, and, since $\emptyset^* = \{\varepsilon\}$, it holds that $L(r) = \{\sigma_0\sigma_1 \ldots \sigma_{n-1}\}$.

We define a class of supervisors, named $\Sigma_{uc} \cap \Sigma_{uo}$-compatible supervisors, in the following manner.

Definition 2.5 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$, a supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ is said to be $\Sigma_{uc} \cap \Sigma_{uo}$-compatible if

$$\forall y \in Y, \forall \sigma \in \Sigma_{uc} \cap \Sigma_{uo} : \beta(y, \sigma) = \{y\}.$$

Note that a supervisor $S \in \mathcal{A}(\Sigma)$ that works under partial observation has to be admissible, but it need not be $\Sigma_{uc} \cap \Sigma_{uo}$-compatible. Namely, for any state $y \in Y$ and any $\sigma \in \Sigma_{uc} \cap \Sigma_{uo}$, if there does not exist $(y, x) \in \mathit{Reach}(S \| G)$ such that $\alpha(x, \sigma) \neq \emptyset$, then $\beta(y, \sigma)$ need not be nonempty. Moreover, by the definition of admissibility, the following proposition is trivial.

Proposition 2.3 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and an admissible supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$, we consider a supervisor $S_{com} = (Y, \Sigma, \beta_{com}, Y_0, Y) \in \mathcal{A}(\Sigma)$, where

$$\beta_{com}(y, \sigma) = \begin{cases} \{y\}, & \text{if } \sigma \in (\Sigma_{uc} \cap \Sigma_{uo}) - \Sigma_{uo}(y) \\ \beta(y, \sigma), & \text{otherwise} \end{cases}$$

for each $y \in Y$ and each $\sigma \in \Sigma$. Then, it holds that $S \| G = S_{com} \| G$ and S_{com} is admissible and $\Sigma_{uc} \cap \Sigma_{uo}$-compatible.

Proposition 2.3 shows that any admissible supervisor S can be turned into an equivalent admissible and $\Sigma_{uc} \cap \Sigma_{uo}$-compatible one S_{com} by complementing the transitions of each state y with self-loop transitions by events in $\Sigma_{uc} \cap \Sigma_{uo}$ which are not originally defined at y.

Remark 2.5 In the conventional language-based supervisory control problem for deterministic DESs, the system is modeled as a DFA $G \in \mathcal{A}(\Sigma)$, and the specification is given as a sublanguage $K \subseteq L(G)$ or $K \subseteq L_m(G)$. In the former case where $K \subseteq L(G)$ is given (or equivalently, a DFA $R \in \mathcal{A}(\Sigma)$ such that $L(R) = K \subseteq L(G)$ is given), an admissible deterministic supervisor $S \in \mathcal{A}(\Sigma)$ that satisfies $L(S\|G) = K$ is required to be synthesized. There exists such a supervisor S that exactly achieves K if and only if K is controllable ($\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$) [1] and observable ($\forall s, s' \in \overline{K}, \forall \sigma \in \Sigma_c : [P(s) = P(s') \wedge s\sigma \in \overline{K} \wedge s'\sigma \in L(G)] \Rightarrow s'\sigma \in \overline{K}$) [18, 19]. In the latter case where $K \subseteq L_m(G)$ is given (or equivalently, a DFA $R \in \mathcal{A}(\Sigma)$ such that $L_m(R) = K \subseteq L_m(G)$ is given), the synthesized admissible supervisor S is required to satisfy $L_m(S\|G) = K$ and be nonblocking. The necessary and sufficient conditions for the existence of such a nonblocking supervisor S is that K is controllable, observable, and $L_m(G)$-closed ($\overline{K} \cap L_m(G) = K$) [1].

If the afore-mentioned conditions are not satisfied, a deterministic supervisor $S' \in \mathcal{A}(\Sigma)$ is synthesized for a controllable and observable (and $L_m(G)$-closed, if nonblockingness is considered,) sublanguage K' of K. Here, S' enforces language inclusion as it is such that $L(S'\|G) = K' \subseteq K$ or $L_m(S'\|G) = K' \subseteq K$. Since observability is not preserved under union, there does not exist a supremal solution to the language inclusion control problem (under partial observation) in general.

Chapter 3

Maximally Permissive Similarity Control under Partial Observation

In this chapter, we consider the control problem of enforcing similarity for the nondeterministic system and specification where only a part of the events are observable. This problem requires us to synthesize an admissible supervisor so that the supervised system is simulated by the specification. We provide a necessary and sufficient condition for the existence of such a similarity-enforcing supervisor. When the condition holds, we synthesize such a supervisor which is maximally permissive in the sense that the system supervised by any similarity-enforcing supervisor is simulated by the system supervised by the proposed supervisor. Finally, we show that some redundant states of the synthesized supervisor can be removed without reducing its permissiveness.

The issue of nonblockingness is beyond the scope of this chapter (or equivalently, all the states of the system and specification are regarded as marked), so the marked state sets are omitted in the automaton models.

3.1 Similarity Control Problem under Partial Observation

Given the partially observed nondeterministic DES with the nondeterministic specification, the supervisory control problem of synthesizing a supervisor such that the supervised system is bisimilar to the specification is studied in [54, 56]. A necessary and sufficient condition for the existence of a supervisor that enforces bisimilarity under partial observation was provided in [56]. In the case where the existence condition of such a supervisor is not satisfied, we relax the requirement of enforcing bisimilarity and consider the similarity control problem.

3.1.1 Problem Formulation

Given the partially observed system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ with the uncontrollable event set Σ_{uc} and the unobservable event set Σ_{uo}, the similarity control problem (under partial observation) requires us to synthesize an admissible supervisor $S = (Y, \Sigma, \beta, Y_0) \in \mathcal{A}(\Sigma)$ such that $S \| G \sqsubseteq R$ holds. Such a supervisor is called a similarity-enforcing supervisor. We let

$$\mathcal{S}(G, R) = \{S \in \mathcal{A}(\Sigma) \mid S \text{ is admissible and satisfies } S\|G \sqsubseteq R\}$$

be the set of all similarity-enforcing supervisors for G and R.

Since the similarity control problem only requires that the supervised system is simulated by the specification, a conservative supervisor that unnecessarily restricts the behavior of the system could be a solution to the problem. This motivates us to synthesize a supervisor that is as permissive as possible. We evaluate the permissiveness of a supervisor with respect to the simulation relation. Now, we define the set $\mathcal{MPS}(G, R) \subseteq \mathcal{S}(G, R)$ of all maximally permissive similarity-enforcing supervisors as

$$\mathcal{MPS}(G, R) = \{S \in \mathcal{S}(G, R) \mid \forall S' \in \mathcal{S}(G, R) : S'\|G \sqsubseteq S\|G\}.$$

In this chapter, we synthesize a supervisor that is an element of $\mathcal{MPS}(G, R)$.

Note that a maximally permissive similarity-enforcing supervisor $S \in \mathcal{MPS}(G, R)$ can also be regarded as a supervisor that enforces bisimilarity to a supremal feasible subspecification of R. Let $\mathcal{R}_{\sqsubseteq} \subseteq \mathcal{A}(\Sigma)$ be the set of all subspecifications of R for which the bisimilarity control problem is solvable. Namely,

$$\begin{aligned} \mathcal{R}_{\sqsubseteq} \;=\; & \{R' \in \mathcal{A}(\Sigma) \mid R' \sqsubseteq R \text{ holds, and there exists} \\ & \text{an admissible supervisor } S \in \mathcal{A}(\Sigma) \text{ such that } S\|G \simeq R'\}. \end{aligned}$$

We call $R^* \in \mathcal{A}(\Sigma)$ a supremum of $\mathcal{R}_{\sqsubseteq}$ if

- $\forall R' \in \mathcal{R}_{\sqsubseteq}, R' \sqsubseteq R^*$, and

- $\forall \hat{R} \in \mathcal{A}(\Sigma) : [\forall R' \in \mathcal{R}_{\sqsubseteq} : R' \sqsubseteq \hat{R}] \Rightarrow R^* \sqsubseteq \hat{R}$.

Then, we have the following proposition that is straightforwardly obtained by extending Proposition 4 of [70] to the partial observation case.

Proposition 3.1 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that $\mathcal{MPS}(G, R) \neq \emptyset$, we consider any $S \in \mathcal{MPS}(G, R)$. It holds that $S \parallel G \in \mathcal{R}_{\sqsubseteq}$, and $S \parallel G$ is a supremum of $\mathcal{R}_{\sqsubseteq}$.

Remark 3.1 By the definition of $\mathcal{MPS}(G, R)$, a supervisor $S \in \mathcal{MPS}(G, R)$ is maximally permissive in a global sense. This is achievable by introducing nondeterminism in control. It was shown in [31] that, for the deterministic system and specification, there exists a globally maximal solution when a supervisor is allowed to be nondeterministic. However, such a solution is not synthesized in [31].

Since the order $\sqsubseteq$ on $\mathcal{A}(\Sigma)$ is not antisymmetric, for two arbitrary $S_1, S_2 \in \mathcal{MPS}(G, R)$, we cannot claim that $S_1 = S_2$ or $S_1\|G = S_2\|G$. We only have $S_1\|G \sim S_2\|G$ ($S_1\|G \simeq S_2\|G$ does not necessarily hold, either), which implies that S_1 is as permissive as S_2 (with respect to the simulation relation). Since simulation equivalence is stronger than language equivalence, $L(S_1\|G)$ and $L(S_2\|G)$ are the same. That is, all maximally permissive supervisors achieve the same generated language, which is the supremal one achievable by similarity-enforcing supervisors.

Also note that, when $\Sigma_{uo} = \emptyset$, the similarity control problem under partial observation is reduced to that under full observation, which is a problem of synthesizing a Σ_{uc}-admissible supervisor $S \in \mathcal{A}(\Sigma)$ such that $S\|G \sqsubseteq R$ holds for the given $G \in \mathcal{A}(\Sigma)$ and $R \in \mathcal{A}(\Sigma)$. In this special case, to emphasize that the system under consideration is fully observed, we use $\mathcal{S}^{fo}(G, R)$ and $\mathcal{MPS}^{fo}(G, R)$ to denote the set of solutions and the set of maximal solutions to the similarity control problem, respectively.

3.1.2 Motivative Example

We consider a manufacturing system shown in Figure 3.1 (a), which consists of two stations, two rooms (R1 and R2), a storehouse (SH), a rail, and a robot. There are two kinds of parts A and B in station 1. Initially, the robot is in station 1, and moves to area 1 after picking a part. In area 1, the robot first goes to one of the two rooms for processing the picked part.

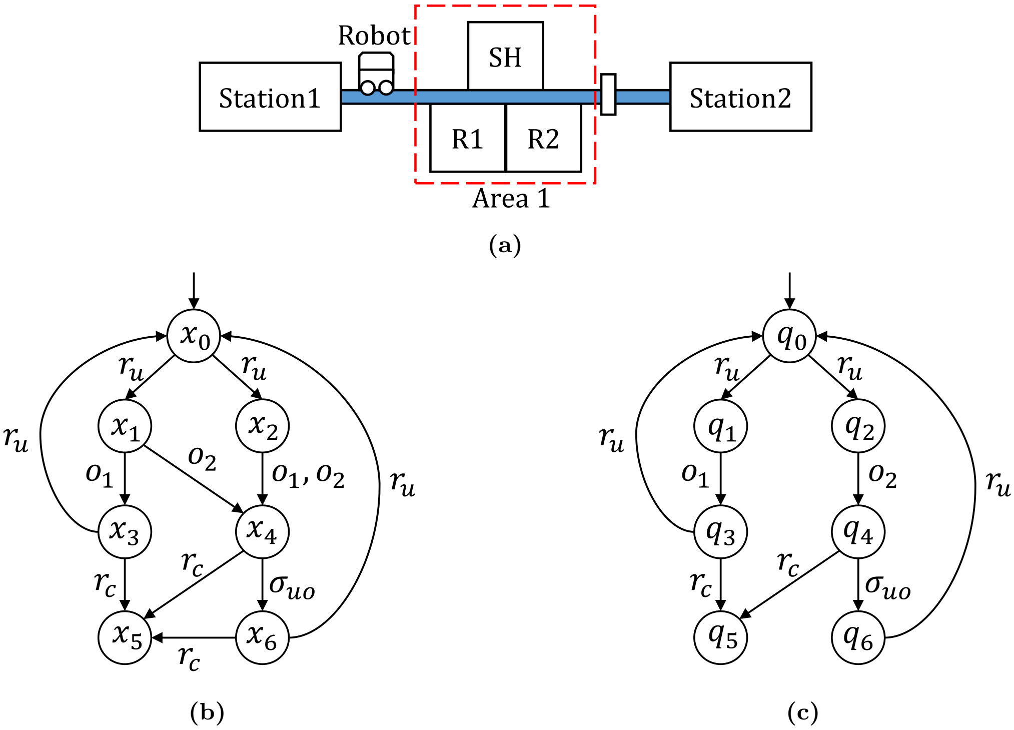


Figure 3.1: (a) A manufacturing system, (b) system model G, and (c) specification R.

Operations 1 and 2 can be performed in R1 and R2, respectively. When part A is processed by operation 1, the robot directly moves to station 1 or 2 with the processed part. Otherwise, the robot can directly bring the part to station 2, or unload the part in SH and then moves to one of the two stations.

This manufacturing system is modeled as $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ shown in Figure 3.1 (b). Move of the robot between station 1 and area 1 and that between station 2 and area 1 are detected, which are modeled as events r_u and r_c, respectively, while the picked part cannot be identified. The transitions $x_0 \xrightarrow{r_u} x_1$ and $x_0 \xrightarrow{r_u} x_2$ denote that the robot moves to area 1 with parts A and B, respectively. Besides, both operations 1 and 2 can be detected, which are denoted by o_1 and o_2, respectively. σ_{uo} denotes the unobservable action of moving to SH and unloading a part. Events o_1, o_2, and σ_{uo} are controlled by opening or closing the entrances of R1, R2, and SH, respectively. The robot can freely travel back and forth between station 1 and area 1, while there is a controllable barrier on the rail from area 1 to station 2 so that r_c can be disabled. Then, we have $\Sigma = \{r_u, r_c, o_1, o_2, \sigma_{uo}\}$, $\Sigma_{uo} = \{\sigma_{uo}\}$, and $\Sigma_{uc} = \{r_u\}$.

As the specification, we consider the nondeterministic automaton $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ shown in Figure 3.1 (c). After the occurrence of r_u, regardless of the kind of the picked part, the choice between the operations o_1 and o_2 is prohibited. This requirement is not captured by the language inclusion control as discussed later, where a supervisor $S_{det} \in \mathcal{A}(\Sigma)$ such that $L(S_{det} \| G) \subseteq L(R)$ is synthesized. Besides, only the part processed by o_2 can be unloaded in SH, and if the part is unloaded, the robot must go back to station 1.

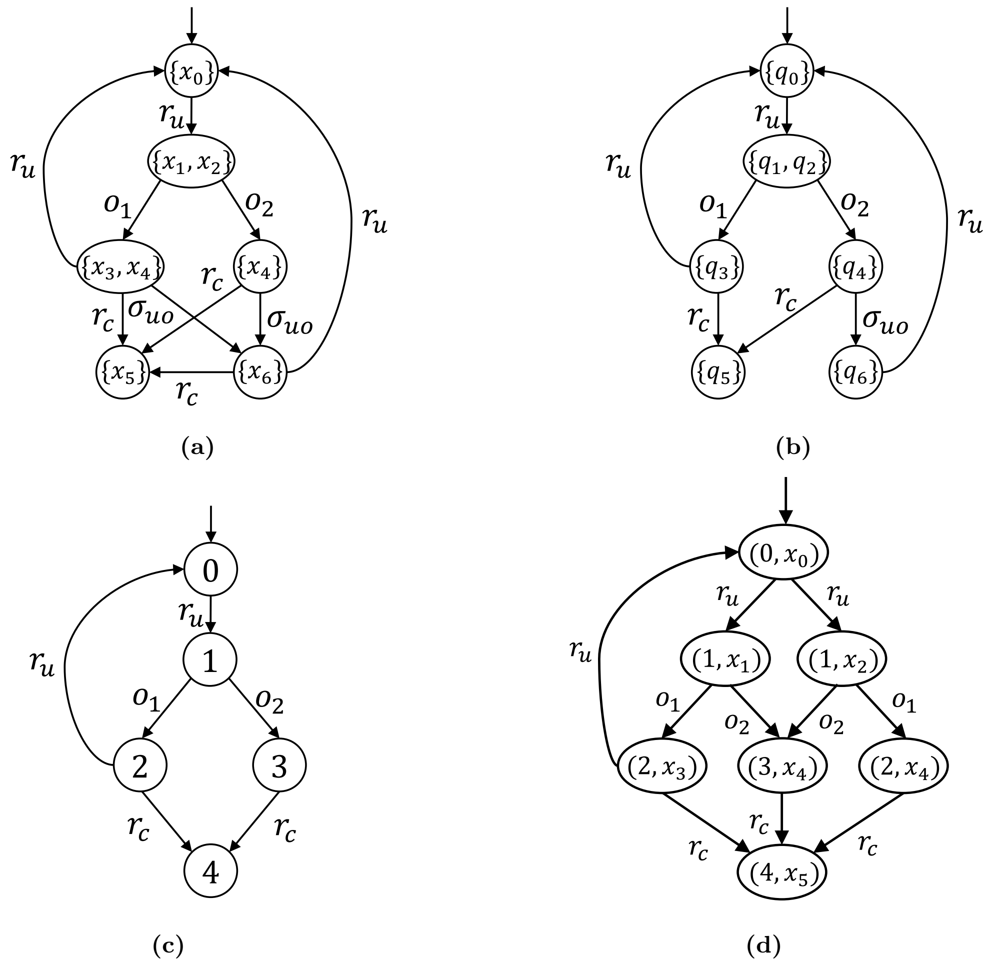


Figure 3.2: (a) Determinized system G_{det}, (b) determinized specification R_{det}, (c) deterministic supervisor S_{det}, and (d) supervised system $S_{det} \| G$.

By transforming G and R into language-equivalent deterministic automata G_{det} and R_{det} as shown in Figures 3.2 (a) and (b), respectively, we can synthesize a maximally permissive supervisor S_{det} shown in Figure 3.2 (c) that achieves a maximal controllable and observable sublanguage of $L(R_{det})$ using the existing methods for the deterministic system and specification [25, 26, 29]. The reachable part of the supervised system $S_{det} \| G$ is shown in Figure 3.2 (d). It can be verified that both o_1 and o_2 are possible at states $(1, x_1)$ and $(1, x_2)$ in $S_{det} \| G$. This violates the requirement that the choice between the operations o_1 and o_2 should be prohibited after r_u occurs.

Also note that there does not exist a supervisor that enforces bisimilarity for the given G and R. These motivate the consideration of the similarity control problem.

3.2 Solvability of the Similarity Control Problem under Partial Observation

3.2.1 Σ_{uc}-Similarity

To verify whether there exists a similarity-enforcing supervisor, we need the following notion of Σ_{uc}-similarity.

Definition 3.1 [63] For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$, a relation $\Phi \subseteq X \times Q$ is said to be a Σ_{uc}-simulation relation from G to R if

- $\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in \Phi$, and
- $\forall (x, q) \in \Phi, \forall \sigma \in \Sigma_{uc}, \forall x' \in \alpha(x, \sigma), \exists q' \in \delta(q, \sigma) : (x', q') \in \Phi$.

If there exists a Σ_{uc}-simulation relation from G to R, G is said to be Σ_{uc}-simulated by R.

When G is Σ_{uc}-simulated by R, all the sequences of transitions by uncontrollable events from initial states of G are allowed by R. It was shown in Theorem 1 of [64] that Σ_{uc}-similarity is a necessary and sufficient condition for $\mathcal{S}^{fo}(G, R) \neq \emptyset$. This result can be extended to the case of partial observation as follows.

Proposition 3.2 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$, $\mathcal{S}(G, R) \neq \emptyset$ if and only if G is Σ_{uc}-simulated by R.

Proof ($\Leftarrow$) Let the supervisor

$$S_u = (Y_u, \Sigma, \beta_u, Y_{u0}) \in \mathcal{A}(\Sigma) \tag{3.1}$$

be such that $Y_u = Y_{u0} = \{y_0\}$ and

$$\beta(y_0, \sigma) = \begin{cases} \{y_0\}, & \text{if } \sigma \in \Sigma_{uc} \\ \emptyset, & \text{otherwise} \end{cases}$$

for each $\sigma \in \Sigma$. Since S_u only has a single state y_0 at which self-loop transitions caused by uncontrollable events are defined, it is admissible. Since G is Σ_{uc}-simulated by R, there exists a Σ_{uc}-simulation relation $\Phi_u \subseteq X \times Q$ from G to R. We show that the relation $\Psi_u \subseteq (Y_u \times X) \times Q$ defined as

$$\Psi_u = \{((y_0, x), q) \in (Y_u \times X) \times Q \mid (x, q) \in \Phi_u\}$$

is a simulation relation from $S_u \| G$ to R.

- We consider any $(y_0, x_0) \in Y_{u0} \times X_0$. Since Φ_u is a Σ_{uc}-simulation relation from G to R, there exists $q_0 \in Q_0$ such that $(x_0, q_0) \in \Phi_u$. It follows that $((y_0, x_0), q_0) \in \Psi_u$.

- For any $((y_0, x), q) \in \Psi_u$ and any $\sigma \in \Sigma$, we consider any $(y_0, x') \in \xi_u((y_0, x), \sigma)$, where $\xi_u : (Y_u \times X) \times \Sigma \to 2^{Y_u \times X}$ is the transition function of $S_u \| G$. We have $(x, q) \in \Phi_u$, $y_0 \in \beta_u(y_0, \sigma)$ and $x' \in \alpha(x, \sigma)$. It follows from $\beta_u(y_0, \sigma) \neq \emptyset$ that $\sigma \in \Sigma_{uc}$. Since Φ_u is a Σ_{uc}-simulation relation from G to R, there exists $q' \in \delta(q, \sigma)$ such that $(x', q') \in \Phi_u$. It follows that $((y_0, x'), q') \in \Psi_u$.

Hence, we have $S_u \in \mathcal{S}(G, R) \neq \emptyset$.

($\Rightarrow$) Let $S = (Y, \Sigma, \beta, Y_0) \in \mathcal{A}(\Sigma)$ be any supervisor such that $S \in \mathcal{S}(G, R) \neq \emptyset$. Since $S \| G \sqsubseteq R$, there exists a simulation relation $\Psi \subseteq (Y \times X) \times Q$ from $S \| G$ to R. We show that the relation $\Phi \subseteq X \times Q$ defined as

$$\Phi = \{(x, q) \in X \times Q \mid \exists y \in Y : ((y, x), q) \in \Psi \wedge (y, x) \in \mathit{Reach}(S \| G)\} \tag{3.2}$$

is a Σ_{uc}-simulation relation from G to R.

- We consider any $x_0 \in X_0$. For any $y_0 \in Y_0$, we have $(y_0, x_0) \in Y_0 \times X_0 \subseteq \mathit{Reach}(S \| G)$. Since Ψ is a simulation relation from $S \| G$ to R, there exists $q_0 \in Q_0$ such that $((y_0, x_0), q_0) \in \Psi$. It follows that $(x_0, q_0) \in \Phi$.

- For any $(x, q) \in \Phi$ and any $\sigma \in \Sigma_{uc}$, we consider any $x' \in \alpha(x, \sigma)$. Since $(x, q) \in \Phi$, there exists $y \in Y$ such that $((y, x), q) \in \Psi$ and $(y, x) \in \mathit{Reach}(S \| G)$. As S is admissible, it follows from $x' \in \alpha(x, \sigma) \neq \emptyset$ that $\beta(y, \sigma) \neq \emptyset$. For any $y' \in \beta(y, \sigma)$, it holds that $(y', x') \in \beta(y, \sigma) \times \alpha(x, \sigma) = \xi((y, x), \sigma) \subseteq \mathit{Reach}(S \| G)$, where $\xi : (Y \times X) \times \Sigma \to 2^{Y \times X}$

is the transition function of $S\|G$. Moreover, since Ψ is a simulation relation from $S\|G$ to R, there exists $q' \in \delta(q,\sigma)$ such that $((y',x'),q') \in \Psi$. It follows that $(x',q') \in \Phi$.

Hence, G is Σ_{uc}-simulated by R. $\square$

If G is Σ_{uc}-simulated by R, regardless of whether the system is fully or partially observed, the supervisor that always disables all controllable events is an element of $\mathcal{S}(G,R)$. When both G and R are deterministic, the condition of Σ_{uc}-similarity is reduced to $\Sigma_{uc}^* \cap L(G) \subseteq L(R)$.

3.2.2 Computation of the Unique Largest Σ_{uc}-Simulation Relation

To effectively verify whether the similarity control problem is solvable, and also to find the "largest" Σ_{uc}-simulation relation which will be used in the supervisor synthesis later, we consider a function $F : 2^{X\times Q} \to 2^{X\times Q}$ defined over $2^{X\times Q}$ as follows [63]:

$$F(W) = \{(x,q) \in W \mid \forall\sigma \in \Sigma_{uc}, \forall x' \in \alpha(x,\sigma), \exists q' \in \delta(q,\sigma) : (x',q') \in W\}, \tag{3.3}$$

where $W \in 2^{X\times Q}$ is an arbitrary relation over $X \times Q$. W is said to be a fixpoint of the function F if $W = F(W)$. Since $(2^{X\times Q}, \subseteq)$ is a complete lattice and F is monotone, that is,

$$\forall W, W' \in 2^{X\times Q} : W \subseteq W' \Rightarrow F(W) \subseteq F(W'),$$

it follows from Tarski's fixpoint theorem [71] that there exists the unique maximal fixpoint $W^{\uparrow}$ of F. As the state sets X and Q of G and R, respectively, are finite, the unique maximal fixpoint can be computed by

$$W^{\uparrow} = \bigcap_{i\geq 0} F^i(X \times Q)$$

after at most $|X||Q|$ iterations, where $F^i(X \times Q)$ is computed inductively as

$$F^i(X \times Q) = \begin{cases} X \times Q, & i = 0 \\ F(F^{i-1}(X \times Q)), & i \geq 1. \end{cases}$$

The complexity of this computation is $O(|\Sigma_{uc}||X|^3|Q|^3)$ [63].

The following proposition indicates that Σ_{uc}-simulation relations can be characterized as fixpoints of F [63].

Proposition 3.3 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$, the following two statements hold for the function $F : 2^{X\times Q} \to 2^{X\times Q}$ defined by (3.3).

- For any $\Phi \subseteq X \times Q$, Φ is a Σ_{uc}-simulation relation from G to R if and only if $\Phi = F(\Phi)$ and

$$\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in \Phi.$$

- G is Σ_{uc}-simulated by R if and only if the unique maximal fixpoint $W^{\uparrow}$ of the function F satisfies

$$\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in W^{\uparrow}.$$

By Proposition 3.3, $W^{\uparrow}$ is a Σ_{uc}-simulation relation from G to R when G is Σ_{uc}-simulated by R. Also note that, for any Σ_{uc}-simulation relation Φ from G to R, it holds that $\Phi \subseteq W^{\uparrow}$. Namely, $W^{\uparrow}$ is the "largest" Σ_{uc}-simulation relation from G to R.

Example 3.1 We consider the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ shown in Figures 3.3 (a) and (b), respectively, where $\Sigma = \{a, b, c, d, uc, uo\}$, $\Sigma_{uc} = \{\sigma_{uc}\}$, and $\Sigma_{uo} = \{\sigma_{uo}\}$.

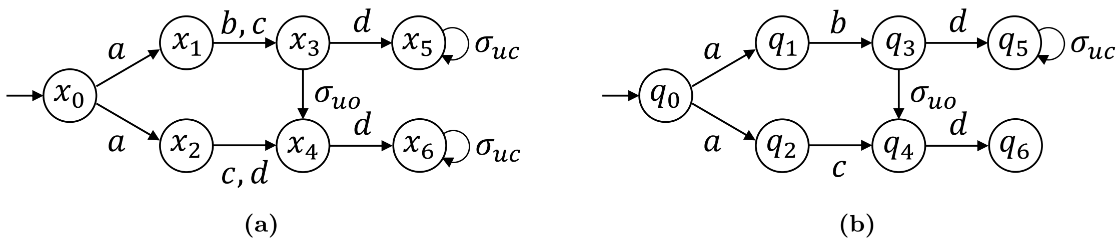


Figure 3.3: (a) System G and (b) specification R.

By iteratively applying the function $F : 2^{X\times Q} \to 2^{X\times Q}$ defined by (3.3) to $X \times Q$, the unique maximal fixpoint $W^{\uparrow}$ of F is obtained as

$$\begin{aligned}
F^1(X \times Q) &= X \times Q - \{x_5, x_6\} \times \{q_0, q_1, q_2, q_3, q_4, q_6\} \\
&= \{(x_0, q_0), (x_0, q_1), (x_0, q_2), (x_0, q_3), (x_0, q_4), (x_0, q_5), (x_0, q_6), (x_1, q_0), (x_1, q_1), \\
&\quad (x_1, q_2), (x_1, q_3), (x_1, q_4), (x_1, q_5), (x_1, q_6), (x_2, q_0), (x_2, q_1), (x_2, q_2), (x_2, q_3), \\
&\quad (x_2, q_4), (x_2, q_5), (x_2, q_6), (x_3, q_0), (x_3, q_1), (x_3, q_2), (x_3, q_3), (x_3, q_4), (x_3, q_5), \\
&\quad (x_3, q_6), (x_4, q_0), (x_4, q_1), (x_4, q_2), (x_4, q_3), (x_4, q_4), (x_4, q_5), (x_4, q_6), (x_5, q_5), \\
&\quad (x_6, q_5)\} \\
&= F^2(X \times Q) \\
&= W^{\uparrow}.
\end{aligned}$$

Since $(x_0, q_0) \in W^{\uparrow}$ for the initial states x_0 and q_0 of G and R, respectively, by Proposition 3.3, G is Σ_{uc}-simulated by R, and $W^{\uparrow}$ is the largest Σ_{uc}-simulation relation from G to R. Besides, by Proposition 3.2, we have $\mathcal{S}(G, R) \neq \emptyset$.

3.3 Supervisor Synthesis

Hereafter, we assume that the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ is Σ_{uc}-simulated by the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$. In this section, we synthesize a supervisor using the unique maximal fixpoint $W^{\uparrow} \subseteq X \times Q$ of the function $F : 2^{X \times Q} \to 2^{X \times Q}$ defined by (3.3).

We define the set of admissible control patterns on unobservable events as

$$\Gamma_{uo} = \{\gamma \in 2^{\Sigma_{uo}} \mid \Sigma_{uc} \cap \Sigma_{uo} \subseteq \gamma\}.$$

For any $W \subseteq X \times Q$, we let

$$\pi_X(W) = \{x \in X \mid \exists q \in Q : (x, q) \in W\}.$$

Besides, for any $\mathcal{W} \subseteq 2^{W^{\uparrow}}$, $\min \mathcal{W} \subseteq \mathcal{W}$ denotes the set of minimal elements of $\mathcal{W}$, that is,

$$\min \mathcal{W} = \{W \in \mathcal{W} \mid \nexists W' \in \mathcal{W} : W' \subseteq W \wedge W' \neq W\}.$$

As a state of the supervisor to be synthesized, we consider a 3-tuple $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in 2^{W^\uparrow} \times \Gamma_{uo} \times 2^{W^\uparrow}$, whose elements can be informally explained as follows:

- $W_1^y \in 2^{W^\uparrow}$ is used to estimate the current state of G at the moment an observable event occurs. In each element (x, q) of W_1^y, x is a candidate of the current state of G, and q is a state of R, which is related to x by $W^\uparrow$ such that transitions from x should be simulated by those from q.

- $\gamma_{uo}^y \in \Gamma_{uo}$ is the set of unobservable events to be enabled at y, which contains all the uncontrollable (and unobservable) events.

- Some enabled unobservable events in γ_{uo}^y may occur in G before the next occurrence of an observable event. $W_2^y \in 2^{W^\uparrow}$ is used to estimate states that are reachable from those in $\pi_X(W_1^y)$ by enabled unobservable events. Similar to W_1^y, W_2^y consists of pairs of states x and q that are related by $W^\uparrow$.

To compute W_2^y from W_1^y and γ_{uo}^y, a function $U : 2^{W^\uparrow} \times \Gamma_{uo} \to 2^{2^{W^\uparrow}}$ is defined as

$$\begin{aligned} U(W,\gamma) \quad = \quad & \{W' \in 2^{W^\uparrow} \mid W \subseteq W' \wedge [\forall (x,q) \in W', \forall \sigma \in \gamma, \\ & \forall x' \in \alpha(x,\sigma), \exists q' \in \delta(q,\sigma) : (x',q') \in W']\} \end{aligned} \tag{3.4}$$

for each $W \in 2^{W^\uparrow}$ and each $\gamma \in \Gamma_{uo}$. Each element W' of $U(W, \gamma)$ is a certain invariant set that includes W. For any pair $(x, q) \in W'$, if there is a transition from x to $x' \in \alpha(x, \sigma)$ by an unobservable event $\sigma \in \gamma$, then a corresponding transition from q to $q' \in \delta(q, \sigma)$ such that $(x', q') \in W'$ must be defined, namely, all transitions from x by σ are allowed by the specification R. Hence, the nonemptiness of $U(W, \gamma)$ indicates that, for the estimated states in $\pi_X(W)$, γ is a "safe" control pattern on unobservable events. For each $x' \in \alpha(x, \sigma)$, $q' \in \delta(q, \sigma)$ such that $(x', q') \in W'$ is not unique in general. It is sufficient for W' to include one such pair (x', q'). In addition, any state pair that cannot be reached from any pair in W is irrelevant. These are the reasons that we only consider minimal elements of $U(W, \gamma)$. We have the following lemma on $\min U(W, \gamma)$.

Lemma 3.1 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, we consider any $W \in 2^{W^\uparrow} - \{\emptyset\}$, any $\gamma \in \Gamma_{uo}$, and any $W' \in \min U(W, \gamma)$. It holds that

$$\forall (x, q) \in W, \forall s \in \gamma^*, \forall x' \in \alpha(x, s), \exists q' \in \delta(q, s) : (x', q') \in W', \tag{3.5}$$

$$W' \subseteq \bigcup_{s \in \gamma^*} \bigcup_{(x,q) \in W} \alpha(x, s) \times \delta(q, s), \tag{3.6}$$

and

$$\bigcup_{s \in \gamma^*} \alpha(\pi_X(W), s) = \pi_X(W'). \tag{3.7}$$

Proof To prove (3.5), for any $(x, q) \in W$ and any $s \in \gamma^*$, we let $n = |s|$ and consider any $x' \in \alpha(x, s)$. Since $W' \in \min U(W, \gamma) \subseteq U(W, \gamma)$, it follows from the definition of U that $(x, q) \in W \subseteq W'$. In the case where $n = 0$, we have $s = \varepsilon$ and $\{x\} = \alpha(x, \varepsilon)$, that is, $x' = x$. Moreover, we have $\{q\} = \delta(q, \varepsilon)$. Then, we have $(x, q) \in W'$. In the case where $n > 0$, we write s as $s = \sigma_1\sigma_2 \cdots \sigma_n$. Then, there exist $x_1, x_2, \cdots, x_n \in X$ such that $x_1 \in \alpha(x, \sigma_1)$, $x_n = x'$, and $x_i \in \alpha(x_{i-1}, \sigma_i)$ for $i = 2, 3, \cdots, n$. Since $(x, q) \in W'$, by the definition of U, there exist $q_1, q_2, \cdots, q_n \in Q$ such that $q_1 \in \delta(q, \sigma_1)$, $q_i \in \delta(q_{i-1}, \sigma_i)$ for $i = 2, 3, \cdots, n$, and $(x_i, q_i) \in W'$ for $i = 1, 2, \cdots, n$. That is, there exists $q_n \in \delta(q, s)$ such that $(x_n, q_n) \in W'$.

Since W' is a minimal element of $U(W, \gamma)$, (3.6) holds. By (3.5) and (3.6), we have (3.7).
□

Remark 3.2 For any $W \in 2^{W^\uparrow}$ and any $\gamma \in \Gamma_{uo}$, the standard unobservable reach $SUR(W, \gamma) \subseteq X \times Q$ is the right-hand side of (3.6), namely,

$$SUR(W, \gamma) = \bigcup_{s \in \gamma^*} \bigcup_{(x,q) \in W} \alpha(x, s) \times \delta(q, s).$$

By (3.6), we have $W' \subseteq SUR(W, \gamma)$ for any $W' \in \min U(W, \gamma)$, but the equality does not hold in general. For any $(x, q) \in W$ and any $s \in \gamma^*$, we consider any $x' \in \alpha(x, s)$. Then, a corresponding state $q' \in \delta(q, s)$ such that $(x', q') \in W^\uparrow$ is not necessarily unique. Here, one possible pair $(x', q') \in W^\uparrow$ is included in $W' \in \min U(W, \gamma)$, while all possible pairs are included in $SUR(W, \gamma)$. This is the reason that $W' = SUR(W, \gamma)$ does not necessarily hold. Note that if R is deterministic, then it holds that $W' = SUR(W, \gamma)$.

Now, we define the state set $Y_{W^\uparrow} \subseteq 2^{W^\uparrow} \times \Gamma_{uo} \times 2^{W^\uparrow}$ as

$$\begin{aligned} Y_{W^\uparrow} &= \{(W_1^y, \gamma_{uo}^y, W_2^y) \in 2^{W^\uparrow} \times \Gamma_{uo} \times 2^{W^\uparrow} \mid W_1^y \neq \emptyset \wedge \\ &\quad W_2^y \in \min U(W_1^y, \gamma_{uo}^y) \wedge [\forall \sigma \in \gamma_{uo}^y \cap \Sigma_c : \alpha(\pi_X(W_2^y), \sigma) \neq \emptyset]\}. \end{aligned} \tag{3.8}$$

As the initial state set, a subset $Y_{W^\uparrow 0} \subseteq Y_{W^\uparrow}$ is defined as

$$Y_{W^\uparrow 0} = \{(W_1^{y_0}, \gamma_{uo}^{y_0}, W_2^{y_0}) \in Y_{W^\uparrow} \mid |W_1^{y_0}| = |X_0| \wedge [\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in W_1^{y_0}]\}.$$

In order to synthesize a supervisor, we need to prove $Y_{W^\uparrow 0} \neq \emptyset$ first. For this purpose, we need the following lemma.

Lemma 3.2 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the state set $Y_{W^\uparrow} \subseteq 2^{W^\uparrow} \times \Gamma_{uo} \times 2^{W^\uparrow}$ defined by (3.8) satisfies

$$\forall W \in 2^{W^\uparrow} - \{\emptyset\}, \exists W' \in 2^{W^\uparrow} : (W, \Sigma_{uc} \cap \Sigma_{uo}, W') \in Y_{W^\uparrow}.$$

Proof Since G is Σ_{uc}-simulated by R, by Proposition 3.3, $W^\uparrow$ is a Σ_{uc}-simulation relation from G to R. For any $W \in 2^{W^\uparrow} - \{\emptyset\}$, we have $W \subseteq W^\uparrow$ and

$$\forall (x, q) \in W^\uparrow, \forall \sigma \in \Sigma_{uc} \cap \Sigma_{uo}, \forall x' \in \alpha(x, \sigma), \exists q' \in \delta(q, \sigma) : (x', q') \in W^\uparrow.$$

It follows that $W^\uparrow \in U(W, \Sigma_{uc} \cap \Sigma_{uo}) \neq \emptyset$. Since $W \neq \emptyset$ and $(\Sigma_{uc} \cap \Sigma_{uo}) \cap \Sigma_c = \emptyset$, for any $W' \in \min U(W, \Sigma_{uc} \cap \Sigma_{uo}) \subseteq 2^{W^\uparrow}$, we have $(W, \Sigma_{uc} \cap \Sigma_{uo}, W') \in Y_{W^\uparrow}$. $\square$

Lemma 3.2 says that any nonempty $W \in 2^{W^\uparrow}$ can form at least one state $y \in Y_{W^\uparrow}$ that disables all controllable and unobservable events.

Since G is Σ_{uc}-simulated by R, by Proposition 3.3, there exists $W_0 \in 2^{W^\uparrow}$ such that $W_0 \neq \emptyset$, $|W_0| = |X_0|$, and

$$\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in W_0.$$

By Lemma 3.2, there exists $W_0' \in 2^{W^\uparrow}$ such that $(W_0, \Sigma_{uc} \cap \Sigma_{uo}, W_0') \in Y_{W^\uparrow}$. It follows from the definition of $Y_{W^{\uparrow}0}$ that $(W_0, \Sigma_{uc} \cap \Sigma_{uo}, W_0') \in Y_{W^{\uparrow}0} \neq \emptyset$. Now, we can synthesize a nondeterministic supervisor

$$S_{W^{\uparrow}} = (Y_{W^{\uparrow}}, \Sigma, \beta_{W^{\uparrow}}, Y_{W^{\uparrow}0}) \in \mathcal{A}(\Sigma). \tag{3.9}$$

The nondeterministic transition function $\beta_{W^{\uparrow}} : Y_{W^{\uparrow}} \times \Sigma \to 2^{Y_{W^{\uparrow}}}$ is defined as

$$\beta_{W^{\uparrow}}(y, \sigma) = \begin{cases} \{y' = (W_1^{y'}, \gamma_{uo}^{y'}, W_2^{y'}) \in Y_{W^{\uparrow}} \mid W_1^{y'} \in \min N(W_2^y, \sigma)\}, & \text{if } \sigma \in \Sigma_{W^{\uparrow}}(y) \\ \{y\}, & \text{if } \sigma \in \gamma_{uo}^y \\ \emptyset, & \text{otherwise} \end{cases}$$

for each $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W^{\uparrow}}$ and each $\sigma \in \Sigma$, where

$$N(W_2^y, \sigma) = \{W \in 2^{W^{\uparrow}} \mid \forall (x, q) \in W_2^y, \forall x' \in \alpha(x, \sigma), \exists q' \in \delta(q, \sigma) : (x', q') \in W\},$$

and

$$\begin{aligned} \Sigma_{W^{\uparrow}}(y) \quad = \quad & \{\sigma \in \Sigma_o \mid \alpha(\pi_X(W_2^y), \sigma) \neq \emptyset \wedge [\forall (x, q) \in W_2^y, \\ & \forall x' \in \alpha(x, \sigma), \exists q' \in \delta(q, \sigma) : (x', q') \in W^{\uparrow}]\}. \end{aligned}$$

For any $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W^{\uparrow}}$, the set $\Sigma_{W^{\uparrow}}(y)$ is determined by W_2^y. The above definition means that $\Sigma_{W^{\uparrow}}(y)$ is the set of observable events such that all transitions from $\pi_X(W_2^y)$ by these observable events are allowed by the specification R. We prove in the following lemma that those events in $\Sigma_{W^{\uparrow}}(y)$ are always enabled by $S_{W^{\uparrow}}$, namely, $\beta_{W^{\uparrow}}(y, \sigma)$ is nonempty if $\sigma \in \Sigma_{W^{\uparrow}}(y)$.

Lemma 3.3 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $S_{W^{\uparrow}} = (Y_{W^{\uparrow}}, \Sigma, \beta_{W^{\uparrow}}, Y_{W^{\uparrow}0}) \in \mathcal{A}(\Sigma)$ defined by (3.9) satisfies

$$\forall y \in Y_{W^{\uparrow}}, \forall \sigma \in \Sigma_{W^{\uparrow}}(y) : \beta_{W^{\uparrow}}(y, \sigma) \neq \emptyset.$$

Proof For any $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W^{\uparrow}}$, we consider any $\sigma \in \Sigma_{W^{\uparrow}}(y)$. By the definition of $\Sigma_{W^\uparrow}(y)$, there exists $(x,q) \in W_2^y$ such that $\alpha(x,\sigma) \neq \emptyset$. Moreover, it holds that

$$\forall(\tilde{x},\tilde{q}) \in W_2^y, \forall \tilde{x}' \in \alpha(\tilde{x},\sigma), \exists \tilde{q}' \in \delta(\tilde{q},\sigma) : (\tilde{x}',\tilde{q}') \in W^\uparrow,$$

which implies $W^\uparrow \in N(W_2^y,\sigma) \neq \emptyset$. Then, there exists a minimal element $W \in \min N(W_2^y,\sigma)$. Since $\alpha(x,\sigma) \neq \emptyset$ for $(x,q) \in W_2^y$, we have $W \neq \emptyset$. By Lemma 3.2, there exists $W' \in 2^{W^\uparrow}$ such that $(W, \Sigma_{uc} \cap \Sigma_{uo}, W') \in Y_{W^\uparrow}$. Thus, we have $(W, \Sigma_{uc} \cap \Sigma_{uo}, W') \in \beta_{W^\uparrow}(y,\sigma) \neq \emptyset$. $\square$

Example 3.2 For the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$ shown in Figures 3.3 (a) and (b), respectively, using $W^\uparrow$ computed in Example 3.1, we synthesize the supervisor $S_{W^\uparrow} = (Y_{W^\uparrow}, \Sigma, \beta_{W^\uparrow}, Y_{W^\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.9). The reachable part of $S_{W^\uparrow}$ is shown in Figure 3.4, where

$$\begin{aligned}
y_0 &= (\{(x_0,q_0)\}, \emptyset, \{(x_0,q_0)\}),\\
y_1 &= (\{(x_1,q_1),(x_2,q_1)\}, \emptyset, \{(x_1,q_1),(x_2,q_1)\}),\\
y_2 &= (\{(x_1,q_1),(x_2,q_2)\}, \emptyset, \{(x_1,q_1),(x_2,q_2)\}),\\
y_3 &= (\{(x_1,q_2),(x_2,q_1)\}, \emptyset, \{(x_1,q_2),(x_2,q_1)\}),\\
y_4 &= (\{(x_1,q_2),(x_2,q_2)\}, \emptyset, \{(x_1,q_2),(x_2,q_2)\}),\\
y_5 &= (\{(x_3,q_3)\}, \emptyset, \{(x_3,q_3)\}),\\
y_6 &= (\{(x_3,q_3)\}, \{uo\}, \{(x_3,q_3),(x_4,q_4)\}),\\
y_7 &= (\{(x_3,q_4),(x_4,q_4)\}, \emptyset, \{(x_3,q_4),(x_4,q_4)\}),\\
y_8 &= (\{(x_5,q_5)\}, \emptyset, \{(x_5,q_5)\}).
\end{aligned}$$

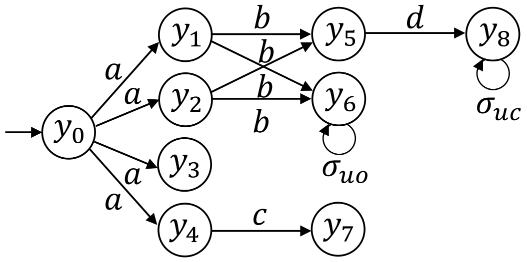


Figure 3.4: Supervisor $S_{W^\uparrow}$.

Remark 3.3 Each state $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W^\uparrow}$ of $S_{W^\uparrow}$ consists of $W_1^y \in 2^{W^\uparrow}$, $\gamma_{uo}^y \in \Gamma_{uo}$, and $W_2^y \in 2^{W^\uparrow}$. $2^{W^\uparrow}$ and Γ_{uo} have at most $2^{|X||Q|}$ and $2^{|\Sigma_c \cap \Sigma_{uo}|}$ elements, respectively. Then, the numbers of states and transitions of $S_{W^\uparrow}$ are bounded by $2^{2|X||Q|+|\Sigma_c\cap\Sigma_{uo}|}$ and $|\Sigma|2^{4|X||Q|+2|\Sigma_c\cap\Sigma_{uo}|}$, respectively. Since the current state of the system G is not necessarily identified uniquely, the exponential complexity is common in supervisor synthesis for nondeterministic DESs and under partial observation.

In the case where $\Sigma_{uo} = \emptyset$, that is, G is fully observed, we have $\gamma_{uo}^y = \emptyset$ for each state $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W^\uparrow}$ of $S_{W^\uparrow}$. In addition, by the definition of $U : 2^{W^\uparrow} \times \Gamma_{uo} \to 2^{2^{W^\uparrow}}$ defined by (3.4), we have $\{W\} = \min U(W, \emptyset)$ for any $W \in 2^{W^\uparrow}$. Hence, y is in the form of $y = (W, \emptyset, W)$. Here, the numbers of states and transitions of $S_{W^\uparrow}$ in the worst case are reduced to $2^{|X||Q|}$ and $|\Sigma|2^{2|X||Q|}$, respectively.

In fact, if $\Sigma_{uo} = \emptyset$, $S_{W^\uparrow}$ defined by (3.9) is reduced to the supervisor

$$S^\uparrow = (Y^\uparrow, \Sigma, \beta^\uparrow, Y_0^\uparrow) \in \mathcal{A}(\Sigma) \tag{3.10}$$

synthesized in [65] for fully observed nondeterministic DESs, which is defined as follows.

- The state set is $Y^\uparrow = 2^{W^\uparrow}$.
- The nondeterministic transition function $\beta^\uparrow : Y^\uparrow \times \Sigma \to 2^{Y^\uparrow}$ is defined as

$$\beta^\uparrow(y, \sigma) = \begin{cases} \min N(W, \sigma) & \text{if } \sigma \in \Sigma^\uparrow(y) \\ \emptyset, & \text{otherwise} \end{cases}$$

for each $W \in Y^\uparrow$ and each $\sigma \in \Sigma$, where

$$N(W, \sigma) = \{W' \in Y^\uparrow \mid \forall (x, q) \in W, \forall x' \in \alpha(x, \sigma), \exists q' \in \delta(q, \sigma) : (x', q') \in W'\},$$

and

$$\begin{aligned} \Sigma^\uparrow(W) \quad = \quad & \{\sigma \in \Sigma \mid \alpha(\pi_X(W), \sigma) \neq \emptyset \wedge [\forall (x, q) \in W, \\ & \forall x' \in \alpha(x, \sigma), \exists q' \in \delta(q, \sigma) : (x', q') \in W^\uparrow]\}. \end{aligned}$$

- The initial state set is defined as

$$Y_{W^{\uparrow}0} = \{W_0 \in Y^{\uparrow} \mid |W| = |X_0| \wedge [\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in W_0]\}.$$

It was shown that $S^{\uparrow}$ is a maximally permissive supervisor that solves the similarity control problem under full observation.

Proposition 3.4 (Theorem 2 of [65]) For the system $G = (X, \Sigma, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $S^{\uparrow} = (Y^{\uparrow}, \Sigma, \beta^{\uparrow}, Y_0^{\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (3.10) satisfies $S^{\uparrow} \in \mathcal{MPS}^{fo}(G, R)$.

Remark 3.4 The supervisor $S_{W^{\uparrow}}$ defined by (3.9) has a similar structure as the Bipartite Transition System (BTS) in [29], which was proposed to synthesize a maximally permissive supervisor for the deterministic system and specification under partial observation. The first element W_1^y and the third element W_2^y of a state $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W^{\uparrow}}$ of $S_{W^{\uparrow}}$ correspond to Y-state and Z-state of BTS, respectively. However, due to the nondeterminism, transitions in $S_{W^{\uparrow}}$ are different from those in BTS. For example, a transition from Y-state to Z-state is determined by the standard unobservable reach in BTS, while a transition from W_1^y to W_2^y is determined by $\min U(W_1^y, \gamma_{uo}^y)$, whose elements are different from the standard unobservable reach, as explained in Remark 3.2.

3.4 Maximal Permissiveness of Synthesized Supervisor

In this section, we prove that the supervisor $S_{W^{\uparrow}}$ defined by (3.9) is a maximally permissive similarity-enforcing supervisor in $\mathcal{MPS}(G, R)$.

3.4.1 Proof of Enforcing Similarity

We first show that $S_{W^{\uparrow}}$ solves the similarity control problem. Here, we prove in the following lemma that $S_{W^{\uparrow}}$ correctly estimates the current state of G.

Lemma 3.4 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $S_{W^{\uparrow}} =$

$(Y_{W^\uparrow}, \Sigma, \beta_{W^\uparrow}, Y_{W^\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.9) satisfies

$$\forall t \in \Sigma_o^* \cap L(S_{W^\uparrow}), \forall y \in \beta_{W^\uparrow}(Y_{W^\uparrow 0}, t) : \bigcup_{s \in P^{-1}(t) \cap L_y(S_{W^\uparrow})} \alpha(X_0, s) = \pi_X(W_2^y), \tag{3.11}$$

where $y = (W_1^y, \gamma_{uo}^y, W_2^y)$.

Proof By the definition of $Y_{W^\uparrow}$, for any $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W^\uparrow}$, we have $W_1^y \in 2^{W^\uparrow} - \{\emptyset\}$, $\gamma_{uo}^y \in \Gamma_{uo}$, and $W_2^y \in \min U(W_1^y, \gamma_{uo}^y)$. Then, the results of Lemma 3.1 can be applied to W_1^y, γ_{uo}^y, and W_2^y. We prove (3.11) by the induction on the length of an observable event string $t \in \Sigma_o^* \cap L(S_{W^\uparrow})$.

For the base step, we consider the case where $|t| = 0$, that is, $t = \varepsilon \in \Sigma_o^* \cap L(S_{W^\uparrow})$. Since $Y_{W^\uparrow 0} = \beta_{W^\uparrow}(Y_{W^\uparrow 0}, \varepsilon)$, we consider any $y_0 = (W_1^{y_0}, \gamma_{uo}^{y_0}, W_2^{y_0}) \in Y_{W^\uparrow 0}$. It follows from the definition of $Y_{W^\uparrow 0}$ that $\pi_X(W_1^{y_0}) = X_0$. By the definition of $\beta_{W^\uparrow}$, we have $P^{-1}(\varepsilon) \cap L_{y_0}(S_{W^\uparrow}) = \gamma_{uo}^{y_0 *}$. By (3.7), we have

$$\bigcup_{s \in P^{-1}(\varepsilon) \cap L_{y_0}(S_{W^\uparrow})} \alpha(X_0, s) = \pi_X(W_2^{y_0}).$$

For the induction step, we suppose that, for any $t \in \Sigma_o^* \cap L(S_{W^\uparrow})$ with $|t| = n \geq 0$ and any $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}}) \in \beta_{W^\uparrow}(Y_{W^\uparrow 0}, t)$,

$$\bigcup_{s \in P^{-1}(t) \cap L_{\tilde{y}}(S_{W^\uparrow})} \alpha(X_0, s) = \pi_X(W_2^{\tilde{y}})$$

holds. We consider the case where $|t| = n + 1$. Any $t \in \Sigma_o^* \cap L(S_{W^\uparrow})$ with $|t| = n + 1$ is written as $t = t'\sigma$, where $|t'| = n$ and $\sigma \in \Sigma_o$.

For any $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in \beta_{W^\uparrow}(Y_{W^\uparrow 0}, t'\sigma)$, we prove

$$\bigcup_{s \in P^{-1}(t'\sigma) \cap L_y(S_{W^\uparrow})} \alpha(X_0, s) = \pi_X(W_2^y). \tag{3.12}$$

- $\left(\pi_X(W_2^y) \subseteq \bigcup_{s \in P^{-1}(t'\sigma) \cap L_y(S_{W^\uparrow})} \alpha(X_0, s)\right)$ We consider any $x \in \pi_X(W_2^y)$. It follows from (3.7) that $x \in \pi_X(W_2^y) = \bigcup_{s'' \in \gamma_{uo}^{y*}} \alpha(\pi_X(W_1^y), s'')$, which implies that there exist $(x'', q'') \in W_1^y$ and $s'' \in \gamma_{uo}^{y*}$ such that $x \in \alpha(x'', s'')$. Since $y \in \beta_{W^\uparrow}(Y_{W^\uparrow 0}, t'\sigma)$, there exists $y' = (W_1^{y'}, \gamma_{uo}^{y'}, W_2^{y'}) \in \beta_{W^\uparrow}(Y_{W^\uparrow 0}, t')$ such that $\sigma \in \Sigma_{W^\uparrow}(y')$ and $y \in \beta_{W^\uparrow}(y', \sigma)$.

We have $W_1^y \in \min N(W_2^{y'}, \sigma)$, which implies

$$W_1^y \subseteq \bigcup_{(x',q') \in W_2^{y'}} \alpha(x', \sigma) \times \delta(q', \sigma).$$

Since $(x'', q'') \in W_1^y$, there exists $(x', q') \in W_2^{y'}$ such that $x'' \in \alpha(x', \sigma)$, which implies $x \in \alpha(x', \sigma s'')$. By the induction hypothesis, for $x' \in \pi_X(W_2^{y'})$, there exists $s' \in P^{-1}(t') \cap L_{y'}(S_{W\uparrow})$ such that $x' \in \alpha(X_0, s')$. It follows that $x \in \alpha(x', \sigma s'') \subseteq \alpha(X_0, s'\sigma s'')$. Moreover, by the definition of P and $\beta_{W\uparrow}$, we have $s'\sigma s'' \in P^{-1}(t'\sigma) \cap L_y(S_{W\uparrow})$. Then, we have $\pi_X(W_2^y) \subseteq \bigcup_{s \in P^{-1}(t'\sigma) \cap L_y(S_{W\uparrow})} \alpha(X_0, s)$.

- $\left(\bigcup_{s \in P^{-1}(t'\sigma) \cap L_y(S_{W\uparrow})} \alpha(X_0, s) \subseteq \pi_X(W_2^y)\right)$ We consider any $s \in P^{-1}(t'\sigma) \cap L_y(S_{W\uparrow})$ and any $x \in \alpha(X_0, s)$. By the definition of P and $\beta_{W\uparrow}$, s can be written as $s = s'\sigma s''$, where $s' \in P^{-1}(t')$ and $s'' \in \gamma_{uo}^{y*}$. Then, there exist $x' \in \alpha(X_0, s')$ and $x'' \in \alpha(x', \sigma)$ such that $x \in \alpha(x'', s'')$. Since $s'\sigma s'' \in L_y(S_{W\uparrow})$, there exists $y' = (W_1^{y'}, \gamma_{uo}^{y'}, W_2^{y'}) \in \beta_{W\uparrow}(Y_{W\uparrow 0}, s') = \beta_{W\uparrow}(Y_{W\uparrow 0}, t')$ such that $\sigma \in \Sigma_{W\uparrow}(y')$ and $y \in \beta_{W\uparrow}(y', \sigma)$. It follows that $s' \in P^{-1}(t') \cap L_{y'}(S_{W\uparrow})$. By the induction hypothesis, we have $x' \in \alpha(X_0, s') \subseteq \pi_X(W_2^{y'})$, which implies that there exists $q' \in Q$ such that $(x', q') \in W_2^{y'}$. Since $W_1^y \in \min N(W_2^{y'}, \sigma)$, there exists $q'' \in \delta(q', \sigma)$ such that $(x'', q'') \in W_1^y$. In addition, it follows from (3.5) that there exists $q \in \delta(q'', s'')$ such that $(x, q) \in W_2^y$. It holds that $\bigcup_{s \in P^{-1}(t'\sigma) \cap L_y(S_{W\uparrow})} \alpha(X_0, s) \subseteq \pi_X(W_2^y)$.

Thus, we have (3.12). These complete the induction. $\square$

The following theorem shows that $S_{W\uparrow}$ is a similarity-enforcing supervisor in $\mathcal{S}(G, R)$.

Theorem 3.1 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $S_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \beta_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.9) satisfies $S_{W\uparrow} \in \mathcal{S}(G, R)$.

Proof We first show the admissibility of $S_{W\uparrow}$. We consider any $(y, x) \in \mathit{Reach}(S_{W\uparrow} \| G)$ and any $\sigma \in \Sigma_{uc}$ such that $\alpha(x, \sigma) \neq \emptyset$, where $y = (W_1^y, \gamma_{uo}^y, W_2^y)$. There exists $s \in L(S_{W\uparrow} \| G)$ such that $(y, x) \in \xi_{W\uparrow}((Y_{W\uparrow 0} \times X_0), s)$, where $\xi_{W\uparrow}$ is the nondeterministic transition function of $S_{W\uparrow} \| G$. It follows that $y \in \beta_{W\uparrow}(Y_{W\uparrow 0}, s)$, $x \in \alpha(X_0, s)$, and $s \in P^{-1}(P(s)) \cap L_y(S_{W\uparrow})$. By the definition of $\beta_{W\uparrow}$, we have $P(s) \in \Sigma_o^* \cap L(S_{W\uparrow})$ and $y \in \beta_{W\uparrow}(Y_{W\uparrow 0}, P(s))$. By Lemma 3.4, it holds that $x \in \pi_X(W_2^y)$, which implies together with $\alpha(x,\sigma) \neq \emptyset$ that $\alpha(\pi_X(W_2^y),\sigma) \neq \emptyset$. Besides, there exists $q \in Q$ such that $(x,q) \in W_2^y$. We consider the case where $\sigma \in \Sigma_{uc} \cap \Sigma_o$. Since $W^\uparrow$ is a Σ_{uc}-simulation relation from G to R, we have

$$\forall (\tilde{x},\tilde{q}) \in W_2^y \subseteq W^\uparrow, \forall \tilde{x}' \in \alpha(\tilde{x},\sigma), \exists \tilde{q}' \in \delta(\tilde{q},\sigma) : (\tilde{x}',\tilde{q}') \in W^\uparrow,$$

which implies together with $\alpha(\pi_X(W_2^y),\sigma) \neq \emptyset$ that $\sigma \in \Sigma_{W^\uparrow}(y)$. Then, by Lemma 3.3, we have $\beta_{W^\uparrow}(y,\sigma) \neq \emptyset$. In the case where $\sigma \in \Sigma_{uc} \cap \Sigma_{uo}$, we have $\sigma \in \gamma_{uo}^y$. It follows from the definition of $\beta_{W^\uparrow}$ that $\{y\} = \beta_{W^\uparrow}(y,\sigma) \neq \emptyset$, which implies together with $\alpha(x,\sigma) \neq \emptyset$ that $\xi_{W^\uparrow}((y,x),\sigma) = \beta_{W^\uparrow}(y,\sigma) \times \alpha(x,\sigma) \neq \emptyset$. Thus, $S_{W^\uparrow}$ satisfies the first condition of admissibility.

By the definition of $\beta_{W^\uparrow}$, it is trivial that $S_{W^\uparrow}$ satisfies the second condition of admissibility. Thus, $S_{W^\uparrow}$ is admissible.

We prove $S_{W^\uparrow} \| G \sqsubseteq R$ by showing that the relation $\Psi \subseteq (Y_{W^\uparrow} \times X) \times Q$ defined as

$$\Psi = \{((y,x),q) \in (Y_{W^\uparrow} \times X) \times Q \mid (y,x) \in \mathit{Reach}(S_{W^\uparrow} \| G) \wedge (x,q) \in W_2^y\}$$

is a simulation relation from $S_{W^\uparrow} \| G$ to R, where $y = (W_1^y, \gamma_{uo}^y, W_2^y)$.

- We consider any $(y_0,x_0) \in Y_{W^\uparrow 0} \times X_0$, where $y_0 = (W_1^{y_0}, \gamma_{uo}^{y_0}, W_2^{y_0})$. It follows that $(y_0,x_0) \in \mathit{Reach}(S_{W^\uparrow} \| G)$. By the definition of $Y_{W^\uparrow 0}$, there exists $q_0 \in Q_0$ such that $(x_0,q_0) \in W_1^{y_0} \subseteq W_2^{y_0}$. Thus, we have $((y_0,x_0),q_0) \in \Psi$.

- For any $((y,x),q) \in \Psi$ and any $\sigma \in \Sigma$, we consider any $(y',x') \in \xi_{W^\uparrow}((y,x),\sigma)$, where $y = (W_1^y, \gamma_{uo}^y, W_2^y)$ and $y' = (W_1^{y'}, \gamma_{uo}^{y'}, W_2^{y'})$. It follows that $y' \in \beta_{W^\uparrow}(y,\sigma)$ and $x' \in \alpha(x,\sigma)$. By the definition of Ψ, we have $(x,q) \in W_2^y$ and $(y,x) \in \mathit{Reach}(S_{W^\uparrow} \| G)$. Since $(y,x) \in \mathit{Reach}(S_{W^\uparrow} \| G)$, we have $(y',x') \in \xi_{W^\uparrow}((y,x),\sigma) \subseteq \mathit{Reach}(S_{W^\uparrow} \| G)$. We consider the case where $\sigma \in \Sigma_o$. Since $y' \in \beta_{W^\uparrow}(y,\sigma) \neq \emptyset$, we have $\sigma \in \Sigma_{W^\uparrow}(y)$. By the definition of $\beta_{W^\uparrow}$, we have $W_1^{y'} \in \min N(W_2^y,\sigma) \subseteq N(W_2^y,\sigma)$. There exists $q' \in \delta(q,\sigma)$ such that $(x',q') \in W_1^{y'} \subseteq W_2^{y'}$, which implies $((y',x'),q') \in \Psi$. In the case where $\sigma \in \Sigma_{uo}$, we have $\sigma \in \gamma_{uo}^y$. It follows from the definition of $\beta_{W^\uparrow}$ that $\{y\} = \beta_{W^\uparrow}(y,\sigma)$, that is, $y = y'$. Since $W_2^y \in U(W_1^y, \gamma_{uo}^y)$, there exists $q' \in \delta(q,\sigma)$ such that $(x',q') \in W_2^y$, which implies $((y,x'),q') \in \Psi$.

Hence, Ψ is a simulation relation from $S_{W^\uparrow} \| G$ to R, and $S_{W^\uparrow} \| G \sqsubseteq R$ holds.

Thus, we have $S_{W^\uparrow} \in \mathcal{S}(G, R)$. $\square$

Example 3.3 We consider the system $G \in \mathcal{A}(\Sigma)$, the specification $R \in \mathcal{A}(\Sigma)$, and the supervisor $S_{W^\uparrow} \in \mathcal{A}(\Sigma)$ shown in Figures 3.3 (a), (b), and Figure 3.4, respectively. The reachable part of the supervised system $S_{W^\uparrow} \| G$ is given in Figure 3.5. We consider the

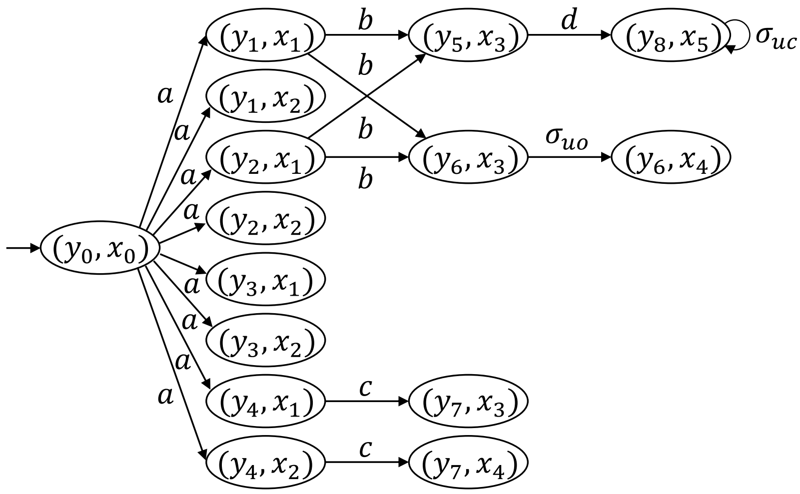


Figure 3.5: Supervised system $S_{W^\uparrow} \| G$.

uncontrollable event σ_{uc} defined at x_5 and x_6 of the system G. σ_{uc} defined at x_5 is not disabled by $S_{W^\uparrow}$, and we have $(y, x_6) \notin \mathit{Reach}(S_{W^\uparrow} \| G)$ for any $y \in Y_{W^\uparrow}$. The unobservable event σ_{uo} is enabled at y_6 with $\{y_6\} = \beta_{W^\uparrow}(y_6, \sigma_{uo})$. It follows that the supervisor $S_{W^\uparrow}$ is admissible. Since the relation $\Psi \subseteq (Y_{W^\uparrow} \times X) \times Q$ given as

$$\begin{aligned}
\Psi &= \{((y, x), q) \in (Y_{W^\uparrow} \times X) \times Q \mid (y, x) \in \mathit{Reach}(S_{W^\uparrow} \| G) \wedge (x, q) \in W_2^y\}, \\
&= \{((y_0, x_0), q_0), ((y_1, x_1), q_1), ((y_1, x_2), q_1), ((y_2, x_1), q_1), ((y_2, x_2), q_2), ((y_3, x_1), q_2), \\
&\quad ((y_3, x_2), q_1), ((y_4, x_1), q_2), ((y_4, x_2), q_2), ((y_5, x_3), q_3), ((y_6, x_3), q_3), ((y_7, x_3), q_4), \\
&\quad ((y_7, x_4), q_4), ((y_8, x_5), q_5), ((y_6, x_4), q_4)\},
\end{aligned}$$

where $y = (W_1^y, \gamma_{uo}^y, W_2^y)$, is a simulation relation from $S_{W^\uparrow} \| G$ to R. Then, we can verify that $S_{W^\uparrow} \in \mathcal{S}(G, R)$.

3.4.2 Proof of Maximal Permissiveness

The maximal permissiveness of $S_{W\uparrow}$ is still left to be proved. Since it is difficult to directly show that $S\|G \sqsubseteq S_{W\uparrow}\|G$ holds for any $S = (Y, \Sigma, \beta, Y_0) \in \mathcal{S}(G, R)$, we introduce another supervisor

$$\tilde{S}_{W\uparrow} = (\tilde{Y}_{W\uparrow}, \Sigma, \tilde{\beta}_{W\uparrow}, \tilde{Y}_{W\uparrow 0}) \in \mathcal{A}(\Sigma) \tag{3.13}$$

defined as follows.

- The state set $\tilde{Y}_{W\uparrow} \subseteq 2^{W^\uparrow} \times \Gamma_{uo} \times 2^{W^\uparrow}$ is defined as

$$\tilde{Y}_{W\uparrow} = \{(W_1^y, \gamma_{uo}^y, W_2^y) \in 2^{W^\uparrow} \times \Gamma_{uo} \times 2^{W^\uparrow} \mid W_1^y \neq \emptyset \wedge W_2^y \in \min U(W_1^y, \gamma_{uo}^y)\}. \tag{3.14}$$

- The nondeterministic transition function $\tilde{\beta}_{W\uparrow} : \tilde{Y}_{W\uparrow} \times \Sigma \to 2^{\tilde{Y}_{W\uparrow}}$ is defined as

$$\tilde{\beta}_{W\uparrow}(y, \sigma) = \begin{cases} \{y' = (W_1^{y'}, \gamma_{uo}^{y'}, W_2^{y'}) \in \tilde{Y}_{W\uparrow} \mid W_1^{y'} \in \min N(W_2^y, \sigma)\}, & \text{if } \sigma \in \Sigma_{W\uparrow}(y) \\ \{y\}, & \text{if } \sigma \in \gamma_{uo}^y \\ \emptyset, & \text{otherwise} \end{cases}$$

  for each $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in \tilde{Y}_{W\uparrow}$ and each $\sigma \in \Sigma$, where $N(W_2^y, \sigma)$ and $\Sigma_{W\uparrow}(y)$ are defined in the same manner as those of $S_{W\uparrow}$.

- The initial state set $\tilde{Y}_{W\uparrow 0} \subseteq \tilde{Y}_{W\uparrow}$ is defined as

$$\tilde{Y}_{W\uparrow 0} = \{(W_1^{y_0}, \gamma_{uo}^{y_0}, W_2^{y_0}) \in \tilde{Y}_{W\uparrow} \mid |W_1^{y_0}| = |X_0| \wedge [\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in W_1^{y_0}]\}.$$

Example 3.4 For the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$ shown in Figures 3.3 (a) and (b), respectively, we can synthesize another supervisor $\tilde{S}_{W\uparrow} = (\tilde{Y}_{W\uparrow}, \Sigma, \tilde{\beta}_{W\uparrow}, \tilde{Y}_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.13). The reachable part of $\tilde{S}_{W\uparrow}$ is shown in Figure 3.6, where some labels of transitions by a and the self-loops by σ_{uo} defined at $\tilde{y}_i$ ($i = 0, 1, 2, 3, 4, 7, 8$) are omitted. Here, states $y_0, y_1, \ldots, y_8$ are the same as those of $S_{W\uparrow} \in \mathcal{A}(\Sigma)$ in Example 3.2. For each $i \in \{0, 1, 2, 3, 4, 7, 8\}$,

$$\tilde{y}_i = (W_1^{\tilde{y}_i}, \gamma_{uo}^{\tilde{y}_i}, W_2^{\tilde{y}_i}) = (W_1^{y_i}, \{\sigma_{uo}\}, W_2^{y_i}),$$

while $y_i = (W_1^{y_i}, \emptyset, W_2^{y_i})$. $\tilde{y}_i$ can be considered as a duplicate of y_i, since the unobservable event σ_{uo} enabled at $\tilde{y}_i$ is not defined at any state of G in $C(\tilde{y}_i)$.

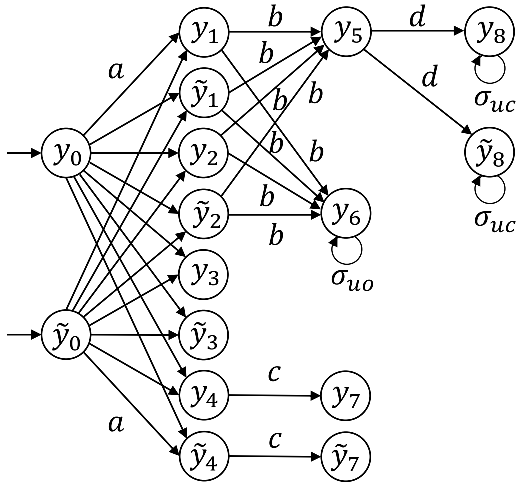


Figure 3.6: Supervisor $\tilde{S}_{W\uparrow}$.

Note that the state set $\tilde{Y}_{W\uparrow}$ of $\tilde{S}_{W\uparrow}$ is obtained by removing the condition

$$\forall \sigma \in \gamma_{uo}^{y} \cap \Sigma_c : \alpha(\pi_X(W_2^y), \sigma) \neq \emptyset \tag{3.15}$$

from $Y_{W\uparrow}$ of $S_{W\uparrow}$. As in Example 3.4, without the restriction of (3.15), $\tilde{S}_{W\uparrow}$ searches every possible unobservable control pattern in Γ_{uo}. This creates some redundant states but provides convenience for the comparison on $\tilde{S}_{W\uparrow}$ and other supervisors. Here, we show that (i) $\tilde{S}_{W\uparrow} \| G \sqsubseteq S_{W\uparrow} \| G$ holds, and that (ii) for any $S \in \mathcal{S}(G, R)$, $S \| G \sqsubseteq \tilde{S}_{W\uparrow} \| G$ holds. Then, we have $S \| G \sqsubseteq S_{W\uparrow} \| G$ by the transitivity of $\sqsubseteq$.

Note that $Y_{W\uparrow} \subseteq \tilde{Y}_{W\uparrow}$ and the following statements hold.

- $Y_{W\uparrow 0} = \tilde{Y}_{W\uparrow 0} \cap Y_{W\uparrow} \subseteq \tilde{Y}_{W\uparrow 0}$, and
- For each $y \in Y_{W\uparrow} \subseteq \tilde{Y}_{W\uparrow}$ and each $\sigma \in \Sigma$, $\beta_{W\uparrow}(y, \sigma) = \tilde{\beta}_{W\uparrow}(y, \sigma) \cap Y_{W\uparrow} \subseteq \tilde{\beta}_{W\uparrow}(y, \sigma)$.

The following corollary can be proved in the same way as Lemma 3.4.

Corollary 3.1 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $\tilde{S}_{W\uparrow}$

$= (\tilde{Y}_{W\uparrow}, \Sigma, \tilde{\beta}_{W\uparrow}, \tilde{Y}_{W\uparrow 0})$ defined by (3.13) satisfies

$$\forall t \in \Sigma_o^* \cap L(\tilde{S}_{W\uparrow}), \forall y \in \tilde{\beta}_{W\uparrow}(\tilde{Y}_{W\uparrow 0}, t) : \bigcup_{s \in P^{-1}(t) \cap L_y(\tilde{S}_{W\uparrow})} \alpha(X_0, s) = \pi_X(W_2^y),$$

where $y = (W_1^y, \gamma_{uo}^y, W_2^y)$.

By the definitions of $Y_{W\uparrow}$ and $\tilde{Y}_{W\uparrow}$, we have the following lemma.

Lemma 3.5 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, we consider the state sets $Y_{W\uparrow}$ and $\tilde{Y}_{W\uparrow}$ defined by (3.8) and (3.14), respectively. For any $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}}) \in \tilde{Y}_{W\uparrow}$, there exists $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W\uparrow}$ such that $W_1^{\tilde{y}} = W_1^y$, $\gamma_{uo}^y \subseteq \gamma_{uo}^{\tilde{y}}$, $W_2^{\tilde{y}} = W_2^y$, and

$$\forall \sigma \in \gamma_{uo}^{\tilde{y}} - \gamma_{uo}^y : \alpha(\pi_X(W_2^{\tilde{y}}), \sigma) = \emptyset.$$

Proof For any $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}}) \in \tilde{Y}_{W\uparrow}$, we consider a subset $\gamma_{uo}^y \subseteq \gamma_{uo}^{\tilde{y}}$ defined as

$$\gamma_{uo}^y = \gamma_{uo}^{\tilde{y}} - \{\sigma \in \gamma_{uo}^{\tilde{y}} \cap \Sigma_c \mid \alpha(\pi_X(W_2^{\tilde{y}}), \sigma) = \emptyset\}. \tag{3.16}$$

It holds that $\gamma_{uo}^y \in \Gamma_{uo}$ and

$$[\forall \sigma \in \gamma_{uo}^{\tilde{y}} - \gamma_{uo}^y : \alpha(\pi_X(W_2^{\tilde{y}}), \sigma) = \emptyset] \wedge [\forall \sigma \in \gamma_{uo}^y \cap \Sigma_c : \alpha(\pi_X(W_2^{\tilde{y}}), \sigma) \neq \emptyset].$$

Since $W_2^{\tilde{y}} \in \min U(W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}})$, it follows from (3.16) that $W_2^{\tilde{y}} \in \min U(W_1^{\tilde{y}}, \gamma_{uo}^y)$. Then, letting $W_1^y = W_1^{\tilde{y}}$ and $W_2^y = W_2^{\tilde{y}}$, we have $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Y_{W\uparrow}$. $\square$

Now, we show in the following lemma that $\tilde{S}_{W\uparrow} \| G \sqsubseteq S_{W\uparrow} \| G$ holds.

Lemma 3.6 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, we consider the supervisors $S_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \beta_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ and $\tilde{S}_{W\uparrow} = (\tilde{Y}_{W\uparrow}, \Sigma, \tilde{\beta}_{W\uparrow}, \tilde{Y}_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.9) and (3.13), respectively. It holds that $\tilde{S}_{W\uparrow} \| G \sqsubseteq S_{W\uparrow} \| G$.

Proof We prove $\tilde{S}_{W\uparrow} \| G \sqsubseteq S_{W\uparrow} \| G$ by showing that the relation $\hat{\Phi} \subseteq (\tilde{Y}_{W\uparrow} \times X) \times (Y_{W\uparrow} \times X)$

defined as

$$\begin{aligned}\hat{\Phi} &= \{((\tilde{y},x),(y,x)) \in (\tilde{Y}_{W^\uparrow} \times X) \times (Y_{W^\uparrow} \times X) \mid W_1^{\tilde{y}} = W_1^y \wedge \gamma_{uo}^y \subseteq \gamma_{uo}^{\tilde{y}} \wedge W_2^{\tilde{y}} = W_2^y \wedge \\ &\quad [\forall \sigma \in \gamma_{uo}^{\tilde{y}} - \gamma_{uo}^y : \alpha(\pi_X(W_2^{\tilde{y}}),\sigma) = \emptyset] \wedge (\tilde{y},x) \in \mathit{Reach}(\tilde{S}_{W^\uparrow} \| G)\}\end{aligned}$$

is a simulation relation from $\tilde{S}_{W^\uparrow}\|G$ to $S_{W^\uparrow}\|G$, where $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}})$ and $y = (W_1^y, \gamma_{uo}^y, W_2^y)$. Let $\tilde{\xi}_{W^\uparrow} : (\tilde{Y}_{W^\uparrow} \times X) \times \Sigma \to 2^{\tilde{Y}_{W^\uparrow} \times X}$ and $\xi_{W^\uparrow} : (Y_{W^\uparrow} \times X) \times \Sigma \to 2^{Y_{W^\uparrow} \times X}$ be the nondeterministic transition functions of $\tilde{S}_{W^\uparrow}\|G$ and $S_{W^\uparrow}\|G$, respectively.

- We consider any $(\tilde{y}_0, x_0) \in \tilde{Y}_{W^\uparrow 0} \times X_0$ with $\tilde{y}_0 = (W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0}, W_2^{\tilde{y}_0})$. By Lemma 3.5, there exists $y_0 = (W_1^{y_0}, \gamma_{uo}^{y_0}, W_2^{y_0}) \in Y_{W^\uparrow} \subseteq \tilde{Y}_{W^\uparrow}$ such that $W_1^{\tilde{y}_0} = W_1^{y_0}$, $\gamma_{uo}^{y_0} \subseteq \gamma_{uo}^{\tilde{y}_0}$, $W_2^{\tilde{y}_0} = W_2^{y_0}$, and
$$\forall \sigma \in \gamma_{uo}^{\tilde{y}_0} - \gamma_{uo}^{y_0} : \alpha(\pi_X(W_2^{\tilde{y}_0}),\sigma) = \emptyset.$$
Since $\tilde{y}_0 \in \tilde{Y}_{W^\uparrow 0}$, we have $y_0 \in \tilde{Y}_{W^\uparrow 0}$. It follows that $y_0 \in \tilde{Y}_{W^\uparrow 0} \cap Y_{W^\uparrow} = Y_{W^\uparrow 0}$. Since $(\tilde{y}_0, x_0) \in \tilde{Y}_{W^\uparrow 0} \times X_0 \subseteq \mathit{Reach}(\tilde{S}_{W^\uparrow}\|G)$, we have $((\tilde{y}_0,x_0),(y_0,x_0)) \in \hat{\Phi}$.

- For any $((\tilde{y},x),(y,x)) \in \hat{\Phi}$ and any $\sigma \in \Sigma$, we consider any $(\tilde{y}',x') \in \tilde{\xi}_{W^\uparrow}((\tilde{y},x),\sigma)$, where $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}})$, $y = (W_1^y, \gamma_{uo}^y, W_2^y)$, and $\tilde{y}' = (W_1^{\tilde{y}'}, \gamma_{uo}^{\tilde{y}'}, W_2^{\tilde{y}'})$. It follows from the definition of $\tilde{\xi}_{W^\uparrow}$ that $\tilde{y}' \in \tilde{\beta}_{W^\uparrow}(\tilde{y},\sigma) \neq \emptyset$ and $x' \in \alpha(x,\sigma) \neq \emptyset$. Since $((\tilde{y},x),(y,x)) \in \hat{\Phi}$, we have $(\tilde{y},x) \in \mathit{Reach}(\tilde{S}_{W^\uparrow}\|G)$, which implies $(\tilde{y}',x') \in \tilde{\xi}_{W^\uparrow}((\tilde{y},x),\sigma) \subseteq \mathit{Reach}(\tilde{S}_{W^\uparrow}\|G)$.

  We consider the case where $\sigma \in \Sigma_o$. It follows from $\tilde{\beta}_{W^\uparrow}(\tilde{y},\sigma) \neq \emptyset$ that $\sigma \in \Sigma_{W^\uparrow}(\tilde{y})$. Since $((\tilde{y},x),(y,x)) \in \hat{\Phi}$, we have $W_2^{\tilde{y}} = W_2^y$, which implies $\sigma \in \Sigma_{W^\uparrow}(y)$. It follows from the definition of $\tilde{\beta}_{W^\uparrow}$ that $\tilde{\beta}_{W^\uparrow}(y,\sigma) = \tilde{\beta}_{W^\uparrow}(\tilde{y},\sigma)$. By Lemma 3.5, there exists $y' = (W_1^{y'}, \gamma_{uo}^{y'}, W_2^{y'}) \in Y_{W^\uparrow} \subseteq \tilde{Y}_{W^\uparrow}$ that satisfies $W_1^{\tilde{y}'} = W_1^{y'}$, $\gamma_{uo}^{y'} \subseteq \gamma_{uo}^{\tilde{y}'}$, $W_2^{\tilde{y}'} = W_2^{y'}$, and
$$\forall \sigma \in \gamma_{uo}^{\tilde{y}'} - \gamma_{uo}^{y'} : \alpha(\pi_X(W_2^{\tilde{y}'}),\sigma) = \emptyset.$$
Since $W_1^{y'} = W_1^{\tilde{y}'} \in \min N(W_2^{\tilde{y}},\sigma)$, we have $y' \in \tilde{\beta}_{W^\uparrow}(\tilde{y},\sigma) \cap Y_{W^\uparrow} = \tilde{\beta}_{W^\uparrow}(y,\sigma) \cap Y_{W^\uparrow} = \beta_{W^\uparrow}(y,\sigma)$. It follows that $(y',x') \in \beta_{W^\uparrow}(y,\sigma) \times \alpha(x,\sigma) = \xi_{W^\uparrow}((y,x),\sigma)$. Thus, we have $((\tilde{y}',x'),(y',x')) \in \hat{\Phi}$.

  In the case where $\sigma \in \Sigma_{uo}$, we have $\sigma \in \gamma_{uo}^{\tilde{y}}$. Since $(\tilde{y},x) \in \mathit{Reach}(\tilde{S}_{W^\uparrow}\|G)$, there exists $s \in L(\tilde{S}_{W^\uparrow}\|G)$ such that $(\tilde{y},x) \in \tilde{\beta}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0},s) \times \alpha(X_0,s) = \tilde{\xi}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0} \times X_0, s)$. By the definition of $\tilde{\beta}_{W^\uparrow}$, we have $P(s) \in \Sigma_o^* \cap L(\tilde{S}_{W^\uparrow})$ and $\tilde{y} \in \tilde{\beta}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0}, P(s))$. Then, it follows from Corollary 3.1 that $x \in \alpha(X_0, s) \subseteq \bigcup_{s' \in P^{-1}(P(s)) \cap L_{\tilde{y}}(S_{W^\uparrow})} \alpha(X_0, s') = \pi_X(W_2^{\tilde{y}})$. Since $x' \in \alpha(x, \sigma) \subseteq \alpha(\pi_X(W_2^{\tilde{y}}), \sigma) \neq \emptyset$ and $((\tilde{y}, x), (y, x)) \in \hat{\Phi}$, it follows from the definition of $\hat{\Phi}$ that $\sigma \in \gamma_{uo}^y$. By the definitions of $\tilde{\beta}_{W^\uparrow}$ and $\beta_{W^\uparrow}$, we have $\{\tilde{y}\} = \tilde{\beta}_{W^\uparrow}(\tilde{y}, \sigma)$ and $\{y\} = \beta_{W^\uparrow}(y, \sigma)$. It follows that $\tilde{y} = \tilde{y}'$ and $(y, x') \in \beta_{W^\uparrow}(y, \sigma) \times \alpha(x, \sigma) = \xi_{W^\uparrow}((y, x), \sigma)$. Besides, by the definition of $\hat{\Phi}$, we have $W_1^{\tilde{y}} = W_1^y$, $\gamma_{uo}^y \subseteq \gamma_{uo}^{\tilde{y}}$, $W_2^{\tilde{y}} = W_2^y$, and

$$\forall \sigma \in \gamma_{uo}^{\tilde{y}} - \gamma_{uo}^y : \alpha(\pi_X(W_2^{\tilde{y}}), \sigma) = \emptyset.$$

Thus, it holds that $((\tilde{y}, x'), (y, x')) \in \hat{\Phi}$.

Hence, $\hat{\Phi}$ is a simulation relation from $\tilde{S}_{W^\uparrow} \| G$ to $S_{W^\uparrow} \| G$. □

We next show $S \| G \sqsubseteq \tilde{S}_{W^\uparrow} \| G$ for any $S \in \mathcal{S}(G, R)$. By the proof of the necessity part of Proposition 3.2, we have the following lemma.

Lemma 3.7 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, we consider any supervisor $S = (Y, \Sigma, \beta, Y_0) \in \mathcal{S}(G, R)$, any simulation relation $\Psi \subseteq (Y \times X) \times Q$ from $S \| G$ to R, and any $y \in Y$. If a relation $\Phi_y \subseteq X \times Q$ defined by

$$\Phi_y = \{(x, q) \in X \times Q \mid ((y, x), q) \in \Psi \wedge (y, x) \in \mathit{Reach}(S \| G)\} \tag{3.17}$$

is nonempty, then, for any nonempty subset $W \subseteq \Phi_y$, it holds that $\Phi_y \in U(W, \Sigma_{uo}(y) \cup (\Sigma_{uc} \cap \Sigma_{uo})) \neq \emptyset$.

Proof We consider any $y \in Y$ such that $\Phi_y \subseteq X \times Q$ defined by (3.17) is nonempty. It holds that $\Phi_y \subseteq \Phi$ for the relation $\Phi \subseteq X \times Q$ defined by (3.2), which is a Σ_{uc}-simulation relation from G to R as shown in the proof of necessity part of Proposition 3.2. Then, for the unique maximal fixpoint $W^\uparrow$ of the function $F : 2^{X \times Q} \to 2^{X \times Q}$ defined by (3.3), we have $\Phi_y \subseteq \Phi \subseteq W^\uparrow$. For any $W \subseteq \Phi_y$ with $W \neq \emptyset$, it holds that $W \in 2^{W^\uparrow}$. To show $\Phi_y \in U(W, \Sigma_{uo}(y) \cup (\Sigma_{uc} \cap \Sigma_{uo}))$, we consider any $(x, q) \in \Phi_y$, any $\sigma \in \Sigma_{uo}(y) \cup (\Sigma_{uc} \cap \Sigma_{uo})$, and any $x' \in \alpha(x, \sigma)$. By the definition of Φ_y, we have $((y, x), q) \in \Psi$ and $(y, x) \in \mathit{Reach}(S \| G)$. Since $\alpha(x, \sigma) \neq \emptyset$, it follows from the admissibility of S that $\sigma \in \Sigma_{uo}(y)$ and $(y, x') \in \beta(y, \sigma) \times \alpha(x, \sigma) = \xi((y, x), \sigma) \subseteq \mathit{Reach}(S \| G)$. Since Ψ is a simulation relation from $S \| G$ to R, there exists $q' \in \delta(q, \sigma)$ such that $((y, x'), q') \in \Psi$, which implies together with $(y, x') \in Reach(S\|G)$ that $(x', q') \in \Phi_y$. It follows from $W \subseteq \Phi_y$ that $\Phi_y \in U(W, \Sigma_{uo}(y) \cup (\Sigma_{uc} \cap \Sigma_{uo})) \neq \emptyset$. $\square$

Now, we can prove the maximal permissiveness of $S_{W^\uparrow}$ in the following theorem.

Theorem 3.2 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $S_{W^\uparrow} = (Y_{W^\uparrow}, \Sigma, \beta_{W^\uparrow}, Y_{W^\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.9) satisfies $S_{W^\uparrow} \in \mathcal{MPS}(G, R)$.

Proof We consider any admissible supervisor $S = (Y, \Sigma, \beta, Y_0) \in \mathcal{S}(G, R)$. There exists a simulation relation $\Psi \subseteq (Y \times X) \times Q$ from $S\|G$ to R. Then, by the proof of necessity part of Proposition 3.2, the relation $\Phi \subseteq X \times Q$ defined by (3.2) is a Σ_{uc}-simulation relation from G to R. For the unique maximal fixpoint $W^\uparrow$ of the function $F : 2^{X \times Q} \to 2^{X \times Q}$ defined by (3.3), we have $\Phi \subseteq W^\uparrow$. Hence, it holds that

$$\forall (x, q) \in X \times Q : [\exists y \in Y : ((y, x), q) \in \Psi \wedge (y, x) \in Reach(S\|G)] \Rightarrow (x, q) \in \Phi \subseteq W^\uparrow. \tag{3.18}$$

By Lemma 3.6, we have $\tilde{S}_{W^\uparrow}\|G \sqsubseteq S_{W^\uparrow}\|G$ for the supervisor $\tilde{S}_{W^\uparrow} = (\tilde{Y}_{W^\uparrow}, \Sigma, \tilde{\beta}_{W^\uparrow}, \tilde{Y}_{W^\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.13). Since $\sqsubseteq$ is transitive, it is sufficient to prove $S\|G \sqsubseteq \tilde{S}_{W^\uparrow}\|G$. Let $\xi : (Y \times X) \times \Sigma \to 2^{Y \times X}$ and $\tilde{\xi}_{W^\uparrow} : (\tilde{Y}_{W^\uparrow} \times X) \times \Sigma \to 2^{\tilde{Y}_{W^\uparrow} \times X}$ denote the nondeterministic transition functions of $S\|G$ and $\tilde{S}_{W^\uparrow}\|G$, respectively. We prove $S\|G \sqsubseteq \tilde{S}_{W^\uparrow}\|G$ by showing that the relation $\tilde{\Phi} \subseteq (Y \times X) \times (\tilde{Y}_{W^\uparrow} \times X)$ defined as

$$\begin{aligned} \tilde{\Phi} \;=\; & \{((y, x), (\tilde{y}, x)) \in (Y \times X) \times (\tilde{Y}_{W^\uparrow} \times X) \mid \gamma_{uo}^{\tilde{y}} = \Sigma_{uo}(y) \cup (\Sigma_{uc} \cap \Sigma_{uo}) \wedge \\ & [\forall (\tilde{x}, \tilde{q}) \in W_2^{\tilde{y}} : ((y, \tilde{x}), \tilde{q}) \in \Psi \wedge (y, \tilde{x}) \in Reach(S\|G)] \wedge \\ & [\exists s \in L(S\|G) : (y, x) \in \xi(Y_0 \times X_0, s) \wedge (\tilde{y}, x) \in \tilde{\xi}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0} \times X_0, s)]\} \end{aligned} \tag{3.19}$$

is a simulation relation from $S\|G$ to $\tilde{S}_{W^\uparrow}\|G$, where $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}})$.

- We consider any $(y_0, x_0) \in Y_0 \times X_0$. To prove that there exists $\tilde{y}_0 = (W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0}, W_2^{\tilde{y}_0}) \in \tilde{Y}_{W^\uparrow 0}$ such that $((y_0, x_0), (\tilde{y}_0, x_0)) \in \tilde{\Phi}$, we first prove that there exists $W_1^{\tilde{y}_0} \in 2^{W^\uparrow}$ such that

$$|W_1^{\tilde{y}_0}| = |X_0| \wedge [\forall x_0' \in X_0, \exists q_0' \in Q_0 : (x_0', q_0') \in W_1^{\tilde{y}_0}], \tag{3.20}$$

and

$$\forall(x_0', q_0') \in W_1^{\tilde{y}_0} : ((y_0, x_0'), q_0') \in \Psi \wedge (y_0, x_0') \in Reach(S\|G). \tag{3.21}$$

For any $x_0' \in X_0$, we have $(y_0, x_0') \in Y_0 \times X_0 \subseteq Reach(S\|G)$. Since Ψ is a simulation relation from $S\|G$ to R, there exists $q_0' \in Q_0$ such that $((y_0, x_0'), q_0') \in \Psi$. By (3.18), we have $(x_0', q_0') \in \Phi \subseteq W^{\uparrow}$. For every $x_0'' \in X_0$, we pick $q_0'' \in Q_0$ with $((y_0, x_0''), q_0'') \in \Psi$ to form a pair $(x_0'', q_0'') \in W^{\uparrow}$ and let $W_1^{\tilde{y}_0} \in 2^{W^{\uparrow}}$ be the set of these $|X_0|$ pairs. Then, $W_1^{\tilde{y}_0}$ satisfies (3.20) and (3.21). Since $X_0 \neq \emptyset$, we have $W_1^{\tilde{y}_0} \neq \emptyset$. For the relation $\Phi_{y_0} \subseteq X \times Q$ defined by (3.17), it follows from (3.21) that $W_1^{\tilde{y}_0} \subseteq \Phi_{y_0} \neq \emptyset$. Let $\gamma_{uo}^{\tilde{y}_0} = \Sigma_{uo}(y_0) \cup (\Sigma_{uc} \cap \Sigma_{uo}) \in \Gamma_{uo}$. It follows from Lemma 3.7 that $\Phi_{y_0} \in U(W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0})$. Then, there exists $W_2^{\tilde{y}_0} \in \min U(W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0}) \subseteq 2^{W^{\uparrow}}$ such that $W_2^{\tilde{y}_0} \subseteq \Phi_{y_0}$. Letting $\tilde{y}_0 = (W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0}, W_2^{\tilde{y}_0})$, we have $\tilde{y}_0 \in \tilde{Y}_{W^{\uparrow}}$. By (3.20), it holds that $\tilde{y}_0 \in \tilde{Y}_{W^{\uparrow}0}$. Then, we have $\varepsilon \in L(S\|G)$, $(y_0, x_0) \in \xi(Y_0 \times X_0, \varepsilon)$, and $(\tilde{y}_0, x_0) \in \tilde{Y}_{W^{\uparrow}0} \times X_0 = \tilde{\xi}_{W^{\uparrow}}(\tilde{Y}_{W^{\uparrow}0} \times X_0, \varepsilon)$. Besides, by $W_2^{\tilde{y}_0} \subseteq \Phi_{y_0}$, we have

$$\forall(\tilde{x}_0, \tilde{q}_0) \in W_2^{\tilde{y}_0} : ((y_0, \tilde{x}_0), \tilde{q}_0) \in \Psi \wedge (y_0, \tilde{x}_0) \in Reach(S\|G).$$

It follows that $((y_0, x_0), (\tilde{y}_0, x_0)) \in \tilde{\Phi}$.

- For any $((y, x), (\tilde{y}, x)) \in \tilde{\Phi}$ and any $\sigma \in \Sigma$, we consider any $(y', x') \in \xi((y, x), \sigma)$. It follows from the definition of ξ that $y' \in \beta(y, \sigma) \neq \emptyset$ and $x' \in \alpha(x, \sigma) \neq \emptyset$. Since $((y, x), (\tilde{y}, x)) \in \tilde{\Phi}$, there exists $s \in L(S\|G)$ such that $(y, x) \in \xi(Y_0 \times X_0, s)$ and $(\tilde{y}, x) \in \tilde{\xi}_{W^{\uparrow}}(\tilde{Y}_{W^{\uparrow}0} \times X_0, s)$. Then, we have $y \in \beta(Y_0, s)$, $\tilde{y} \in \tilde{\beta}_{W^{\uparrow}}(\tilde{Y}_{W^{\uparrow}0}, s)$, and $x \in \alpha(X_0, s)$. By the definition of $\tilde{\beta}_{W^{\uparrow}}$, we have $P(s) \in \Sigma_o^* \cap L(\tilde{S}_{W^{\uparrow}})$ and $\tilde{y} \in \tilde{\beta}_{W^{\uparrow}}(\tilde{Y}_{W^{\uparrow}0}, P(s))$. Letting $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}})$, it follows from Corollary 3.1 that $x \in \alpha(X_0, s) \subseteq \bigcup_{s' \in P^{-1}(P(s)) \cap L_{\tilde{y}}(\tilde{S}_{W^{\uparrow}})} \alpha(X_0, s') = \pi_X(W_2^{\tilde{y}})$, which implies

$$\alpha(\pi_X(W_2^{\tilde{y}}), \sigma) \neq \emptyset. \tag{3.22}$$

We consider the case where $\sigma \in \Sigma_o$. To prove that there exists $(\tilde{y}', x') \in \tilde{\xi}_{W^{\uparrow}}((\tilde{y}, x), \sigma)$ such that $((y', x'), (\tilde{y}', x')) \in \tilde{\Phi}$, we first show $\sigma \in \Sigma_{W^{\uparrow}}(\tilde{y})$. For any $(\tilde{x}, \tilde{q}) \in W_2^{\tilde{y}}$, we consider any $\tilde{x}' \in \alpha(\tilde{x}, \sigma)$. By the definition of $\tilde{\Phi}$, we have $((y, \tilde{x}), \tilde{q}) \in \Psi$ and $(y, \tilde{x}) \in Reach(S\|G)$. Then, it holds that $(y', \tilde{x}') \in \beta(y, \sigma) \times \alpha(\tilde{x}, \sigma) = \xi((y, \tilde{x}), \sigma) \subseteq$

$Reach(S\|G)$. Since Ψ is a simulation relation from $S\|G$ to R, there exists $\tilde{q}' \in \delta(\tilde{q}, \sigma)$ such that $((y', \tilde{x}'), \tilde{q}') \in \Psi$. It follows from (3.18) that $(\tilde{x}', \tilde{q}') \in \Phi \subseteq W^{\uparrow}$. By (3.22), we have $\sigma \in \Sigma_{W^{\uparrow}}(\tilde{y})$. Moreover, letting

$$W = \bigcup_{(\tilde{x},\tilde{q}) \in W_2^{\tilde{y}}} \{(\tilde{x}', \tilde{q}') \in \alpha(\tilde{x}, \sigma) \times \delta(\tilde{q}, \sigma) \mid ((y', \tilde{x}'), \tilde{q}') \in \Psi\},$$

we have $W \subseteq \Phi \subseteq W^{\uparrow}$ and $W \in N(W_2^{\tilde{y}}, \sigma)$. Then, there exists a subset $W_1 \subseteq W$ that is a minimal element of $N(W_2^{\tilde{y}}, \sigma)$, that is, $W_1 \in \min N(W_2^{\tilde{y}}, \sigma)$ such that

$$\forall (\tilde{x}, \tilde{q}) \in W_2^{\tilde{y}}, \forall \tilde{x}' \in \alpha(\tilde{x}, \sigma), \exists \tilde{q}' \in \delta(\tilde{q}, \sigma) : (\tilde{x}', \tilde{q}') \in W_1,$$

and

$$\forall (\tilde{x}', \tilde{q}') \in W_1 : ((y', \tilde{x}'), \tilde{q}') \in \Psi \wedge (y', \tilde{x}') \in Reach(S\|G). \tag{3.23}$$

By (3.22), $W_1 \neq \emptyset$ holds. For the relation $\Phi_{y'} \subseteq X \times Q$ defined by (3.17), it follows from (3.23) that $W_1 \subseteq \Phi_{y'} \neq \emptyset$. Let $\gamma_{uo} = \Sigma_{uo}(y') \cup (\Sigma_{uc} \cap \Sigma_{uo}) \in \Gamma_{uo}$. By Lemma 3.7, we have $\Phi_{y'} \in U(W_1, \gamma_{uo})$. Then, there exists $W_2 \in \min U(W_1, \gamma_{uo})$ such that $W_2 \subseteq \Phi_{y'}$. Letting $\tilde{y}' = (W_1, \gamma_{uo}, W_2)$, we have $\tilde{y}' \in \tilde{Y}_{W^{\uparrow}}$. Since $W_1 \in \min N(W_2^{\tilde{y}}, \sigma)$, by the definition of $\tilde{\beta}_{W^{\uparrow}}$, we have $\tilde{y}' \in \tilde{\beta}_{W^{\uparrow}}(\tilde{y}, \sigma)$. Then, it holds that $(\tilde{y}', x') \in \tilde{\beta}_{W^{\uparrow}}(\tilde{y}, \sigma) \times \alpha(x, \sigma) = \tilde{\xi}_{W^{\uparrow}}((\tilde{y}, x), \sigma)$. Since $(y', x') \in \xi((y, x), \sigma)$, we have $s\sigma \in L(S\|G)$, $(y', x') \in \xi(Y_0 \times X_0, s\sigma)$, and $(\tilde{y}', x') \in \tilde{\xi}_{W^{\uparrow}}(\tilde{Y}_{W^{\uparrow}0} \times X_0, s\sigma)$. Besides, by $W_2 \subseteq \Phi_{y'}$, it holds that

$$\forall (\tilde{x}'', \tilde{q}'') \in W_2 : ((y', \tilde{x}''), \tilde{q}'') \in \Psi \wedge (y', \tilde{x}'') \in Reach(S\|G).$$

It follows that $((y', x'), (\tilde{y}', x')) \in \tilde{\Phi}$.

We next consider the case where $\sigma \in \Sigma_{uo}$. It follows from $\beta(y, \sigma) \neq \emptyset$ that $\sigma \in \Sigma_{uo}(y)$. Since $((y, x), (\tilde{y}, x)) \in \tilde{\Phi}$, it holds that $\Sigma_{uo}(y) \subseteq \gamma_{uo}^{\tilde{y}} = \Sigma_{uo}(y) \cup (\Sigma_{uc} \cap \Sigma_{uo})$. It follows that $\sigma \in \gamma_{uo}^{\tilde{y}}$. By the definition of $\tilde{\beta}_{W^{\uparrow}}$, we have $\{\tilde{y}\} = \tilde{\beta}_{W^{\uparrow}}(\tilde{y}, \sigma)$, which further implies $(\tilde{y}, x') \in \tilde{\beta}_{W^{\uparrow}}(\tilde{y}, \sigma) \times \alpha(x, \sigma) = \tilde{\xi}_{W^{\uparrow}}((\tilde{y}, x), \sigma)$. Since S is admissible, we have $\{y\} = \beta(y, \sigma)$, that is, $y = y'$. Hence, we have $s\sigma \in L(S\|G)$, $(y, x') \in \xi(Y_0 \times X_0, s\sigma)$, and $(\tilde{y}, x') \in \tilde{\xi}_{W^{\uparrow}}(\tilde{Y}_{W^{\uparrow}0} \times X_0, s\sigma)$. Since $((y, x), (\tilde{y}, x)) \in \tilde{\Phi}$, we have

$$\forall (\tilde{x}, \tilde{q}) \in W_2^{\tilde{y}} : ((y, \tilde{x}), \tilde{q}) \in \Psi \wedge (y, \tilde{x}) \in Reach(S\|G).$$

It follows that $((y, x'), (\tilde{y}, x')) \in \tilde{\Phi}$.

Thus, $\tilde{\Phi}$ is a simulation relation from $S \| G$ to $\tilde{S}_{W^\uparrow} \| G$. It follows from $\tilde{S}_{W^\uparrow} \| G \sqsubseteq S_{W^\uparrow} \| G$ and the transitivity of $\sqsubseteq$ that $S_{W^\uparrow} \in \mathcal{MPS}(G, R)$. $\square$

Since $S_{W^\uparrow} \in \mathcal{A}(\Sigma)$ can be synthesized as long as G is Σ_{uc}-simulated by R, we have

$$\mathcal{S}(G, R) \neq \emptyset \Leftrightarrow \mathcal{MPS}(G, R) \neq \emptyset$$

by Proposition 3.2 and Theorem 3.2. It follows from Proposition 3.1 that, if G is Σ_{uc}-simulated by R, then $S_{W^\uparrow} \| G \in \mathcal{R}_{\sqsubseteq}$, and $S_{W^\uparrow} \| G$ is a supremum of $\mathcal{R}_{\sqsubseteq}$.

Remark 3.5 In [62], the control problem of enforcing simulation equivalence for fully observed nondeterministic DESs was studied. Given the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$, a Σ_{uc}-compatible supervisor (which is essentially equivalent to a Σ_{uc}-admissible supervisor) $S \in \mathcal{A}(\Sigma)$ is synthesized such that $S \| G \sim R$ holds. Such a supervisor S that enforces similarity equivalence, if exists, is a solution to the similarity control problem. Besides, S is a maximally permissive one in $\mathcal{MPS}^{fo}(G, R)$ by the transitivity of $\sqsubseteq$. On the other hand, if the control problem of enforcing simulation equivalence is solvable, we have $\mathcal{S}^{fo}(G, R) \neq \emptyset$ and $\mathcal{MPS}^{fo}(G, R) \neq \emptyset$. In addition, a supervisor $S' \in \mathcal{MPS}^{fo}(G, R)$ satisfies $S' \| G \sim R$.

To the best of the author's knowledge, the control problem of enforcing simulation equivalence under partial observation remains open. Similar to the fully observed case, if this problem is solvable under partial observation, we have $\mathcal{S}(G, R) \neq \emptyset$. In addition, the supervisor $S_{W^\uparrow} \in \mathcal{MPS}(G, R)$ defined by (3.9) is an admissible one such that $S_{W^\uparrow} \| G \sim R$.

3.5 Elimination of Certain Redundant States

Let us recall the maximally permissive supervisor $S_{W^\uparrow} \in \mathcal{A}(\Sigma)$ synthesized in Example 3.2. It holds that $\{y_1, y_2, y_3, y_4\} = \beta_{W^\uparrow}(y_0, a)$, where the control decisions of states y_1, y_2, and y_4 are strictly more permissive than that of y_3. This means that y_3 can be removed from $\beta_{W^\uparrow}(y_0, a)$ without loss of the (maximal) permissiveness of $S_{W^\uparrow}$. In this section, we present a technique for eliminating such redundant states from $S_{W^\uparrow}$.

We consider the supervisor $S_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \beta_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.9). For each $y \in Y_{W\uparrow}$ and each $\sigma \in \Sigma$, we consider a subset $\beta_{red}(y,\sigma) \subseteq \beta_{W\uparrow}(y,\sigma)$, which is the set of states that we remove from $\beta_{W\uparrow}(y,\sigma)$, defined as

$$\begin{aligned}\beta_{red}(y,\sigma) \quad = \quad & \{y' \in \beta_{W\uparrow}(y,\sigma) \mid \Sigma_{W\uparrow}(y') = \emptyset \wedge [\exists y'' \in \beta_{W\uparrow}(y,\sigma) : \\ & \gamma_{uo}^{y'} \subseteq \gamma_{uo}^{y''} \wedge (\Sigma_{W\uparrow}(y'') \neq \emptyset \vee \gamma_{uo}^{y'} \neq \gamma_{uo}^{y''})]\},\end{aligned}$$

where $y' = (W_1^{y'}, \gamma_{uo}^{y'}, W_2^{y'})$ and $y'' = (W_1^{y''}, \gamma_{uo}^{y''}, W_2^{y''})$. Each element $y' \in \beta_{red}(y,\sigma)$ enables no observable events and is less permissive than another state $y'' \in \beta_{W\uparrow}(y,\sigma)$. Note that

$$\forall y \in Y_{W\uparrow}, \forall \sigma \in \Sigma_{uo} : \beta_{red}(y,\sigma) = \emptyset \tag{3.24}$$

holds since $|\beta_{W\uparrow}(y,\sigma)| \leq 1$. Now, we can define a nondeterministic supervisor

$$\bar{S}_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \bar{\beta}_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma), \tag{3.25}$$

with a refined transition function $\bar{\beta}_{W\uparrow} : Y_{W\uparrow} \times \Sigma \to 2^{Y_{W\uparrow}}$ defined as

$$\bar{\beta}_{W\uparrow}(y,\sigma) = \beta_{W\uparrow}(y,\sigma) - \beta_{red}(y,\sigma)$$

for each $y \in Y_{W\uparrow}$ and each $\sigma \in \Sigma$.

By the above definition, the events defined at a state in $\beta_{red}(y,\sigma)$ are strictly fewer than those defined at a state in $\bar{\beta}_{W\uparrow}(y,\sigma)$, and we have the following lemma.

Lemma 3.8 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, we consider the supervisors $\bar{S}_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \bar{\beta}_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ and $S_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \beta_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.25) and (3.9), respectively. It holds that

$$\forall y \in Y_{W\uparrow}, \forall \sigma \in \Sigma, \forall y' \in \beta_{W\uparrow}(y,\sigma), \exists \bar{y} \in \bar{\beta}_{W\uparrow}(y,\sigma) : \gamma_{uo}^{y'} \subseteq \gamma_{uo}^{\bar{y}},$$

where $y' = (W_1^{y'}, \gamma_{uo}^{y'}, W_2^{y'})$ and $\bar{y} = (W_1^{\bar{y}}, \gamma_{uo}^{\bar{y}}, W_2^{\bar{y}})$.

Proof For any $y \in Y_{W\uparrow}$ and any $\sigma \in \Sigma$, we consider any $y' \in \beta_{W\uparrow}(y,\sigma)$. If $y' \notin \beta_{red}(y,\sigma)$, then we have $y' \in \bar{\beta}_{W\uparrow}(y,\sigma) = \beta_{W\uparrow}(y,\sigma) - \beta_{red}(y,\sigma)$ and $\gamma_{uo}^{y'} \subseteq \gamma_{uo}^{y'}$.

We next consider the case where $y' \in \beta_{red}(y,\sigma)$. By the definition of $\beta_{red}(y,\sigma)$, the following two cases are possible: (i) there exists $y'' = (W_1^{y''}, \gamma_{uo}^{y''}, W_2^{y''}) \in \beta_{W\uparrow}(y,\sigma)$ such that $\gamma_{uo}^{y'} \subseteq \gamma_{uo}^{y''} \wedge \Sigma_{W\uparrow}(y'') \neq \emptyset$; or (ii) there does not exist such $y'' \in \beta_{W\uparrow}(y,\sigma)$, but there exists $y''' = (W_1^{y'''}, \gamma_{uo}^{y'''}, W_2^{y'''}) \in \beta_{W\uparrow}(y,\sigma)$ such that $\gamma_{uo}^{y'} \subseteq \gamma_{uo}^{y'''} \wedge \gamma_{uo}^{y'} \neq \gamma_{uo}^{y'''} \wedge \Sigma_{W\uparrow}(y''') = \emptyset$. In case (i), we have $y'' \in \bar{\beta}_{W\uparrow}(y,\sigma)$ and $\gamma_{uo}^{y'} \subseteq \gamma_{uo}^{y''}$. In case (ii), since Γ_{uo} is finite, there exists $\bar{y} = (W_1^{\bar{y}}, \gamma_{uo}^{\bar{y}}, W_2^{\bar{y}}) \in \beta_{W\uparrow}(y,\sigma)$ such that $\gamma_{uo}^{\bar{y}} \in \Gamma_{uo}$ is a maximal element of the set $\{\gamma_{uo}^{\tilde{y}} \in \Gamma_{uo} \mid \exists \tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}}) \in \beta_{W\uparrow}(y,\sigma) : \gamma_{uo}^{y'''} \subseteq \gamma_{uo}^{\tilde{y}} \wedge \Sigma_{W\uparrow}(\tilde{y}) = \emptyset\}$. Then, there does not exist $\bar{y}' = (W_1^{\bar{y}'}, \gamma_{uo}^{\bar{y}'}, W_2^{\bar{y}'}) \in \beta_{W\uparrow}(y,\sigma)$ such that $\gamma_{uo}^{\bar{y}} \subseteq \gamma_{uo}^{\bar{y}'} \wedge (\Sigma_{W\uparrow}(\bar{y}') \neq \emptyset \vee \gamma_{uo}^{\bar{y}} \neq \gamma_{uo}^{\bar{y}'})$. It follows that $\bar{y} \in \bar{\beta}_{W\uparrow}(y,\sigma)$ and $\gamma_{uo}^{y'} \subseteq \gamma_{uo}^{y'''} \subseteq \gamma_{uo}^{\bar{y}}$. □

We prove in the following theorem that $\bar{S}_{W\uparrow}$ is a similarity-enforcing supervisor that has the same permissiveness as $S_{W\uparrow}$.

Theorem 3.3 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $\bar{S}_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \bar{\beta}_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.25) satisfies $\bar{S}_{W\uparrow} \in \mathcal{MPS}(G,R)$.

Proof Let $S_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \beta_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ be the supervisor defined by (3.9). Moreover, $\xi : (Y_{W\uparrow} \times X) \times \Sigma \to 2^{Y_{W\uparrow} \times X}$ and $\bar{\xi} : (Y_{W\uparrow} \times X) \times \Sigma \to 2^{Y_{W\uparrow} \times X}$ denote the nondeterministic transition functions of $S_{W\uparrow} \| G$ and $\bar{S}_{W\uparrow} \| G$, respectively.

We first prove the admissibility of $\bar{S}_{W\uparrow}$. We consider any $(y,x) \in \mathit{Reach}(\bar{S}_{W\uparrow} \| G)$ and $\sigma \in \Sigma_{uc}$ such that $\alpha(x,\sigma) \neq \emptyset$. Since $\bar{S}_{W\uparrow}$ is obtained by removing certain transitions from $S_{W\uparrow}$, we have $(y,x) \in \mathit{Reach}(S_{W\uparrow} \| G)$. By Theorem 3.1, $S_{W\uparrow}$ is admissible, which implies $\xi((y,x),\sigma) \neq \emptyset$. It follows that $\beta_{W\uparrow}(y,\sigma) \neq \emptyset$. By Lemma 3.8, we have $\bar{\beta}_{W\uparrow}(y,\sigma) \neq \emptyset$. Then, we have $\bar{\xi}((y,x),\sigma) = \bar{\beta}_{W\uparrow}(y,\sigma) \times \alpha(x,\sigma) \neq \emptyset$. Hence, $\bar{S}_{W\uparrow}$ satisfies the first condition of admissibility. Besides, for any $y \in Y_{W\uparrow}$ and any $\sigma \in \Sigma_{uo}$, it holds that $\bar{\beta}_{W\uparrow}(y,\sigma) = \beta_{W\uparrow}(y,\sigma)$. It follows that $\bar{S}_{W\uparrow}$ satisfies the second condition of admissibility. Hence, $\bar{S}_{W\uparrow}$ is admissible.

We next prove $\bar{S}_{W\uparrow} \| G \sqsubseteq R$. Since $\bar{S}_{W\uparrow}$ is obtained by removing certain transitions from $S_{W\uparrow}$, we have $\bar{S}_{W\uparrow} \sqsubseteq S_{W\uparrow}$. By the reflexivity of $\sqsubseteq$, we have $G \sqsubseteq G$. Then, it follows from Proposition 2.2 that we have $\bar{S}_{W\uparrow} \| G \sqsubseteq S_{W\uparrow} \| G$. Since the simulation relation $\sqsubseteq$ is transitive, it follows from Theorem 3.1 that $\bar{S}_{W\uparrow} \| G \sqsubseteq R$.

Finally, we prove the maximal permissiveness of $\bar{S}_{W\uparrow}$. By Theorem 3.2 and transitivity of

$\sqsubseteq$, it is sufficient to show $S_{W^\uparrow}\|G \sqsubseteq \bar{S}_{W^\uparrow}\|G$. Besides, by Proposition 2.2 and $G \sqsubseteq G$, we only need to show $S_{W^\uparrow} \sqsubseteq \bar{S}_{W^\uparrow}$. For this purpose, we show the relation $\bar{\Phi} \subseteq Y_{W^\uparrow} \times Y_{W^\uparrow}$ defined as

$$\begin{aligned}\bar{\Phi} \;=\; & \{(y,\bar{y}) \in Y_{W^\uparrow} \times Y_{W^\uparrow} \mid y = \bar{y} \vee [\Sigma_{W^\uparrow}(y) = \emptyset \wedge \\ & \gamma_{uo}^{y} \subseteq \gamma_{uo}^{\bar{y}} \wedge (\exists y'' \in Y_{W^\uparrow}, \exists \sigma \in \Sigma : y, \bar{y} \in \beta_{W^\uparrow}(y'', \sigma))]\}\end{aligned}$$

is a simulation relation from $S_{W^\uparrow}$ to $\bar{S}_{W^\uparrow}$, where $y = (W_1^y, \gamma_{uo}^y, W_2^y)$ and $\bar{y} = (W_1^{\bar{y}}, \gamma_{uo}^{\bar{y}}, W_2^{\bar{y}})$.

- For any $y_0 \in Y_{W^\uparrow 0}$, we have $(y_0, y_0) \in \bar{\Phi}$.
- For any $(y, \bar{y}) \in \bar{\Phi}$ and any $\sigma \in \Sigma$, we consider any $y' \in \beta_{W^\uparrow}(y, \sigma) \neq \emptyset$.

  In the case where $y = \bar{y}$ and $y' \notin \beta_{red}(y, \sigma)$, we have $y' \in \beta_{W^\uparrow}(y,\sigma) - \beta_{red}(y,\sigma) = \bar{\beta}_{W^\uparrow}(y,\sigma)$. It follows that $(y', y') \in \bar{\Phi}$.

  We next consider the case where $y = \bar{y}$ and $y' \in \beta_{red}(y,\sigma) \subseteq \beta_{W^\uparrow}(y,\sigma)$. Then, we have $\Sigma_{W^\uparrow}(y') = \emptyset$. By Lemma 3.8, there exists $\bar{y}' = (W_1^{\bar{y}'}, \gamma_{uo}^{\bar{y}'}, W_2^{\bar{y}'}) \in \bar{\beta}_{W^\uparrow}(y,\sigma) \subseteq \beta_{W^\uparrow}(y,\sigma)$ such that $\gamma_{uo}^{y'} \subseteq \gamma_{uo}^{\bar{y}'}$. It follows that $(y', \bar{y}') \in \bar{\Phi}$.

  In the case where $y \neq \bar{y}$, by the definition of $\bar{\Phi}$, we have $\Sigma_{W^\uparrow}(y) = \emptyset$ and $\gamma_{uo}^y \subseteq \gamma_{uo}^{\bar{y}}$, which imply $\sigma \in \gamma_{uo}^y \subseteq \gamma_{uo}^{\bar{y}} \subseteq \Sigma_{uo}$. Then, we have $\beta_{red}(\bar{y},\sigma) = \emptyset$. By the definition of $\beta_{W^\uparrow}$, we have $\{y\} = \beta_{W^\uparrow}(y,\sigma)$ and $\{\bar{y}\} = \beta_{W^\uparrow}(\bar{y},\sigma)$. It follows that $\{\bar{y}\} = \beta_{W^\uparrow}(\bar{y},\sigma) - \beta_{red}(\bar{y},\sigma) = \bar{\beta}_{W^\uparrow}(\bar{y},\sigma)$ and $(y', \bar{y}) = (y, \bar{y}) \in \bar{\Phi}$.

Thus, we have $S_{W^\uparrow}\|G \sqsubseteq \bar{S}_{W^\uparrow}\|G$. □

Example 3.5 We consider the system $G \in \mathcal{A}(\Sigma)$, the specification $R \in \mathcal{A}(\Sigma)$, and the supervisor $S_{W^\uparrow} \in \mathcal{A}(\Sigma)$ shown in Figures 3.3 (a), (b), and Figure 3.4, respectively. Then, the supervisor $\bar{S}_{W^\uparrow} = (Y_{W^\uparrow}, \Sigma, \bar{\beta}_{W^\uparrow}, Y_{W^\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.25) is synthesized. Since no event is defined at y_3, while b is defined at y_1 and y_2, and c is defined at y_4, the transition from y_0 to y_3 is removed. The reachable part of the supervisor $\bar{S}_{W^\uparrow}$ and the supervised system $\bar{S}_{W^\uparrow}\|G$ are shown in Figures 3.7 (a) and (b), respectively. We can verify that $\bar{S}_{W^\uparrow}$ is admissible and it holds that $\bar{S}_{W^\uparrow}\|G \sqsubseteq R$. By Theorems 3.2 and 3.3, both $S_{W^\uparrow}$ and $\bar{S}_{W^\uparrow}$ are maximally permissive.

Remark 3.6 The supervisor $\bar{S}_{W^\uparrow} \in \mathcal{A}(\Sigma)$ can be directly constructed without synthesizing $S_{W^\uparrow} \in \mathcal{A}(\Sigma)$ or $\tilde{S}_{W^\uparrow} \in \mathcal{A}(\Sigma)$ first. As it is hard to directly prove the maximal permissiveness of $\bar{S}_{W^\uparrow}$, we first synthesized $S_{W^\uparrow} \in \mathcal{A}(\Sigma)$ in Section 3.3.

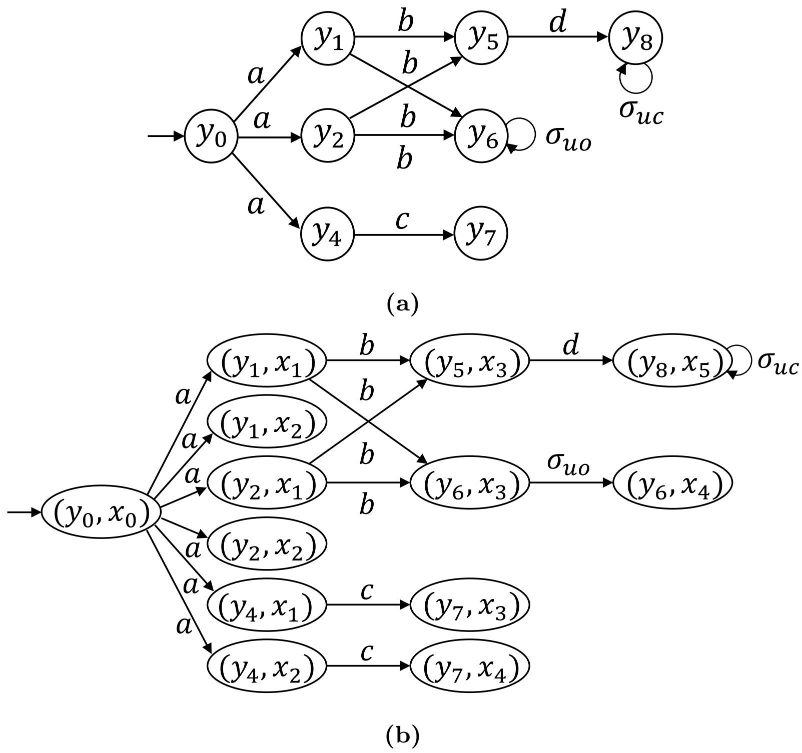


Figure 3.7: (a) Supervisor $\bar{S}_{W^\uparrow}$ and (b) supervised system $\bar{S}_{W^\uparrow} \| G$.

Example 3.6 We consider the similarity control problem for the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$ shown in Figures 3.1 (b) and (c), respectively. As discussed in Subsection 3.1.2 the supervisor S_{det} shown in Figure 3.2 (c) does not solve the problem.

The unique maximal fixpoint $W^\uparrow$ of the the function $F : 2^{X\times Q} \to 2^{X\times Q}$ defined by (3.3) is computed as

$$\begin{aligned}
F^1(X \times Q) &= X \times Q - \{x_0, x_3, x_6\} \times \{q_1, q_2, q_4, q_5\}, \\
F^2(X \times Q) &= F^1(X \times Q) - \{(x_3, q_0), (x_6, q_0)\} \\
&= F^3(X \times Q) \\
&= W^\uparrow.
\end{aligned}$$

The supervisor $\bar{S}_{W^\uparrow}$ defined by (3.25) is synthesized based on $W^\uparrow$, whose reachable part is

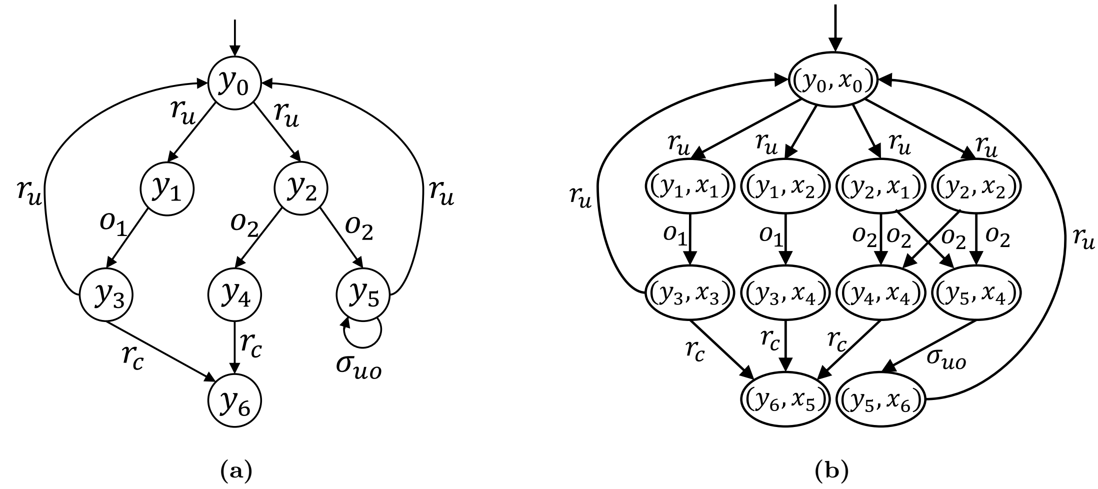


Figure 3.8: (a) Supervisor $\bar{S}_{W^\uparrow}$ and (b) supervised system $\bar{S}_{W^\uparrow} \| G$.

shown in Figure 3.8 (a), where

$$
\begin{aligned}
y_0 &= (\{(x_0, q_0)\}, \emptyset, \{(x_0, q_0)\}), \\
y_1 &= (\{(x_1, q_1), (x_2, q_1)\}, \emptyset, \{(x_1, q_1), (x_2, q_1)\}), \\
y_2 &= (\{(x_1, q_2), (x_2, q_2)\}, \emptyset, \{(x_1, q_2), (x_2, q_2)\}), \\
y_3 &= (\{(x_3, q_3), (x_4, q_3)\}, \emptyset, \{(x_3, q_3), (x_4, q_3)\}), \\
y_4 &= (\{(x_4, q_4)\}, \emptyset, \{(x_4, q_4)\}), \\
y_5 &= (\{(x_4, q_4)\}, \{uo\}, \{(x_4, q_4), (x_6, q_6)\}), \\
y_6 &= (\{(x_5, q_5)\}, \emptyset, \{(x_5, q_5)\}).
\end{aligned}
$$

The reachable part of the supervised system $\bar{S}_{W^\uparrow} \| G$ is shown in Figure 3.8 (b). We can verify that $\bar{S}_{W^\uparrow}$ solves the similarity control problem. Theorem 3.3 guarantees that $\bar{S}_{W^\uparrow} \in \mathcal{MPS}(G, R)$.

3.6 Conclusion

In this chapter, we considered the similarity control problem under partial observation, where both the system and the specification are modeled as NFAs. The existence condition of a similarity-enforcing supervisor under full observation was extended to the case of partial observation. We developed a method to synthesize such a similarity-enforcing supervisor when the existence condition is satisfied and showed that it is a maximal solution to the similarity control problem. Such a maximal solution forms a supremal feasible subspecification of the given specification. Finally, we provided a technique that eliminates certain redundant states from the synthesized supervisor without reducing its permissiveness.

Chapter 4

Maximally Permissive Nonblocking Similarity Control

In this chapter, we consider the nonblocking similarity control problem, where both enforcing similarity and imposing nonblockingness are required. For the given nondeterministic system and specification, we synthesize a nonblocking supervisor that enforces similarity. In order to construct such a supervisor, we propose an algorithm that computes a nonblocking supervisor from a possibly blocking one by iteratively removing certain states. This algorithm is proved to be valid under both full and partial observation. Then, we identify two key properties of input supervisors, named state-unmergedness and strong maximal permissiveness, which together guarantee the maximal permissiveness of output nonblocking supervisors. The algorithm is applied to one of the maximally permissive supervisors synthesized in Chapter 3, which has these two properties, to obtain a maximally permissive nonblocking supervisor. In addition, we show that a nonblocking supervisor is generated by the algorithm if and only if there exists a solution to the nonblocking similarity control problem.

4.1 Problem Formulation

Given the (possibly partially observed) system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$, the nonblocking similarity control problem requires us to synthesize an admissible and nonblocking supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ such that $S \| G \sqsubseteq R$ holds. Such a supervisor is called a nonblocking similarity-enforcing supervisor.

We let

$$\mathcal{NS}(G,R) = \{S \in \mathcal{A}(\Sigma) \mid S \text{ is admissible and nonblocking, and satisfies } S\|G \sqsubseteq R\}$$

be the set of all nonblocking similarity-enforcing supervisors for G and R. By the definition of $\mathcal{NS}(G,R)$, it holds that $\mathcal{NS}(G,R) \subseteq \mathcal{S}(G,R)$.

Similar to the similarity control problem studied in Chapter 3, we define the set $\mathcal{MPNS}(G,R) \subseteq \mathcal{NS}(G,R)$ of all maximally permissive nonblocking similarity-enforcing supervisors (based on the simulation relation) as

$$\mathcal{MPNS}(G,R) = \{S \in \mathcal{NS}(G,R) \mid \forall S' \in \mathcal{NS}(G,R) : S'\|G \sqsubseteq S\|G\}.$$

In this chapter, we synthesize a supervisor that is an element of $\mathcal{MPNS}(G,R)$.

We use $\mathcal{NS}^{fo}(G,R)$ and $\mathcal{MPNS}^{fo}(G,R)$ to denote the set of solutions and the set of maximal solutions to the nonblocking similarity control problem, respectively, when the system under consideration is fully observed.

For $\mathcal{MPNS}(G,R)$, we have similar arguments as discussed in Subsection 3.1.1. A supervisor $S \in \mathcal{MPNS}(G,R)$ is maximally permissive in a global sense, while S in $\mathcal{MPNS}(G,R)$ is not unique.

A maximally permissive nonblocking similarity-enforcing supervisor $S \in \mathcal{MPNS}(G,R)$ also serves as a supervisor that enforces bisimilarity to a supremal feasible nonblocking subspecification of R. Let $\mathcal{R}_{\sqsubseteq}^{NB} \subseteq \mathcal{A}(\Sigma)$ be the set of all nonblocking subspecifications of R for which the bisimilarity control problem is solvable. Namely,

$$\begin{aligned}\mathcal{R}_{\sqsubseteq}^{NB} \;=\; &\{R' \in \mathcal{A}(\Sigma) \mid R' \text{ is nonblocking, } R' \sqsubseteq R \text{ holds, and} \\ &\text{there exists an admissible supervisor } S \in \mathcal{A}(\Sigma) \text{ such that } S\|G \simeq R'\}.\end{aligned}$$

Then, Proposition 3.1 can be extended as follows by taking nonblockingness into consideration.

Proposition 4.1 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$ such that $\mathcal{MPNS}(G,R) \neq \emptyset$, we consider any $S \in \mathcal{MPNS}(G,R)$. It holds that $S\|G \in \mathcal{R}_{\sqsubseteq}^{NB}$, and $S\|G$ is a supremum of $\mathcal{R}_{\sqsubseteq}^{NB}$.

Proof For any $S \in \mathcal{MPNS}(G, R)$, $S \| G$ is nonblocking and $S \| G \sqsubseteq R$ holds. It follows from the reflexivity of $\simeq$ that $S \| G \simeq S \| G$. Since S is admissible, we have $S \| G \in \mathcal{R}_{\sqsubseteq}^{NB}$.

To show that $S \| G$ is a supremum of $\mathcal{R}_{\sqsubseteq}^{NB}$, we first consider any $R' \in \mathcal{R}_{\sqsubseteq}^{NB}$. There exists an admissible supervisor S' such that $S' \| G \simeq R' \sqsubseteq R$. It follows that $S' \| G \sqsubseteq R$. Since R' is nonblocking, S' is nonblocking. Hence, we have $S' \in \mathcal{NS}(G, R)$. By $S \in \mathcal{MPNS}(G, R)$, we have $S' \| G \sqsubseteq S \| G$. By $S' \| G \simeq R'$ and the transitivity of $\sqsubseteq$, we have $R' \sqsubseteq S \| G$.

We consider any $\hat{R} \in \mathcal{A}(\Sigma)$ such that

$$\forall R'' \in \mathcal{R}_{\sqsubseteq}^{NB} : R'' \sqsubseteq \hat{R}.$$

It follows from $S \| G \in \mathcal{R}_{\sqsubseteq}^{NB}$ that $S \| G \sqsubseteq \hat{R}$. Hence, $S \| G$ is a supremum of $\mathcal{R}_{\sqsubseteq}^{NB}$. $\square$

Remark 4.1 The state elimination technique presented in Section 3.5 removes some states in $S_{W^{\uparrow}}$ that possibly cause deadlock in the supervised system. However, this resulting supervisor is not necessarily nonblocking.

4.2 Construction of Nonblocking Supervisors

As a nonblocking similarity-enforcing supervisor in $\mathcal{NS}(G, R)$ is difficult to be directly synthesized, we solve the nonblocking similarity control problem by removing the states of blocking supervisors which cause blockingness.

Here, we define the following notion of the subautomaton with respect to a subset of the state set, which is used in the construction of a new automaton when some states of the original one are removed.

Definition 4.1 For $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and a subset $X' \subseteq X$ such that $X_0 \cap X' \neq \emptyset$, the subautomaton $G' \in \mathcal{A}(\Sigma)$ of G with respect to X' is defined as

$$G' = (X', \Sigma, \alpha', X'_0, X'_m) \in \mathcal{A}(\Sigma),$$

where $X'_0 = X_0 \cap X'$ and $X'_m = X_m \cap X'$. The nondeterministic transition function $\alpha' : X' \times \Sigma \to 2^{X'}$ is given as $\alpha'(x, \sigma) = \alpha(x, \sigma) \cap X'$ for each $x \in X' \subseteq X$ and each $\sigma \in \Sigma$.

By the above definition, a subautomaton $G' \in \mathcal{A}(\Sigma)$ of some $G \in \mathcal{A}(\Sigma)$ is simulated by the original G.

Proposition 4.2 For $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and a subset $X' \subseteq X$ such that $X_0 \cap X' \neq \emptyset$, the subautomaton $G' = (X', \Sigma, \alpha', X_0', X_m') \in \mathcal{A}(\Sigma)$ of G with respect to X' satisfies $G' \sqsubseteq G$.

Proof By the definition of the subautomaton, it can be easily shown that the relation $\Phi \subseteq X' \times X' \subseteq X' \times X$ given as

$$\Phi = \{(x', x) \in X' \times X \mid x' = x\}$$

is a simulation relation from G' to G. □

4.2.1 State Removal Algorithm NBSR

In this subsection, we introduce an algorithm named NBSR (NonBlocking State Removal), which iteratively removes states of a supervisor that violate the first condition of admissibility or cause blockingness. The algorithm NBSR computes a nonblocking supervisor from a possibly blocking one.

In the ith iteration, whether or not the supervisor $S^i = (Y^i, \Sigma, \beta^i, Y_0^i, Y^i) \in \mathcal{A}(\Sigma)$ is nonblocking and satisfies the first condition of admissibility is verified. The synchronous composition $S^i \| G$ of S^i and G is constructed in line 4, where, for each $(y, x) \in Y^i \times X$ and each $\sigma \in \Sigma$, $\xi^i((y, x), \sigma) = \beta^i(y, \sigma) \times \alpha(x, \sigma)$ is computed. Since $|Y^i \times X|$, $|\beta^i(y, \sigma)|$, and $|\alpha(x, \sigma)|$ are at most $|Y||X|$, $|Y|$, and $|X|$, respectively, the computational complexity of constructing $S^i \| G$ is $O(|Y|^2|X|^2|\Sigma|)$. In lines 5–9, each reachable state (y, x) of $S^i \| G$ is examined to decide whether y should be removed or not. Line 6 verifies if y violates the first condition of admissibility or causes blockingness. The condition

$$\forall s \in \Sigma^* : \xi^i((y, x), s) \cap Y^i \times X_m = \emptyset$$

is equivalent to the condition that any marked state of $S^i \| G$ is not reachable from (y, x). Since $S^i \| G$ is finite, it can be evaluated algorithmically by exploring all the states reachable from (y, x). This requires executing all the transitions from (y, x) in $S^i \| G$, whose number is bounded by $|X|^2|Y|^2|\Sigma|$. Then, the computational complexity in lines 5–9 is $O(|Y|^3|X|^3|\Sigma|)$.

Algorithm 1 NBSR

Require: $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma), S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$

$S^1 = (Y^1, \Sigma, \beta^1, Y_0^1, Y^1) \leftarrow S, S^{NB} \leftarrow \text{null}, i \leftarrow 1$
while $S^{NB} = \text{null}$ **do**
  $Y^{i+1} \leftarrow Y^i$
  Compute $S^i \| G = (Y^i \times X, \Sigma, \xi^i, Y_0^i \times X_0, Y^i \times X_m)$
  for all $(y, x) \in \mathit{Reach}(S^i \| G)$ **do**
    if $[\exists \sigma \in \Sigma_{uc} : \alpha(x, \sigma) \neq \emptyset \wedge \beta^i(y, \sigma) = \emptyset] \vee [\forall s \in \Sigma^* : \xi^i((y, x), s) \cap Y^i \times X_m = \emptyset]$ **then**
      $Y^{i+1} \leftarrow Y^{i+1} - \{y\}$
    end if
  end for
  if $Y^{i+1} = Y^i$ **then**
    $S^{NB} \leftarrow S^i$
  else if $Y^{i+1} \cap Y_0^i = \emptyset$ **then**
    $S^{NB} \leftarrow \text{undefined}$
  else
    Compute the subautomaton $S^{i+1} = (Y^{i+1}, \Sigma, \beta^{i+1}, Y_0^{i+1}, Y^{i+1})$ of S^i with respect to Y^{i+1}
    $i \leftarrow i + 1$
  end if
end while
return S^{NB}

The set of remaining states can be written as

$$\begin{aligned} NB_G^{S^i}(Y^i) &= \{y \in Y^i \mid \forall (y, x) \in \mathit{Reach}(S^i \| G) : [\forall \sigma \in \Sigma_{uc} : \alpha(x, \sigma) \neq \emptyset \Rightarrow \\ &\quad \beta^i(y, \sigma) \neq \emptyset] \wedge [\exists s \in \Sigma^* : \xi^i((y, x), s) \cap Y^i \times X_m \neq \emptyset]\}, \end{aligned} \tag{4.1}$$

namely, $Y^{i+1} = NB_G^{S^i}(Y^i)$. If $Y^{i+1} = Y^i$, which implies that S^i is nonblocking, then the algorithm outputs the supervisor S^i. If $Y^{i+1} \neq Y^i$ and $Y^{i+1} \cap Y_0^i = \emptyset$, the subautomaton of S^i with respect to Y^{i+1} cannot be constructed, so the algorithm outputs "undefined". In the case where $Y^{i+1} \neq Y^i$ and $Y^{i+1} \cap Y_0^i \neq \emptyset$, the subautomaton S^{i+1} of S^i with respect to Y^{i+1} is constructed, where, for each $y' \in Y^{i+1}$ and each $\sigma' \in \Sigma$, $\beta^{i+1}(y', \sigma')$ is computed. The computational complexity of constructing S^{i+1} is $O(|Y|^2|\Sigma|)$. Although the "bad" states in $Y^i - Y^{i+1}$ are removed from S^i to form S^{i+1}, such a state removal operation on S^i might eliminate transitions caused by uncontrollable events, and/or might block some path of $S^i \| G$ that leads to marked states. Hence, S^{i+1} does not necessarily satisfy the first condition of admissibility or is possibly blocking, so the operation should be iteratively performed. Since the number of states of the supervisor S^i monotonically decreases, the algorithm NBSR terminates within at most $|Y|$ iterations. Therefore, the complexity of the algorithm NBSR is $O(|Y|^4|X|^3|\Sigma|)$.

4.2.2 Validity of NBSR

We show the validity of NBSR in this subsection. For the sake of notational simplicity, we let NBSR(S, G) denote the output of the algorithm NBSR when it is applied to a supervisor $S \in \mathcal{A}(\Sigma)$. Note that NBSR$(S, G)$ is either a supervisor $S^{NB} \in \mathcal{A}(\Sigma)$ or "undefined". By the definition of $NB^S_G(Y)$, we have the following lemma.

Lemma 4.1 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$, a supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ satisfies the first condition of admissibility and is nonblocking if and only if $NB^S_G(Y) = Y$.

Proof Let $\xi : (Y \times X) \times \Sigma \to 2^{Y \times X}$ be the transition function of $S \| G$.

($\Rightarrow$) By (4.1), we have $NB^S_G(Y) \subseteq Y$. To show $Y \subseteq NB(Y)^S_G$, for any $y \in Y$, we consider any $(y, x) \in \mathit{Reach}(S\|G)$. Since S satisfies the first condition of admissibility, for any $\sigma \in \Sigma_{uc}$ such that $\alpha(x, \sigma) \neq \emptyset$, we have $\beta(y, \sigma) \neq \emptyset$. Besides, there exists $s \in \Sigma^*$ such that $\xi((y, x), s) \cap Y \times X_m \neq \emptyset$ by the nonblockingness of S. It follows that $y \in NB^S_G(Y)$.

($\Leftarrow$) We consider any $(y, x) \in \mathit{Reach}(S\|G)$. Since $y \in Y = NB^S_G(Y)$, there exists $s \in \Sigma^*$ such that $\xi((y, x), s) \cap Y \times X_m \neq \emptyset$. It follows that S is nonblocking. Besides, for any $\sigma \in \Sigma_{uc}$ such that $\alpha(x, \sigma) \neq \emptyset$, we have $\beta(y, \sigma) \neq \emptyset$. Hence, S satisfies the first condition of admissibility. $\square$

Then, we show in the following proposition that a solution to the nonblocking similarity control problem can be computed by applying NBSR to an appropriate supervisor.

Proposition 4.3 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$, we consider a supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ that satisfies the second condition of admissibility and $S\|G \sqsubseteq R$. If NBSR$(S, G) = S^{NB} \in \mathcal{A}(\Sigma)$, then $S^{NB} \in \mathcal{NS}(G, R)$.

Proof Since S^{NB} is a subautomaton of S, we have $S^{NB} \sqsubseteq S$ by Proposition 4.2. Since the order $\sqsubseteq$ on $\mathcal{A}(\Sigma)$ is reflexive, $G \sqsubseteq G$ holds. It follows from Proposition 2.2 that $S^{NB}\|G \sqsubseteq S\|G$. By $S\|G \sqsubseteq R$ and the transitivity of $\sqsubseteq$, we have $S^{NB}\|G \sqsubseteq R$. By Lemma 4.1,

$S^{NB} = \text{NBSR}(S, G) \in \mathcal{A}(\Sigma)$ satisfies the first condition of admissibility and is nonblocking. Since S satisfies the second condition of admissibility, so does S^{NB} by the Definition 4.1 of subautomata. It follows that $S^{NB} \in \mathcal{NS}(G, R)$. $\square$

We provide the following example to illustrate how the algorithm NBSR works.

Example 4.1 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$ shown in Figures 4.1 (a) and (b), respectively, where $\Sigma = \{a, b, c, d, e, u\}$ and $\Sigma_{uc} = \Sigma_{uo} = \{u\}$, we consider a $\Sigma_{uc} \cap \Sigma_{uo}$-compatible supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ shown in Figure 4.1 (c). Here, at each state of S, a self-loop by u is defined but omitted for the sake of simplicity. The reachable part of the supervised system $S \| G$ is shown in Figure 4.1 (d), from which we can verify that $S \in \mathcal{S}(G, R)$ holds, but $S \notin \mathcal{NS}(G, R)$ as it is blocking.

We apply the algorithm NBSR to S. In the first iteration, we have $S^1 = (Y^1, \Sigma, \beta^1, Y_0^1, Y^1) = S$. Since any marked state cannot be reached from (y_6, x_6), state y_6 of S^1 is removed. Then, the supervisor $S^2 \in \mathcal{A}(\Sigma)$ is constructed as the subautomaton of S^1 with respect to $Y^2 = Y^1 - \{y_6\}$. The supervised system $S^2 \| G$ is obtained by eliminating states (y_6, x_6) and (y_6, x_7) from $S \| G$. Since the unique marked successor state (y_6, x_7) of state (y_3, x_4) is eliminated, the supervisor S^2 is still blocking. In the second iteration, state y_3 of S^2 is removed, and the subautomaton $S^3 \in \mathcal{A}(\Sigma)$ of S^2 with respect to $Y^3 = Y^2 - \{y_3\}$ shown in Figure 4.1 (e) is constructed. The reachable part of the supervised system $S^3 \| G$ is shown in Figure 4.1 (f). In the third iteration, any state of S^3 is not removed, since no state of S^3 causes blockingness or violates the first condition of admissibility. Finally, the algorithm outputs the supervisor $S^{NB} = S^3 \in \mathcal{A}(\Sigma)$, which is a solution $S^{NB} \in \mathcal{NS}(G, R)$ to the nonblocking similarity control problem.

4.3 Synthesis of the Input Supervisor

Proposition 4.3 has shown conditions on an input supervisor $S \in \mathcal{A}(\Sigma)$ of the algorithm NBSR, which guarantee that if the output $\text{NBSR}(S, G)$ is $S^{NB} \in \mathcal{A}(\Sigma)$, then it is a solution in $\mathcal{NS}(G, R)$ to the nonblocking similarity control problem. To synthesize an appropriate candidate supervisor to which the algorithm NBSR is applied, we recall the results presented in Chapter 3 and modify them by taking marked states into account.

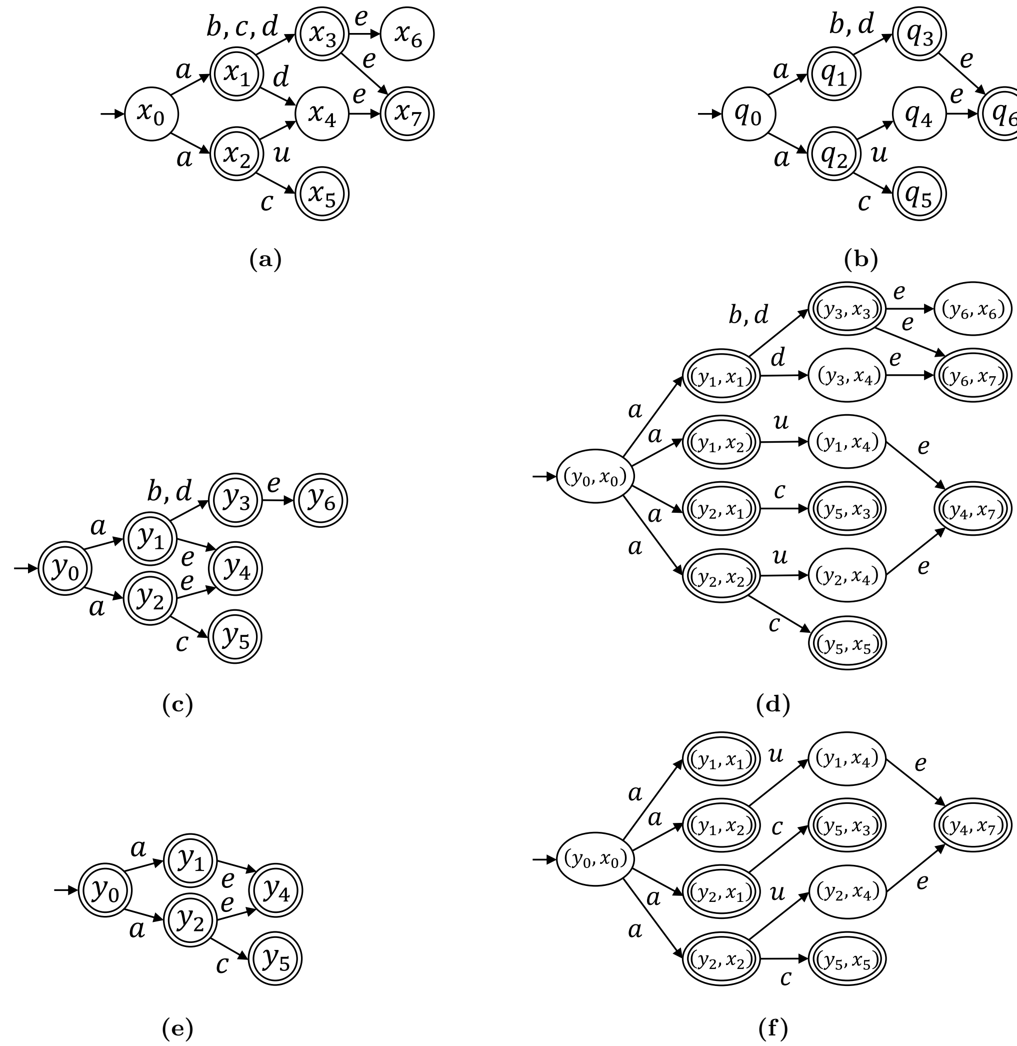


Figure 4.1: (a) System G, (b) specification R, (c) supervisor S, (d) supervised system $S \| G$, (e) supervisor $S^3 = S^{NB}$, and (f) supervised system $S^3 \| G = S^{NB} \| G$.

The notion of Σ_{uc}-similarity is extended by taking marking into consideration as follows.

Definition 4.2 [66] For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$, a relation $\Phi \subseteq X \times Q$ is said to be a Σ_{uc}-simulation relation from G to R if

- $\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in \Phi$,
- $\forall (x, q) \in \Phi, \forall \sigma \in \Sigma_{uc}, \forall x' \in \alpha(x, \sigma), \exists q' \in \delta(q, \sigma) : (x', q') \in \Phi$, and
- $\forall (x, q) \in \Phi : x \in X_m \Rightarrow q \in Q_m$.

The additional third condition corresponds to that of a simulation relation. For any pair (x, q) of a Σ_{uc}-simulation relation, q should be marked if x is marked. The following proposition extends Proposition 3.2.

Proposition 4.4 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$, $\mathcal{S}(G, R) \neq \emptyset$ if and only if G is Σ_{uc}-simulated by R.

Proof ($\Leftarrow$) Let $S_u = (Y_u, \Sigma, \beta_u, Y_{u0}, Y_u) \in \mathcal{A}(\Sigma)$ be defined in the same manner as (3.1) with all states marked. By sufficiency part of the proof of Proposition3.2, in order to show $S_u \in \mathcal{S}(G, R)$, we only need to show that the relation

$$\Psi_u = \{((y_0, x), q) \in (Y_u \times X) \times Q \mid (x, q) \in \Phi_u\}$$

satisfies the third condition of a simulation relation from $S_u \| G$ to R, where $\Psi_u \subseteq X \times Q$ is a Σ_{uc}-simulation relation from G to R.

For any $((y_0, x), q) \in \Psi_u$ such that $(y_0, x) \in Y_u \times X_m$, since $(x, q) \in \Phi_u$ and Φ_u is a Σ_{uc}-simulation relation from G to R, we have $q \in Q_m$. It follows that $S_u \in \mathcal{S}(G, R) \neq \emptyset$.

($\Rightarrow$) Let $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ be any supervisor such that $S \in \mathcal{S}(G, R) \neq \emptyset$. By necessity part of the proof of Proposition3.2, we only need to show that the relation

$$\Phi = \{(x, q) \in X \times Q \mid \exists y \in Y : ((y, x), q) \in \Psi \wedge (y, x) \in \mathit{Reach}(S \| G)\} \tag{4.2}$$

satisfies the third condition of a Σ_{uc}-simulation relation from G to R, where $\Psi \subseteq (Y \times X) \times Q$ is an arbitrary simulation relation from $S \| G$ to R.

For any $(x, q) \in \Phi$ such that $x \in X_m$, there exists $y \in Y$ such that $((y, x), q) \in \Psi$. Since Ψ is a simulation relation from $S \| G$ to R, we have $q \in Q_m$. Hence, Φ is a Σ_{uc}-simulation relation from G to R and G is Σ_{uc}-simulated by R. $\square$

The verification of Σ_{uc}-similarity requires the following function $\tilde{F} : 2^{X \times Q} \to 2^{X \times Q}$ defined as

$$\begin{aligned} \tilde{F}(W) \;=\; & \{(x, q) \in W \mid [\forall \sigma \in \Sigma_{uc}, \forall x' \in \alpha(x, \sigma), \exists q' \in \delta(q, \sigma) : (x', q') \in W] \wedge \\ & [x \in X_m \Rightarrow q \in Q_m]\}, \end{aligned} \tag{4.3}$$

where $W \in 2^{X \times Q}$ is an arbitrary relation over $X \times Q$. Note that $\tilde{F}$ is obtained by imposing the additional condition

$$x \in X_m \Rightarrow q \in Q_m$$

on the function $F : 2^{X\times Q} \to 2^{X\times Q}$ defined in (3.3). Trivially, $\tilde{F}$ is monotone and there exists the unique maximal fixpoint $W^{\uparrow}$ of $\tilde{F}$ by Tarski's fixpoint theorem [71]. The unique maximal fixpoint $W^{\uparrow} \in 2^{X\times Q}$ of $\tilde{F}$ is computed by

$$W^{\uparrow} = \bigcap_{i\geq 0} \tilde{F}^i(X \times Q).$$

The complexity of this computation is $O(|\Sigma_{uc}||X|^3|Q|^3)$ [66].

The following extension of Proposition 3.3 is straightforward.

Proposition 4.5 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$, the following two statements hold for the function $\tilde{F} : 2^{X\times Q} \to 2^{X\times Q}$ defined by (4.3).

- For any $\Phi \subseteq X \times Q$, Φ is a Σ_{uc}-simulation relation from G to R if and only if $\Phi = \tilde{F}(\Phi)$ and
$$\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in \Phi.$$
- G is Σ_{uc}-simulated by R if and only if the unique maximal fixpoint $W^{\uparrow}$ of the function $\tilde{F}$ satisfies
$$\forall x_0 \in X_0, \exists q_0 \in Q_0 : (x_0, q_0) \in W^{\uparrow}.$$

Since $\mathcal{NS}(G, R) \subseteq \mathcal{S}(G, R)$, by Proposition 4.4, Σ_{uc}-similarity is a necessary condition for $\mathcal{NS}(G, R) \neq \emptyset$. That is, if G is not Σ_{uc}-simulated by R, then the nonblocking similarity control problem is not solvable. Hereafter, we assume that G is Σ_{uc}-simulated by R, so that we have $\mathcal{S}(G, R) \neq \emptyset$.

Using the unique maximal fixpoint $W^{\uparrow} \in 2^{X\times Q}$ of $\tilde{F} : 2^{X\times Q} \to 2^{X\times Q}$ defined by (4.3), we can synthesize a nondeterministic supervisor

$$\tilde{S}_{W^{\uparrow}} = (\tilde{Y}_{W^{\uparrow}}, \Sigma, \tilde{\beta}_{W^{\uparrow}}, \tilde{Y}_{W^{\uparrow}0}, \tilde{Y}_{W^{\uparrow}}) \in \mathcal{A}(\Sigma) \tag{4.4}$$

in the same way as in (3.13), whose states are all marked.

We show in the following proposition that $\tilde{S}_{W^\uparrow}$ is a maximally permissive similarity-enforcing supervisor (in the case where marking is considered).

Proposition 4.6 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $\tilde{S}_{W^\uparrow} = (\tilde{Y}_{W^\uparrow}, \Sigma, \tilde{\beta}_{W^\uparrow}, \tilde{Y}_{W^\uparrow 0}, \tilde{Y}_{W^\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (4.4) satisfies $\tilde{S}_{W^\uparrow} \in \mathcal{MPS}(G, R)$.

Proof The results of Corollary 3.1 also hold for $\tilde{S}_{W^\uparrow}$ (defined by (4.4)), and the admissibility of $\tilde{S}_{W^\uparrow}$ can be proved in the same manner as that of $S_{W^\uparrow} \in \mathcal{A}(\Sigma)$ defined by (3.9) in Theorem 3.1.

We prove $\tilde{S}_{W^\uparrow} \| G \sqsubseteq R$ by showing that the relation $\Psi \subseteq (\tilde{Y}_{W^\uparrow} \times X) \times Q$ defined as

$$\Psi = \{((y, x), q) \in (\tilde{Y}_{W^\uparrow} \times X) \times Q \mid (y, x) \in \mathit{Reach}(\tilde{S}_{W^\uparrow} \| G) \wedge (x, q) \in W_2^y\}$$

is a simulation relation from $\tilde{S}_{W^\uparrow} \| G$ to R, where $y = (W_1^y, \gamma_{uo}^y, W_2^y)$. The first and second conditions of the simulation relation are proved in the same way in the proof $S_{W^\uparrow} \| G \sqsubseteq R$ in Theorem 3.1. To prove the third condition, we consider any $((y, x), q) \in \Psi$ such that $(y, x) \in \tilde{Y}_{W^\uparrow} \times X_m$. It follows that $x \in X_m$ and $(x, q) \in W_2^y \subseteq W^\uparrow$. Since $W^\uparrow$ is a Σ_{uc}-simulation relation from G to R, we have $q \in Q_m$.

To show the maximal permissiveness of $\tilde{S}_{W^\uparrow}$, we consider any admissible supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{S}(G, R)$. By the proof of Theorem 3.2, it is sufficient to show that the relation $\tilde{\Phi} \subseteq (Y \times X) \times (\tilde{Y}_{W^\uparrow} \times X)$ defined by (3.19) satisfies the third condition of the simulation relation. For any $((y, x), (\tilde{y}, \tilde{x})) \in \tilde{\Phi}$, we have $x = \tilde{x}$. It follows that

$$(y, x) \in Y \times X_m \Rightarrow (\tilde{y}, \tilde{x}) \in \tilde{Y}_{W^\uparrow} \times X_m.$$

Hence, we have $\tilde{S}_{W^\uparrow} \in \mathcal{MPS}(G, R)$. □

Note that, if the system is fully observed, $\tilde{S}_{W^\uparrow}$ defined by (4.4) is reduced to the supervisor

$$S^\uparrow = (Y^\uparrow, \Sigma, \beta^\uparrow, Y_0^\uparrow, Y^\uparrow) \in \mathcal{A}(\Sigma) \tag{4.5}$$

synthesized in the same manner as in (3.10) based on the unique maximal fixpoint $W^\uparrow \in 2^{X \times Q}$ of $\tilde{F} : 2^{X \times Q} \to 2^{X \times Q}$ defined by (4.3).

Example 4.2 Recall the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$ shown in Figures 4.1 (a) and (b), respectively. The unique maximal fixpoint $W^{\uparrow}$ of the function $\tilde{F} : 2^{X \times Q} \to 2^{X \times Q}$ defined by (4.3) is obtained as

$$\begin{aligned}
\tilde{F}^1(X \times Q) &= X \times Q - \{x_2\} \times \{q_0, q_1, q_3, q_4, q_5, q_6\} - \{x_1, x_2, x_3, x_5, x_7\} \times \{q_0, q_4\} \\
&= \tilde{F}^2(X \times Q) \\
&= W^{\uparrow}.
\end{aligned}$$

The supervisor $\tilde{S}_{W^{\uparrow}} = (\tilde{Y}_{W^{\uparrow}}, \Sigma, \tilde{\beta}_{W^{\uparrow}}, \tilde{Y}_{W^{\uparrow}0}, \tilde{Y}_{W^{\uparrow}}) \in \mathcal{A}(\Sigma)$ defined by (4.4) is synthesized using $W^{\uparrow}$, whose reachable part is shown in Figure 4.2 (a), where

$$\begin{aligned}
\tilde{y}_0 &= (\{(x_0, q_0)\}, \{u\}, \{(x_0, q_0)\}), \\
\tilde{y}_1 &= (\{(x_1, q_1), (x_2, q_2)\}, \{u\}, \{(x_1, q_1), (x_2, q_2), (x_4, q_4)\}), \\
\tilde{y}_2 &= (\{(x_1, q_2), (x_2, q_2)\}, \{u\}, \{(x_1, q_2), (x_2, q_2), (x_4, q_4)\}), \\
\tilde{y}_3 &= (\{(x_3, q_3)\}, \{u\}, \{(x_3, q_3)\}), \\
\tilde{y}_4 &= (\{(x_3, q_3), (x_4, q_3)\}, \{u\}, \{(x_3, q_3), (x_4, q_3)\}), \\
\tilde{y}_5 &= (\{(x_7, q_6)\}, \{u\}, \{(x_7, q_6)\}), \\
\tilde{y}_6 &= (\{(x_3, q_5), (x_5, q_5)\}, \{u\}, \{(x_3, q_5), (x_5, q_5)\}), \\
\tilde{y}_7 &= (\{(x_6, q_6), (x_7, q_6)\}, \{u\}, \{(x_6, q_6), (x_7, q_6)\}),
\end{aligned}$$

and the self-loop by u is omitted at each state. The supervised system $\tilde{S}_{W^{\uparrow}} \| G$ is shown in Figure 4.2 (b). We have $\tilde{S}_{W^{\uparrow}} \in \mathcal{MPS}(G, R)$ by Proposition 4.6. However, since $(\tilde{y}_7, x_6)$ is an unmarked deadlock state of $\tilde{S}_{W^{\uparrow}} \| G$, $\tilde{S}_{W^{\uparrow}}$ is blocking.

4.4 Synthesis of Maximally Permissive Nonblocking Supervisor

In this section, we show that the supervisor $\tilde{S}_{W^{\uparrow}} \in \mathcal{A}(\Sigma)$ defined by (4.4) can be used as an input of the algorithm NBSR for synthesizing a maximally permissive nonblocking supervisor in $\mathcal{MPNS}(G, R)$. Moreover, we provide a necessary and sufficient condition for $\mathcal{NS}(G, R) \neq \emptyset$.

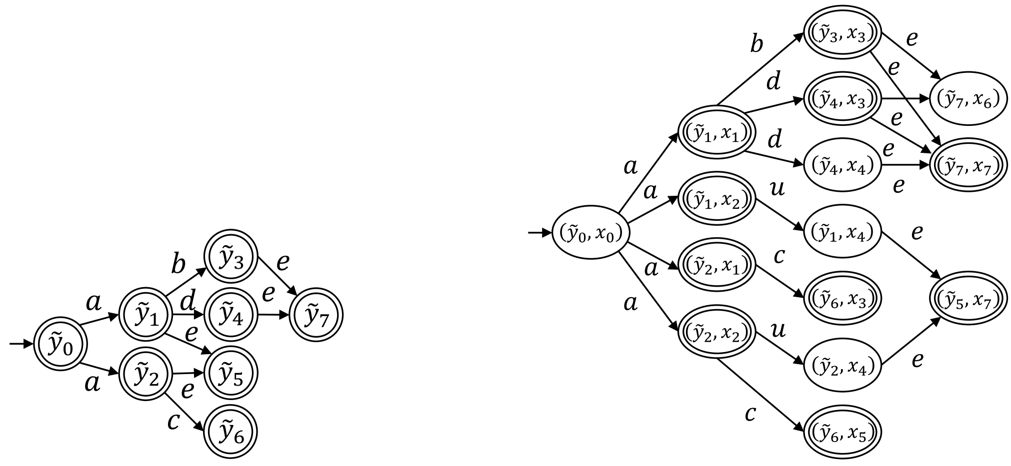


Figure 4.2: (a) Supervisor $\tilde{S}_{W^\uparrow}$ and (b) supervised system $\tilde{S}_{W^\uparrow} \| G$.

4.4.1 State-Unmergedness and Strong Maximal Permissiveness

For any $S \in \mathcal{NS}(G, R) \subseteq \mathcal{S}(G, R)$, since $\tilde{S}_{W^\uparrow} \in \mathcal{MPS}(G, R)$ as shown in Proposition 4.6, we have $S \| G \sqsubseteq \tilde{S}_{W^\uparrow} \| G$. Then, our objective is to prove that $S \| G \sqsubseteq \tilde{S}^{NB}_{W^\uparrow} \| G$ still holds for the output supervisor $\mathrm{NBSR}(\tilde{S}_{W^\uparrow}, G) = \tilde{S}^{NB}_{W^\uparrow}$. However, it is not straightforward.

$$\forall S \in \mathcal{MPS}(G, R) : \mathrm{NBSR}(S, G) \in \mathcal{A}(\Sigma) \Rightarrow \mathrm{NBSR}(S, G) \in \mathcal{MPNS}(G, R)$$

does not hold, which means that, even if an input S of NBSR is a maximally permissive similarity-enforcing supervisor, the output supervisor $\mathrm{NBSR}(S, G)$ is not necessarily a maximally permissive nonblocking one. This can be verified in the following example.

Example 4.3 We apply the algorithm NBSR to the supervisor $\tilde{S}_{W^\uparrow} \in \mathcal{A}(\Sigma)$ shown in Figure 4.2 (a), which is synthesized for the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$ shown in Figures 4.1 (a) and (b), respectively. The output supervisor $\tilde{S}^{NB}_{W^\uparrow} \in \mathcal{A}(\Sigma)$ shown in Figure 4.3 (a) is obtained after three iterations, where the states $\tilde{y}_7$ and $\tilde{y}_4$ are removed in the first and second iterations, respectively. The supervised system $\tilde{S}^{NB}_{W^\uparrow} \| G$ is shown in Figure 4.3 (b), from which we can verify that $\tilde{S}^{NB}_{W^\uparrow} \in \mathcal{NS}(G, R)$.

By comparing the supervised systems $\tilde{S}_{W^\uparrow} \| G$ and $S \| G$ shown in Figures 4.1 (d) and 4.2 (b), respectively, it holds that $S \| G \sim \tilde{S}_{W^\uparrow} \| G$, which means that the supervisor $S \in \mathcal{A}(\Sigma)$ shown in Figure 4.1 (c) is as permissive as $\tilde{S}_{W^\uparrow}$. By $S \in \mathcal{S}(G, R)$, Proposition 4.6, and the transitivity of $\sqsubseteq$, both S and $\tilde{S}_{W^\uparrow}$ are maximally permissive supervisors in $\mathcal{MPS}(G, R)$.

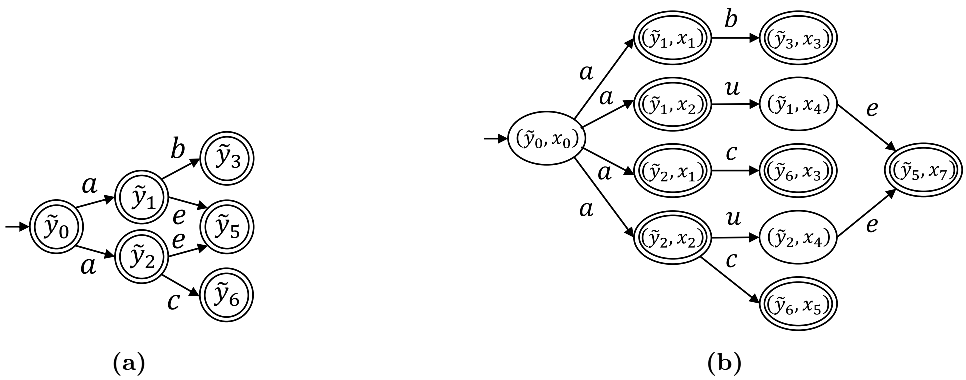


Figure 4.3: (a) Supervisor $\tilde{S}^{NB}_{W\uparrow}$ and (b) supervised system $\tilde{S}^{NB}_{W\uparrow} \| G$.

However, after applying NBSR, the event string ab is disabled by $S^{NB} = S^3 \in \mathcal{A}(\Sigma)$ as shown in Figure 4.1 (f), while it is possible in $\tilde{S}^{NB}_{W\uparrow} \| G$. Since $S^{NB} \| G \sqsubseteq \tilde{S}^{NB}_{W\uparrow} \| G$ holds but $\tilde{S}^{NB}_{W\uparrow} \| G$ is not simulated by $S^{NB} \| G$, $S^{NB} = \text{NBSR}(S, G)$ is not an element of $\mathcal{MPNS}(G, R)$.

In S, state y_3 can be reached via runs $\{u\}a\{u\}b\{u\}$ and $\{u\}a\{u\}d\{u\}$, by which the strings in $\{u\}^*\{a\}\{u\}^*\{b\}\{u\}^*$ and $\{u\}^*\{a\}\{u\}^*\{d\}\{u\}^*$ are enabled, respectively. However, for the system G, the occurrences of ab and ad lead to two different situations: (i) x_3 is the unique possible state when ab occurs; and (ii) both x_3 and x_4 are possible states when ad occurs. This means that state y_3 of S is in charge of these two different situations. As shown in Example 4.1, state y_3 is removed in the second iteration of the algorithm NBSR. Hence, the supervisor $S^{NB} = \text{NBSR}(S, G)$ disables the event strings in $\{u\}^*\{a\}\{u\}^*\{b\}\{u\}^*$ and $\{u\}^*\{a\}\{u\}^*\{d\}\{u\}^*$. On the other hand, in $\tilde{S}_{W\uparrow}$, two different states $\tilde{y}_3$ and $\tilde{y}_4$ deal with the situations (i) and (ii), respectively. As a result, the event strings in $\{u\}^*\{a\}\{u\}^*\{b\}\{u\}^*$ are enabled by $\tilde{S}^{NB}_{W\uparrow} = \text{NBSR}(\tilde{S}_{W\uparrow}, G)$.

Example 4.3 suggests that an input of the algorithm NBSR should be carefully chosen from $\mathcal{MPS}(G, R)$. Now, we show that $\tilde{S}_{W\uparrow}$ is a proper input of NBSR for computing a maximally permissive nonblocking supervisor. In order to do so, we identify two properties, which guarantee the maximal permissiveness of the output $\tilde{S}^{NB}_{W\uparrow}$.

The first property is characterized in the following definition.

Definition 4.3 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and a supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$ that satisfies the second condition of admissibility, S is said to be state-unmerged (with respect to G) if

$$\forall y \in Reach(S), \forall r_1, r_2 \in \mathcal{R}_y(S) : \bigcup_{s \in L(r_1)} \alpha(X_0, s) = \bigcup_{s \in L(r_2)} \alpha(X_0, s).$$

Each state of a state-unmerged supervisor only handles one situation of the system. The supervisor $S \in \mathcal{A}(\Sigma)$ shown in Figure 4.1 (c) is not state-unmerged, so the string ab is disabled by the state removal procedure, although we can verify in Figure 4.3 (b) that enabling the strings in $\{u\}^*\{a\}\{u\}^*\{b\}\{u\}^*$ does not cause blockingness.

Note that, for any state-unmerged supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$, we have

$$\forall y \in Reach(S) : C(y) = \bigcup_{r \in \mathcal{R}_y(S)} \left(\bigcup_{s \in L(r)} \alpha(X_0, s) \right) = \bigcup_{s' \in L(r')} \alpha(X_0, s'), \tag{4.6}$$

where r' is an arbitrary run in $\mathcal{R}_y(S)$.

To identify the second property, we consider the following example.

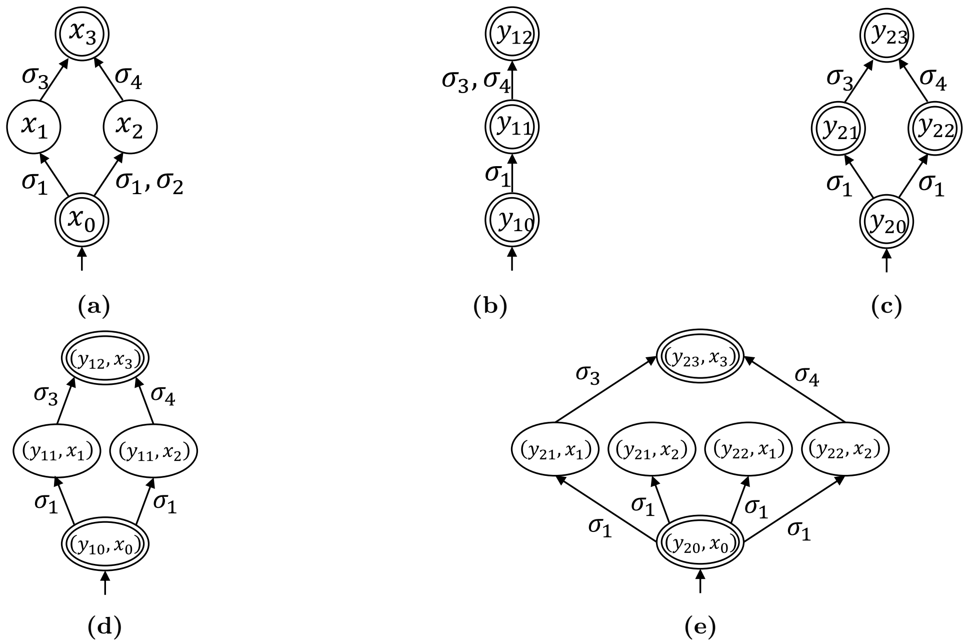


Figure 4.4: (a) System G, (b) supervisor S_1, (c) supervisor S_2, (d) supervised system $S_1 \| G$, and (e) supervised system $S_2 \| G$.

Example 4.4 We consider the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ with $\Sigma = \Sigma_c = \Sigma_o = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ shown in Figure 4.4 (a) and the specification R which only requires that the event σ_2 is disabled. Two supervisors $S_1 = (Y_1, \Sigma, \beta_1, Y_{10}, Y_1), S_2 = (Y_2, \Sigma, \beta_2, Y_{20}, Y_2) \in \mathcal{A}(\Sigma)$ for G are given in Figures 4.4 (b) and (c), respectively. The supervised systems $S_1 \| G$

and $S_2\|G$ are shown in Figures 4.4 (d) and (e), respectively. It can be easily verified that $S_1, S_2 \in \mathcal{MPS}(G, R)$. However, there does not exist a state of S_2 that enables both σ_3 and σ_4 after the occurrence of σ_1. This results in blockingness of $S_2\|G$, as shown in Figure 4.4 (e). When the algorithm NBSR is applied to S_2, states y_{21} and y_{22} are removed from S_2. Although S_2 is state-unmerged, the resulting supervisor is not an element of $\mathcal{MPNS}(G, R)$.

Example 4.4 suggests that a stronger version of a simulation relation is needed to evaluate the permissiveness of a supervisor when nonblockingness is required.

Definition 4.4 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and two supervisor $S_1 = (Y_1, \Sigma, \beta_1, Y_{10}, Y_1) \in \mathcal{A}(\Sigma)$ and $S_2 = (Y_2, \Sigma, \beta_2, Y_{20}, Y_2) \in \mathcal{A}(\Sigma)$ that satisfy the second condition of admissibility, S_2 is said to be strongly more permissive than S_1 (with respect to G) if there exists a simulation relation $\Psi \subseteq (Y_1 \times X) \times (Y_2 \times X)$ from $S_1\|G$ to $S_2\|G$ such that, for any $((y_1, x_1), (y_2, x_2)) \in \Psi$, it holds that

- $x_1 = x_2$,
- $\mathcal{R}_{y_1}(S_1) \cap \mathcal{R}_{y_2}(S_2) \neq \emptyset$, and
- $\forall x \in X, \forall s' \in L(S_1\|G) : [(y_1, x) \in \xi_1(Y_{10} \times X_0, s') \wedge (y_2, x) \in \xi_2(Y_{20} \times X_0, s')] \Rightarrow ((y_1, x), (y_2, x)) \in \Psi$,

where $\xi_1 : (Y_1 \times X) \times \Sigma \to 2^{Y_1 \times X}$ and $\xi_2 : (Y_2 \times X) \times \Sigma \to 2^{Y_2 \times X}$ are the transition functions of $S_1\|G$ and $S_2\|G$, respectively. We call such a simulation relation Ψ a strong simulation relation from $S_1\|G$ to $S_2\|G$.

Suppose that there exists a strong simulation relation Ψ from $S_1\|G$ to $S_2\|G$. For any $((y_1, x'), (y_2, x')) \in \Psi$, the second condition of a strong simulation relation requires that there exists a common run $r \in \mathcal{R}_{y_1}(S_1) \cap \mathcal{R}_{y_2}(S_2)$ that leads to y_1 and y_2 in S_1 and S_2, respectively. The third condition is imposed on y_1 and y_2 in $((y_1, x'), (y_2, x')) \in \Psi$. For any $x \in X$, if (y_1, x) and (y_2, x) can be reached by a common string $s' \in L(S_1\|G)$ as $(y_1, x) \in \xi_1(Y_{10} \times X_0, s')$ and $(y_2, x) \in \xi_2(Y_{20} \times X_0, s')$, respectively, then $((y_1, x), (y_2, x))$ has to be an element of Ψ. Since Ψ is a simulation relation from $S_1\|G$ to $S_2\|G$, for any $\sigma \in \Sigma$ with $\alpha(x, \sigma) \neq \emptyset$, if σ is enabled at y_1 by S_1, then it is also enabled at y_2 by S_2.

Remark 4.2 We can verify that, in Example 4.4, although $S_1\|G \sqsubseteq S_2\|G$ holds, there does not exist a strong simulation relation from $S_1\|G$ to $S_2\|G$. For example, a relation

$$\Phi = \{((y_{10}, x_0), (y_{20}, x_0)), ((y_{11}, x_1), (y_{21}, x_1)), ((y_{11}, x_2), (y_{22}, x_2)), ((y_{12}, x_3), (y_{23}, x_3))\}$$

is a simulation relation from $S_1\|G$ to $S_2\|G$, where each element satisfies the first and second conditions in Definition 4.4. For the pair $((y_{11}, x_1), (y_{21}, x_1)) \in \Phi$, the state $x_2 \in X$, and the string $\sigma_1 \in L(S_1\|G)$, we have $(y_{11}, x_2) \in \xi_1(Y_{10} \times X_0, \sigma_1)$ and $(y_{21}, x_2) \in \xi_2(Y_{20} \times X_0, \sigma_1)$, where $\xi_1 : (Y_1 \times X) \times \Sigma \to 2^{Y_1 \times X}$ and $\xi_2 : (Y_2 \times X) \times \Sigma \to 2^{Y_2 \times X}$ are the transition functions of $S_1\|G$ and $S_2\|G$, respectively. However, the pair $((y_{11}, x_2), (y_{21}, x_2))$ is not an element of Φ. Hence, the third condition in Definition 4.4 is not satisfied. Although all the elements of the modified relation

$$\hat{\Phi} = \Phi \cup \{((y_{11}, x_2), (y_{21}, x_2)), ((y_{11}, x_1), (y_{22}, x_1))\}$$

meet the three conditions in Definition 4.4, for the pair $((y_{11}, x_2), (y_{21}, x_2)) \in \hat{\Phi}$ and the event $\sigma_4 \in \Sigma$, we have $(y_{12}, x_3) \in \xi_1((y_{11}, x_2), \sigma_4)$ and $\xi_2((y_{21}, x_2), \sigma_4) = \emptyset$, which mean that the second condition of a simulation relation is not satisfied. Hence, $\hat{\Phi}$ is no longer a simulation relation from $S_1\|G$ to $S_2\|G$.

On the other hand, it can be verified that the reverse relation

$$\hat{\Phi}^{-1} = \{((y_2, x), (y_1, x)) \in (Y_2 \times X) \times (Y_1 \times X) | ((y_1, x), (y_2, x)) \in \hat{\Phi}\}$$

is a strong simulation relation from $S_2\|G$ to $S_1\|G$. Hence, S_1 is strongly more permissive than S_2.

Now, we show that, for any two supervisors S_1 and S_2 which satisfy the second condition of admissibility, if S_2 is state-unmerged and strongly more permissive than S_1, then these two properties are preserved under each iteration of the algorithm NBSR. We need the following results.

Lemma 4.2 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$, we consider two supervisors $S_1 = (Y_1, \Sigma, \beta_1, Y_{10}, Y_1) \in \mathcal{A}(\Sigma)$ and $S_2 = (Y_2, \Sigma, \beta_2, Y_{20}, Y_2) \in \mathcal{A}(\Sigma)$ which satisfy the second condition of admissibility. If S_2 is state-unmerged and $\Psi \subseteq (Y_1 \times X) \times (Y_2 \times X)$ is a strong simulation relation from $S_1\|G$ to $S_2\|G$, then it holds that

$$\forall((y_1,x),(y_2,x))\in\Psi: y_1\in NB_G^{S_1}(Y_1)\Rightarrow y_2\in NB_G^{S_2}(Y_2).$$

Proof Let $\xi_1:(Y_1\times X)\times\Sigma\rightarrow 2^{Y_1\times X}$ and $\xi_2:(Y_2\times X)\times\Sigma\rightarrow 2^{Y_2\times X}$ denote the transition functions of $S_1\|G$ and $S_2\|G$, respectively. For any $((y_1,x),(y_2,x))\in\Psi$ with $y_1\in NB_G^{S_1}(Y_1)$, we consider any $(y_2,x')\in \mathit{Reach}(S_2\|G)$. Since S_2 is state-unmerged, it follows from (4.6) that $x'\in\bigcup_{s\in L(r)}\alpha(X_0,s)$ for any run $r\in\mathcal{R}_{y_2}(S_2)$. By the second condition of a strong simulation relation, there exists a run r' such that $r'\in\mathcal{R}_{y_1}(S_1)\cap\mathcal{R}_{y_2}(S_2)\neq\emptyset$. Then, we have $x'\in\bigcup_{s'\in L(r')}\alpha(X_0,s')$. Hence, there exists an event string $s'\in L(r')$ such that $x'\in\alpha(X_0,s')$, $y_1\in\beta_1(Y_{10},s')$, and $y_2\in\beta_2(Y_{20},s')$. It follows that $s'\in L(S_1\|G)$, $(y_1,x')\in\beta_1(Y_{10},s')\times\alpha(X_0,s')=\xi_1(Y_{10}\times X_0,s')\subseteq \mathit{Reach}(S_1\|G)$, and $(y_2,x')\in\beta_2(Y_{20},s')\times\alpha(X_0,s')=\xi_2(Y_{20}\times X_0,s')$. By the third condition of a strong simulation relation, we have $((y_1,x'),(y_2,x'))\in\Psi$. Since $y_1\in NB_G^{S_1}(Y_1)$ and $(y_1,x')\in \mathit{Reach}(S_1\|G)$, there exists $s''\in\Sigma^*$ such that $\xi_1((y_1,x'),s'')\cap Y_1\times X_m\neq\emptyset$, that is, there exists $(y_1',x_m)\in Y_1\times X_m$ such that $(y_1',x_m)\in\xi_1((y_1,x'),s'')$. Since $((y_1,x'),(y_2,x'))\in\Psi$ and Ψ is a simulation relation from $S_1\|G$ to $S_2\|G$, by repeatedly applying the second condition of Definition 2.2, there exists $(y_2',x_m)\in\xi_2((y_2,x'),s'')$ such that $((y_1',x_m),(y_2',x_m))\in\Psi$. Since $x_m\in X_m$, we have $(y_2',x_m)\in\xi_2((y_2,x'),s'')\cap Y_2\times X_m\neq\emptyset$. Besides, since $y_1\in NB_G^{S_1}(Y_1)$, for any $\sigma\in\Sigma_{uc}$ such that $\alpha(x',\sigma)\neq\emptyset$, we have $\beta_1(y_1,\sigma)\neq\emptyset$. It follows that $\xi_1((y_1,x'),\sigma)\neq\emptyset$. Since Ψ is a strong simulation relation from $S_1\|G$ to $S_2\|G$, we have $\xi_2((y_2,x'),\sigma)\neq\emptyset$, which implies $\beta_2(y_2,\sigma)\neq\emptyset$. Thus, we have $y_2\in NB_G^{S_2}(Y_2)$. □

Lemma 4.2 just says, for two supervisors S_1 and S_2 such that S_2 is state-unmerged with respect to G and $((y_1,x),(y_2,x))$ is an element of a strong simulation relation Ψ from $S_1\|G$ to $S_2\|G$, if y_1 is not eliminated after an iteration of NBSR, neither is y_2.

Now, we have the following theorem.

Theorem 4.1 For the system $G=(X,\Sigma,\alpha,X_0,X_m)\in\mathcal{A}(\Sigma)$, we consider two supervisors $S_1=(Y_1,\Sigma,\beta_1,Y_{10},Y_1)\in\mathcal{A}(\Sigma)$ and $S_2=(Y_2,\Sigma,\beta_2,Y_{20},Y_2)\in\mathcal{A}(\Sigma)$ which satisfy the second condition of admissibility. We assume that S_2 is state-unmerged and strongly more permissive than S_1. If $Y_{10}\cap NB_G^{S_1}(Y_1)\neq\emptyset$, then the following two statements hold:

1. $Y_{20}\cap NB_G^{S_2}(Y_2)\neq\emptyset$, and

2. the subautomaton $S_2' \in \mathcal{A}(\Sigma)$ of S_2 with respect to $NB_G^{S_2}(Y_2)$ is state-unmerged and strongly more permissive than the subautomaton $S_1' \in \mathcal{A}(\Sigma)$ of S_1 with respect to $NB_G^{S_1}(Y_1)$.

Proof We write $S_1' = (Y_1', \Sigma, \beta_1', Y_{10}', Y_1') \in \mathcal{A}(\Sigma)$, where $Y_1' = NB_G^{S_1}(Y_1)$. Besides, we let $\xi_1 : (Y_1 \times X) \times \Sigma \to 2^{Y_1 \times X}$ and $\xi_2 : (Y_2 \times X) \times \Sigma \to 2^{Y_2 \times X}$ denote the transition functions of $S_1 \| G$ and $S_2 \| G$, respectively. Since S_2 is strongly more permissive than S_1, there exists a strong simulation relation $\Psi \subseteq (Y_1 \times X) \times (Y_2 \times X)$ from $S_1 \| G$ to $S_2 \| G$.

We first show that $Y_{20} \cap NB_G^{S_2}(Y_2) \neq \emptyset$. Since $Y_{10} \cap NB_G^{S_1}(Y_1) \neq \emptyset$, there exists $y_{10} \in Y_{10} \cap NB_G^{S_1}(Y_1)$. For any $x_0 \in X_0$, we have $(y_{10}, x_0) \in Y_{10} \times X_0$. Since Ψ is a strong simulation relation from $S_1 \| G$ to $S_2 \| G$, there exists $(y_{20}, x_0) \in Y_{20} \times X_0$ such that $((y_{10}, x_0), (y_{20}, x_0)) \in \Psi$. Since $y_{10} \in NB_G^{S_1}(Y_1)$, by Lemma 4.2, we have $y_{20} \in NB_G^{S_2}(Y_2)$. It follows that $y_{20} \in Y_{20} \cap NB_G^{S_1}(Y_2) \neq \emptyset$. Letting $Y_2' = NB_G^{S_2}(Y_2)$, we write the subautomaton S_2' of S_2 with respect to Y_2' as $S_2' = (Y_2', \Sigma, \beta_2', Y_{20}', Y_2') \in \mathcal{A}(\Sigma)$.

We next show that S_2' is strongly more permissive than S_1'. We prove that the relation $\Psi' \subseteq (Y_1' \times X) \times (Y_2' \times X)$ given as

$$\Psi' = \{((y_1, x), (y_2, x)) \in \Psi \mid \mathcal{R}_{y_1}(S_1') \cap \mathcal{R}_{y_2}(S_2') \neq \emptyset\}$$

is a simulation relation from $S_1' \| G$ to $S_2' \| G$, where $\xi_1' : (Y_1' \times X) \times \Sigma \to 2^{Y_1' \times X}$ and $\xi_2' : (Y_2' \times X) \times \Sigma \to 2^{Y_2' \times X}$ are the transition functions of $S_1' \| G$ and $S_2' \| G$, respectively.

- We consider any $(y_{10}, x_0) \in Y_{10}' \times X_0 \subseteq Y_{10} \times X_0$. Since Ψ is a strong simulation relation from $S_1 \| G$ to $S_2 \| G$, there exists $(y_{20}, x_0) \in Y_{20} \times X_0$ such that $((y_{10}, x_0), (y_{20}, x_0)) \in \Psi$. Since $y_{10} \in Y_{10}' = Y_{10} \cap NB_G^{S_1}(Y_1) \subseteq NB_G^{S_1}(Y_1)$, by Lemma 4.2, we have $y_{20} \in NB_G^{S_2}(Y_2) = Y_2'$. It follows that $y_{20} \in Y_{20} \cap Y_2' = Y_{20}'$. In addition, by the second condition of a strong simulation relation, we have $\mathcal{R}_{y_{10}}(S_1) \cap \mathcal{R}_{y_{20}}(S_2) \neq \emptyset$, which implies $\Sigma_{uo}(y_{10}) = \Sigma_{uo}(y_{20})$. Since $y_{10} \in Y_{10}'$ and $y_{20} \in Y_{20}'$, we have $\Sigma_{uo}(y_{10}) = \Sigma_{uo}(y_{20}) \in \mathcal{R}_{y_{10}}(S_1') \cap \mathcal{R}_{y_{20}}(S_2') \neq \emptyset$. Then, we have $((y_{10}', x_0'), (y_{20}', x_0')) \in \Psi'$.

- For any $((y_1, x), (y_2, x)) \in \Psi' \subseteq \Psi$ and any $\sigma \in \Sigma$, we consider any $(y_1', x') \in \xi_1'((y_1, x), \sigma) \subseteq \xi_1((y_1, x), \sigma)$. Since Ψ is a strong simulation relation from $S_1 \| G$ to $S_2 \| G$, there exists $(y_2', x') \in \xi_2((y_2, x), \sigma) = \beta_2(y_2, \sigma) \times \alpha(x, \sigma)$ such that $((y_1', x'), (y_2', x')) \in \Psi$. Since $y_1' \in Y_1' = NB_G^{S_1}(Y_1)$, by Lemma 4.2, we have $y_2' \in NB_G^{S_2}(Y_2) = Y_2'$. It follows that $y_2' \in \beta_2(y_2,\sigma)\cap Y_2' = \beta_2'(y_2,\sigma)$, which further implies $(y_2',x') \in \beta_2'(y_2,\sigma)\times\alpha(x,\sigma) = \xi_2'((y_2,x),\sigma)$. By the definition of Ψ', we have $\mathcal{R}_{y_1}(S_1')\cap\mathcal{R}_{y_2}(S_2') \neq \emptyset$. In the case where $\sigma \in \Sigma_{uo}$, we have $y_1' = y_1$ and $y_2' = y_2$. It follows that $\mathcal{R}_{y_1'}(S_1') \cap \mathcal{R}_{y_2'}(S_2') \neq \emptyset$, which further implies $((y_1',x'),(y_2',x')) \in \Psi'$. In the case where $\sigma \in \Sigma_o$, we consider any run $r \in \mathcal{R}_{y_1}(S_1') \cap \mathcal{R}_{y_2}(S_2') \neq \emptyset$. Since $((y_1',x'),(y_2',x')) \in \Psi$, by the second condition of a strong simulation relation, we have $\Sigma_{uo}(y_1') = \Sigma_{uo}(y_2')$. Then, it follows that $r\sigma\Sigma_{uo}(y_1') = r\sigma\Sigma_{uo}(y_2') \in \mathcal{R}_{y_1'}(S_1') \cap \mathcal{R}_{y_2'}(S_2') \neq \emptyset$, which implies $((y_1',x'),(y_2',x')) \in \Psi'$.

- For any $((y_1,x),(y_2,x)) \in \Psi'$, if $(y_1,x) \in Y_1' \times X_m$, then $(y_2,x) \in Y_2' \times X_m$.

Thus, Ψ' is a simulation relation from $S_1'\|G$ to $S_2'\|G$.

In addition, we show that Ψ' is a strong simulation relation from $S_1'\|G$ to $S_2'\|G$. By the definition of Ψ', any $((\hat{y}_1,\hat{x}),(\hat{y}_2,\hat{x})) \in \Psi' \subseteq \Psi$ satisfies the first and second conditions of a strong simulation relation. To prove that it satisfies the third condition, we consider any $\hat{x}' \in X$ and any $\hat{s} \in L(S_1'\|G) \subseteq L(S_1\|G)$ such that $(\hat{y}_1,\hat{x}') \in \xi_1'(Y_{10}' \times X_0,\hat{s})$ and $(\hat{y}_2,\hat{x}') \in \xi_2'(Y_{20}' \times X_0,\hat{s})$. We have $(\hat{y}_1,\hat{x}') \in \xi_1'(Y_{10}' \times X_0,\hat{s}) \subseteq \xi_1(Y_{10} \times X_0,\hat{s})$ and $(\hat{y}_2,\hat{x}') \in \xi_2'(Y_{20}' \times X_0,\hat{s}) \subseteq \xi_2(Y_{20} \times X_0,\hat{s})$. Since Ψ is a strong simulation relation from $S_1\|G$ to $S_2\|G$, we have $((\hat{y}_1,\hat{x}'),(\hat{y}_2,\hat{x}')) \in \Psi$. It follows that $((\hat{y}_1,\hat{x}'),(\hat{y}_2,\hat{x}')) \in \Psi'$. Thus, Ψ' is a strong simulation relation from $S_1'\|G$ to $S_2'\|G$, namely, S_2' is strongly more permissive than S_1'.

We finally show that S_2' is state-unmerged. For any $\bar{y} \in \mathit{Reach}(S_2')$, we consider any $r_1,r_2 \in \mathcal{R}_{\bar{y}}(S_2')$. By the definition of a subautomaton, we have $\bar{y} \in \mathit{Reach}(S_2') \subseteq \mathit{Reach}(S_2)$ and $r_1,r_2 \in \mathcal{R}_{\bar{y}}(S_2)$. Since S_2 is state-unmerged, we have

$$\bigcup_{s\in L(r_1)} \alpha(X_0,s) = \bigcup_{s\in L(r_2)} \alpha(X_0,s).$$

Hence, S_2' is state-unmerged. □

Recall the supervisor $\tilde{S}_{W\uparrow} = (\tilde{Y}_{W\uparrow},\Sigma,\tilde{\beta}_{W\uparrow},\tilde{Y}_{W\uparrow 0},\tilde{Y}_{W\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (4.4), which always enables all the uncontrollable and unobservable events, namely, for any $y = (W_1^y,\gamma_{uo}^y,W_2^y) \in \tilde{Y}_{W\uparrow}$, it holds that $\Sigma_{uc}\cap\Sigma_{uo} \subseteq \gamma_{uo}^y = \Sigma_{uo}(y)$. Since a strong simulation relation requires that the two supervisors being compared share the same run, hereafter, we only consider admissible and $\Sigma_{uc}\cap\Sigma_{uo}$-compatible supervisors without loss of generality. Note that it was shown in Proposition 2.3 that any admissible supervisor can be transformed into an equivalent admissible and $\Sigma_{uc}\cap\Sigma_{uo}$-compatible one. We let $\widehat{\mathcal{MPS}}(G,R) \subseteq \mathcal{MPS}(G,R)$

be the set of all state-unmerged, admissible, and $\Sigma_{uc} \cap \Sigma_{uo}$-compatible supervisors which are strongly more permissive than any $\Sigma_{uc} \cap \Sigma_{uo}$-compatible supervisor in $\mathcal{S}(G, R)$ (namely, strongly maximally permissive).

By Theorem 4.1, we can show that if the nonblocking similarity control problem is solvable, a maximally permissive nonblocking similarity-enforcing supervisor can be computed by applying the algorithm NBSR to a supervisor in $\widehat{\mathcal{MPS}}(G, R)$.

Theorem 4.2 We consider the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$. If $\mathcal{NS}(G, R) \neq \emptyset$, then, for any $\hat{S} = (\hat{Y}, \Sigma, \hat{\beta}, \hat{Y}_0, \hat{Y}) \in \widehat{\mathcal{MPS}}(G, R)$, it holds that NBSR$(\hat{S}, G) \in \mathcal{MPNS}(G, R)$.

Proof We first show NBSR$(\hat{S}, G) \in \mathcal{A}(\Sigma)$. When the algorithm NBSR is applied to $\hat{S}$, it terminates after $n \leq |\hat{Y}|$ iterations. For each $i \in \{1, 2, \ldots, n\}$, we let $\hat{S}^i$ be the result of the ith iteration. We consider any $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{NS}(G, R) \subseteq \mathcal{S}(G, R)$. By Proposition 2.3, there exists an equivalent admissible and $\Sigma_{uc} \cap \Sigma_{uo}$-compatible supervisor $S' = (Y', \Sigma, \beta', Y'_0, Y') \in \mathcal{A}(\Sigma)$ such that $S \| G = S' \| G$. It follows that $S' \in \mathcal{NS}(G, R) \subseteq \mathcal{S}(G, R)$. Since S' satisfies the first condition of admissibility and is nonblocking, we have $NB_G^S(Y') = Y'$ by Lemma 4.1. It follows that $Y'_0 \cap NB_G^S(Y') = Y'_0 \cap Y' \neq \emptyset$, and S' itself is the subautomaton of S' with respect to $NB_G^S(Y')$. Since S' and $\hat{S}$ satisfy the second condition of admissibility, and $\hat{S}$ is state-unmerged and strongly more permissive than S', by Theorem 4.1, $\hat{S}^1 \in \mathcal{A}(\Sigma)$ holds, and $\hat{S}^1$ is state-unmerged and strongly more permissive than S'. By repeating the same argument, we can conclude that, for each $i \in \{1, 2, \ldots, n\}$, $\hat{S}^i \in \mathcal{A}(\Sigma)$ holds, and $\hat{S}^i$ is state-unmerged and strongly more permissive than S'. Hence, NBSR$(\hat{S}, G) = \hat{S}^n \in \mathcal{A}(\Sigma)$ holds. In addition, we have $S \| G = S' \| G \sqsubseteq \hat{S}^i \| G$ for each $i \in \{1, 2, \ldots, n\}$. Since $\hat{S} \in \widehat{\mathcal{MPS}}(G, R) \subseteq \mathcal{MPS}(G, R) \subseteq \mathcal{S}(G, R)$, it follows from Proposition 4.3 that NBSR$(\hat{S}, G) \in \mathcal{NS}(G, R)$. Hence, we have NBSR$(\hat{S}, G) \in \mathcal{MPNS}(G, R)$. □

4.4.2 Computation of a Maximally Permissive Nonblocking Similarity-Enforcing Supervisor Using $\tilde{S}_{W\uparrow}$

In this subsection, we show that the supervisor $\tilde{S}_{W\uparrow}$ defined by (4.4) is an element of $\widehat{\mathcal{MPS}}(G, R)$. The state-unmergedness of $\tilde{S}_{W\uparrow}$ is shown in the lemma below.

Lemma 4.3 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $\tilde{S}_{W^\uparrow} = (\tilde{Y}_{W^\uparrow}, \Sigma, \tilde{\beta}_{W^\uparrow}, \tilde{Y}_{W^\uparrow 0}, \tilde{Y}_{W^\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (4.4) is state-unmerged.

Proof To show the state-unmergedness of $\tilde{S}_{W^\uparrow}$, we show that, for any $y = (W_1^y, \gamma_{uo}^y, W_2^y) \in Reach(\tilde{S}_{W^\uparrow})$ and any $r \in \mathcal{R}_y(\tilde{S}_{W^\uparrow})$, it holds that

$$\bigcup_{s \in L(r)} \alpha(X_0, s) = \pi_X(W_2^y).$$

Any $r \in \mathcal{R}_y(\tilde{S}_{W^\uparrow})$ can be written as $r = \gamma_0$ or $r = \gamma_0\sigma_0 \ldots \gamma_{n-1}\sigma_{n-1}\gamma_n$.

First, we consider the case where $r = \gamma_0$. It follows that $y \in \tilde{Y}_{W^\uparrow 0}$, $\gamma_0 = \gamma_{uo}^y$, and $L(r) = \gamma_{uo}^{y*}$. Since $\varepsilon \in \Sigma_o^* \cap L(\tilde{S}_{W^\uparrow})$ and $y \in \tilde{\beta}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0}, \varepsilon)$, we have

$$\bigcup_{s \in L(r)} \alpha(X_0, s) = \bigcup_{s \in \gamma_{uo}^{y*}} \alpha(X_0, s) = \bigcup_{s \in P^{-1}(\varepsilon) \cap L_y(\tilde{S}_{W^\uparrow})} \alpha(X_0, s) = \pi_X(W_2^y) \tag{4.7}$$

by Corollary 3.1.

Next, we consider the case where $r = \gamma_0\sigma_0 \ldots \gamma_{n-1}\sigma_{n-1}\gamma_n$. There exist $y_0, y_1, \ldots, y_n \in Reach(\tilde{S}_{W^\uparrow})$ with $y_0 \in \tilde{Y}_{W^\uparrow 0}$, $y_n = y$, and $y_{i+1} \in \beta(y_i, \sigma_i)$ for each $i \in \{0, 1, \ldots, n-1\}$, where $\sigma_i \in \Sigma_o$, $y_i = (W_1^{y_i}, \gamma_{uo}^{y_i}, W_2^{y_i})$, and $\gamma_i = \gamma_{uo}^{y_i}$ for each $i \in \{0, 1, \ldots, n\}$. By (4.7), we have

$$\bigcup_{s \in \gamma_{uo}^{y_0 *}} \alpha(X_0, s) = \pi_X(W_2^{y_0}).$$

For each $i \in \{0, 1, \ldots, n-1\}$, we consider $y_{i+1} \in \tilde{\beta}_{W^\uparrow}(y_i, \sigma_i)$. By the definition of $\tilde{\beta}_{W^\uparrow}$ and N, we have $\pi_X(W_1^{y_{i+1}}) = \alpha(\pi_X(W_2^{y_i}), \sigma_i)$. Since $W_1^{y_{i+1}} \in 2^{W^\uparrow} - \{\emptyset\}$, $\gamma_{uo}^{y_{i+1}} \in \Gamma_{uo}$, and $W_2^{y_{i+1}} \in \min U(W_1^{y_{i+1}}, \gamma_{uo}^{y_{i+1}})$ hold by the definition of $\tilde{Y}_{W^\uparrow}$, we have

$$\bigcup_{s \in \gamma_{uo}^{y_{i+1} *}} \alpha(\pi_X(W_1^{y_{i+1}}), s) = \pi_X(W_2^{y_{i+1}})$$

by (3.7) proved in Lemma 3.1. It follows that

$$\bigcup_{s \in \gamma_{uo}^{y_{i+1} *}} \alpha(\pi_X(W_2^{y_i}), \sigma_i s) = \pi_X(W_2^{y_{i+1}}).$$

Since $L(r) = \{s_0\sigma_0 \ldots s_{n-1}\sigma_{n-1}s_n \in L(\tilde{S}_{W^\uparrow}) \mid \forall i \in \{0, 1, \ldots, n\} : s_i \in \gamma_{uo}^{y_i*}\}$, we have

$$\pi_X(W_2^y) = \pi_X(W_2^{y_n}) = \bigcup_{s \in L(r)} \alpha(X_0, s).$$

Hence, for any $r_1, r_2 \in \mathcal{R}_y(\tilde{S}_{W^\uparrow})$, we have

$$\bigcup_{s \in L(r_1)} \alpha(X_0, s) = \bigcup_{s \in L(r_2)} \alpha(X_0, s) = \pi_X(W_2^y),$$

namely, $\tilde{S}_{W^\uparrow}$ is state-unmerged. $\square$

To show that $\tilde{S}_{W^\uparrow}$ is strongly more permissive than any $\Sigma_{uc} \cap \Sigma_{uo}$-compatible supervisor S in $\mathcal{S}(G, R)$, we need the following lemma.

Lemma 4.4 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, we consider any supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{S}(G, R)$, any simulation relation $\Psi \subseteq (Y \times X) \times Q$ from $S \| G$ to R, and any $y \in Y$. If a relation $\Phi_y \subseteq X \times Q$ defined by

$$\Phi_y = \{(x, q) \in X \times Q \mid ((y, x), q) \in \Psi \wedge (y, x) \in \mathit{Reach}(S \| G)\} \tag{4.8}$$

is nonempty, then, for any nonempty subset $W \subseteq \Phi_y$, it holds that $\Phi_y \in U(W, \Sigma_{uo}(y)) \neq \emptyset$.

Proof We consider any $y \in Y$ such that $\Phi_y \subseteq X \times Q$ defined by (4.8) is nonempty. It holds that $\Phi_y \subseteq \Phi$ for the relation $\Phi \subseteq X \times Q$ defined by (4.2), which is a Σ_{uc}-simulation relation from G to R as shown in the proof of necessity part of Proposition 4.4. Then, for the unique maximal fixpoint $W^\uparrow$ of the function $\tilde{F} : 2^{X \times Q} \to 2^{X \times Q}$ defined by (4.3), we have $\Phi_y \subseteq \Phi \subseteq W^\uparrow$. For any $W \subseteq \Phi_y$ with $W \neq \emptyset$, it holds that $W \in 2^{W^\uparrow}$. To show that $\Phi_y \in U(W, \Sigma_{uo}(y))$, we consider any $(x, q) \in \Phi_y$, any $\sigma \in \Sigma_{uo}(y)$, and any $x' \in \alpha(x, \sigma)$. By the definition of Φ_y, we have $((y, x), q) \in \Psi$ and $(y, x) \in \mathit{Reach}(S \| G)$. Since $\sigma \in \Sigma_{uo}(y) \subseteq \Sigma_{uo}$, it follows from the admissibility of S that $\beta(y, \sigma) = \{y\}$, which implies $(y, x') \in \beta(y, \sigma) \times \alpha(x, \sigma) = \xi((y, x), \sigma) \subseteq \mathit{Reach}(S \| G)$. Since Ψ is a simulation relation from $S \| G$ to R, there exists $q' \in \delta(q, \sigma)$ such that $(y, x'), q') \in \Psi$, which implies together with $(y, x') \in \mathit{Reach}(S \| G)$ that $(x', q') \in \Phi_y$. It follows from $W \subseteq \Phi_y$ that $\Phi_y \in U(W, \Sigma_{uo}(y)) \neq \emptyset$.
$\square$

Now, we prove in the following proposition that $\tilde{S}_{W^\uparrow} \in \widehat{\mathcal{MPS}}(G, R)$.

Proposition 4.7 For the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $\tilde{S}_{W^\uparrow} = (\tilde{Y}_{W^\uparrow}, \Sigma, \tilde{\beta}_{W^\uparrow}, \tilde{Y}_{W^\uparrow 0}, \tilde{Y}_{W^\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (4.4) satisfies $\tilde{S}_{W^\uparrow} \in \widehat{\mathcal{MPS}}(G, R)$.

Proof By proposition 4.6, we have $\tilde{S}_{W^\uparrow} \in \mathcal{MPS}(G, R) \subseteq \mathcal{S}(G, R)$. By the definition of $\tilde{S}_{W^\uparrow}$, it is $\Sigma_{uc} \cap \Sigma_{uo}$-compatible. The state-unmergedness of $\tilde{S}_{W^\uparrow}$ is given in Lemma 4.3.

We show that $\tilde{S}_{W^\uparrow}$ is strongly more permissive than any $\Sigma_{uc} \cap \Sigma_{uo}$-compatible supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{S}(G, R)$. There exists a simulation relation $\Psi \subseteq (Y \times X) \times Q$ from $S \| G$ to R. Then, by the proof of necessity part of Proposition 4.4, the relation $\Phi \subseteq X \times Q$ defined by (4.2) is a Σ_{uc}-simulation relation from G to R. For the unique maximal fixpoint $W^\uparrow$ of the function $\tilde{F} : 2^{X \times Q} \to 2^{X \times Q}$ defined by (4.3), we have $\Phi \subseteq W^\uparrow$. It follows that

$$\forall (x, q) \in X \times Q : [\exists y \in Y : ((y, x), q) \in \Psi \wedge (y, x) \in \mathit{Reach}(S \| G)] \Rightarrow (x, q) \in \Phi \subseteq W^\uparrow. \quad (4.9)$$

We let $\xi : (Y \times X) \times \Sigma \to 2^{Y \times X}$ and $\tilde{\xi}_{W^\uparrow} : (\tilde{Y}_{W^\uparrow} \times X) \times \Sigma \to 2^{\tilde{Y}_{W^\uparrow} \times X}$ denote the nondeterministic transition functions of $S \| G$ and $\tilde{S}_{W^\uparrow} \| G$, respectively. We show that the relation $\tilde{\Phi} \subseteq (Y \times X) \times (\tilde{Y}_{W^\uparrow} \times X)$ defined as

$$\begin{aligned} \tilde{\Phi} \;=\; & \{((y, x), (\tilde{y}, x)) \in (Y \times X) \times (\tilde{Y}_{W^\uparrow} \times X) \mid \mathcal{R}_y(S) \cap \mathcal{R}_{\tilde{y}}(\tilde{S}_{W^\uparrow}) \neq \emptyset \wedge \\ & [\forall (\tilde{x}, \tilde{q}) \in W_2^{\tilde{y}} : ((y, \tilde{x}), \tilde{q}) \in \Psi \wedge (y, \tilde{x}) \in \mathit{Reach}(S \| G)] \wedge \\ & [\exists s \in L(S \| G) : (y, x) \in \xi(Y_0 \times X_0, s) \wedge (\tilde{y}, x) \in \tilde{\xi}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0} \times X_0, s)]\} \end{aligned}$$

is a strong simulation relation from $S \| G$ to $\tilde{S}_{W^\uparrow} \| G$, where $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}})$.

- We consider any $(y_0, x_0) \in Y_0 \times X_0$. To prove that there exists $\tilde{y}_0 = (W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0}, W_2^{\tilde{y}_0}) \in \tilde{Y}_{W^\uparrow 0}$ such that $((y_0, x_0), (\tilde{y}_0, x_0)) \in \tilde{\Phi}$, we first prove that there exists $W_1^{\tilde{y}_0} \in 2^{W^\uparrow}$ such that

$$|W_1^{\tilde{y}_0}| = |X_0| \wedge [\forall x_0' \in X_0, \exists q_0' \in Q_0 : (x_0', q_0') \in W_1^{\tilde{y}_0}] \quad (4.10)$$

and

$$\forall (x_0', q_0') \in W_1^{\tilde{y}_0} : ((y_0, x_0'), q_0') \in \Psi \wedge (y_0, x_0') \in \mathit{Reach}(S \| G). \quad (4.11)$$

For any $x_0' \in X_0$, we have $(y_0, x_0') \in Y_0 \times X_0 \subseteq \mathit{Reach}(S \| G)$. Since Ψ is a simulation relation from $S\|G$ to R, there exists $q_0' \in Q_0$ such that $((y_0, x_0'), q_0') \in \Psi$. By (4.9), we have $(x_0', q_0') \in \Phi \subseteq W^\uparrow$. For every $x_0'' \in X_0$, we pick $q_0'' \in Q_0$ with $((y_0, x_0''), q_0'') \in \Psi$ to form a pair $(x_0'', q_0'') \in W^\uparrow$ and let $W_1^{\tilde{y}_0} \in 2^{W^\uparrow}$ be the set of these $|X_0|$ pairs. Then, $W_1^{\tilde{y}_0}$ satisfies (4.10) and (4.11). Since $X_0 \neq \emptyset$, we have $W_1^{\tilde{y}_0} \neq \emptyset$. For the relation $\Phi_{y_0} \subseteq X \times Q$ defined by (4.8), we have $W_1^{\tilde{y}_0} \subseteq \Phi_{y_0} \neq \emptyset$. Let $\gamma_{uo}^{\tilde{y}_0} = \Sigma_{uo}(y_0)$, we have $\gamma_{uo}^{\tilde{y}_0} \in \Gamma_{uo}$ since S is $\Sigma_{uc} \cap \Sigma_{uo}$-compatible. It follows from Lemma 4.4 that $\Phi_{y_0} \in U(W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0})$. Then, there exists $W_2^{\tilde{y}_0} \in \min U(W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0}) \subseteq 2^{W^\uparrow}$ such that $W_2^{\tilde{y}_0} \subseteq \Phi_{y_0}$. Letting $\tilde{y}_0 = (W_1^{\tilde{y}_0}, \gamma_{uo}^{\tilde{y}_0}, W_2^{\tilde{y}_0})$, we have $\tilde{y}_0 \in \tilde{Y}_{W^\uparrow}$. By (4.10), it holds that $\tilde{y}_0 \in \tilde{Y}_{W^\uparrow 0}$. It follows from $\Sigma_{uo}(\tilde{y}_0) = \gamma_{uo}^{\tilde{y}_0} = \Sigma_{uo}(y_0)$ that $\Sigma_{uo}(y_0) = \Sigma_{uo}(\tilde{y}_0) \in \mathcal{R}_{y_0}(S) \cap \mathcal{R}_{\tilde{y}_0}(\tilde{S}_{W^\uparrow})$. Besides, we have $\varepsilon \in L(S\|G)$, $(y_0, x_0) \in \xi(Y_0 \times X_0, \varepsilon)$, and $(\tilde{y}_0, x_0) \in \tilde{Y}_{W^\uparrow 0} \times X_0 = \tilde{\xi}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0} \times X_0, \varepsilon)$. Moreover, by $W_2^{\tilde{y}_0} \subseteq \Phi_{y_0}$, we have

$$\forall (\tilde{x}_0, \tilde{q}_0) \in W_2^{\tilde{y}_0} : ((y_0, \tilde{x}_0), \tilde{q}_0) \in \Psi \wedge (y_0, \tilde{x}_0) \in \mathit{Reach}(S\|G).$$

It follows that $((y_0, x_0), (\tilde{y}_0, x_0)) \in \tilde{\Phi}$.

- For any $((y, x), (\tilde{y}, x)) \in \tilde{\Phi}$ and any $\sigma \in \Sigma$, we consider any $(y', x') \in \xi((y, x), \sigma)$. It follows that $y' \in \beta(y, \sigma) \neq \emptyset$ and $x' \in \alpha(x, \sigma) \neq \emptyset$. Since $((y, x), (\tilde{y}, x)) \in \tilde{\Phi}$, there exists $s \in L(S\|G)$ such that $(y, x) \in \xi(Y_0 \times X_0, s)$ and $(\tilde{y}, x) \in \tilde{\xi}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0} \times X_0, s)$. It follows that $y \in \beta(Y_0, s)$, $\tilde{y} \in \tilde{\beta}_{W^\uparrow}(\tilde{Y}_{W^\uparrow 0}, s)$, and $x \in \alpha(X_0, s)$. Then, there exists $r \in \mathcal{R}_{\tilde{y}}(\tilde{S}_{W^\uparrow})$ such that $s \in L(r)$. We write $\tilde{y} = (W_1^{\tilde{y}}, \gamma_{uo}^{\tilde{y}}, W_2^{\tilde{y}})$. By Lemma 4.3, we have $x \in \pi_X(W_2^{\tilde{y}})$, which further implies
$$x' \in \alpha(x, \sigma) \subseteq \alpha(\pi_X(W_2^{\tilde{y}}), \sigma) \neq \emptyset. \tag{4.12}$$

We consider the case where $\sigma \in \Sigma_o$. To prove that there exists $(\tilde{y}', x') \in \xi_{W^\uparrow}((\tilde{y}, x), \sigma)$ such that $((y', x'), (\tilde{y}', x')) \in \tilde{\Phi}$, we first show $\sigma \in \Sigma_{W^\uparrow}(\tilde{y})$. For any $(\tilde{x}, \tilde{q}) \in W_2^{\tilde{y}}$, we consider any $\tilde{x}' \in \alpha(\tilde{x}, \sigma)$. By the definition of $\tilde{\Phi}$, we have $((y, \tilde{x}), \tilde{q}) \in \Psi$ and $(y, \tilde{x}) \in \mathit{Reach}(S\|G)$. Then, it holds that $(y', \tilde{x}') \in \beta(y, \sigma) \times \alpha(\tilde{x}, \sigma) = \xi((y, \tilde{x}), \sigma) \subseteq \mathit{Reach}(S\|G)$. Since Ψ is a simulation relation from $S\|G$ to R, there exists $\tilde{q}' \in \delta(\tilde{q}, \sigma)$ such that $((y', \tilde{x}'), \tilde{q}') \in \Psi$. It follows from (4.9) that $(\tilde{x}', \tilde{q}') \in \Phi \subseteq W^\uparrow$. By (4.12), we have $\sigma \in \Sigma_{W^\uparrow}(\tilde{y})$. Moreover, there exists a minimal element $W_1 \in 2^\Phi \subseteq 2^{W^\uparrow}$ of

$N(W_2^{\tilde{y}}, \sigma)$, that is, $W_1 \in \min N(W_2^{\tilde{y}}, \sigma)$ such that

$$\forall (\tilde{x}, \tilde{q}) \in W_2^{\tilde{y}}, \forall \tilde{x}' \in \alpha(\tilde{x}, \sigma), \exists \tilde{q}' \in \delta(\tilde{q}, \sigma) : (\tilde{x}', \tilde{q}') \in W_1,$$

and

$$\forall (\tilde{x}', \tilde{q}') \in W_1 : ((y', \tilde{x}'), \tilde{q}') \in \Psi \wedge (y', \tilde{x}') \in \mathit{Reach}(S \| G). \tag{4.13}$$

By (4.12), $W_1 \neq \emptyset$ holds. For the relation $\Phi_{y'} \subseteq X \times Q$ defined by (4.8), it follows from (4.13) that $W_1 \subseteq \Phi_{y'} \neq \emptyset$. Let $\gamma_{uo} = \Sigma_{uo}(y')$, we have $\gamma_{uo} \in \Gamma_{uo}$ since S is $\Sigma_{uc} \cap \Sigma_{uo}$-compatible. By Lemma 4.4, we have $\Phi_{y'} \in U(W_1, \gamma_{uo})$. Then, there exists $W_2 \in \min U(W_1, \gamma_{uo})$ such that $W_2 \subseteq \Phi_{y'}$. Letting $\tilde{y}' = (W_1, \gamma_{uo}, W_2)$, we have $\tilde{y}' \in \tilde{Y}_{W\uparrow}$. Since $W_1 \in \min N(W_2^{\tilde{y}}, \sigma)$, by the definition of $\tilde{\beta}_{W\uparrow}$, we have $\tilde{y}' \in \tilde{\beta}_{W\uparrow}(\tilde{y}, \sigma)$. Then, it holds that $(\tilde{y}', x') \in \tilde{\beta}_{W\uparrow}(\tilde{y}, \sigma) \times \alpha(x, \sigma) = \tilde{\xi}_{W\uparrow}((\tilde{y}, x), \sigma)$. Since $(y', x') \in \xi((y, x), \sigma)$, we have $s\sigma \in L(S\|G)$, $(y', x') \in \xi(Y_0 \times X_0, s\sigma)$, and $(\tilde{y}', x') \in \tilde{\xi}_{W\uparrow}(\tilde{Y}_{W\uparrow 0} \times X_0, s\sigma)$. Besides, by the definition of $\tilde{\Phi}$, there exists $r \in \mathcal{R}_y(S) \cap \mathcal{R}_{\tilde{y}}(\tilde{S}_{W\uparrow})$. It follows from $\Sigma_{uo}(y') = \gamma_{uo} = \Sigma_{uo}(\tilde{y}')$ that $r\sigma\Sigma_{uo}(y') = r\sigma\Sigma_{uo}(\tilde{y}') \in \mathcal{R}_{y'}(S) \cap \mathcal{R}_{\tilde{y}'}(\tilde{S}_{W\uparrow}) \neq \emptyset$. Moreover, by $W_2 \subseteq \Phi_{y'}$, it holds that

$$\forall (\tilde{x}'', \tilde{q}'') \in W_2 : ((y', \tilde{x}''), \tilde{q}'') \in \Psi \wedge (y', \tilde{x}'') \in \mathit{Reach}(S \| G).$$

It follows that $((y', x'), (\tilde{y}', x')) \in \tilde{\Phi}$.

We next consider the case where $\sigma \in \Sigma_{uo}$. It follows from $\beta(y, \sigma) \neq \emptyset$ that $\sigma \in \Sigma_{uo}(y)$. Since $\mathcal{R}_y(S) \cap \mathcal{R}_{\tilde{y}}(\tilde{S}_{W\uparrow}) \neq \emptyset$, we have $\Sigma_{uo}(y) = \Sigma_{uo}(\tilde{y}) = \gamma_{uo}^{\tilde{y}}$, which implies $\sigma \in \gamma_{uo}^{\tilde{y}}$. By the definition of $\tilde{\beta}_{W\uparrow}$, we have $\{\tilde{y}\} = \tilde{\beta}_{W\uparrow}(\tilde{y}, \sigma)$, which further implies $(\tilde{y}, x') \in \tilde{\beta}_{W\uparrow}(\tilde{y}, \sigma) \times \alpha(x, \sigma) = \tilde{\xi}_{W\uparrow}((\tilde{y}, x), \sigma)$. Since S is admissible, we have $\{y\} = \beta(y, \sigma)$, that is, $y = y'$. Hence, we have $s\sigma \in L(S\|G)$, $(y, x') \in \xi(Y_0 \times X_0, s\sigma)$, and $(\tilde{y}, x') \in \tilde{\xi}_{W\uparrow}(\tilde{Y}_{W\uparrow 0} \times X_0, s\sigma)$. Since $((y, x), (\tilde{y}, x)) \in \tilde{\Phi}$, we have $\mathcal{R}_y(S) \cap \mathcal{R}_{\tilde{y}}(\tilde{S}_{W\uparrow}) \neq \emptyset$ and

$$\forall (\tilde{x}, \tilde{q}) \in W_2^{\tilde{y}} : ((y, \tilde{x}), \tilde{q}) \in \Psi \wedge (y, \tilde{x}) \in \mathit{Reach}(S \| G).$$

It follows that $((y, x'), (\tilde{y}, x')) \in \tilde{\Phi}$.

- For any $((y, x), (\tilde{y}, x)) \in \tilde{\Phi}$, if $(y, x) \in Y \times X_m$, then $(\tilde{y}, x) \in \tilde{Y}_{W\uparrow} \times X_m$.

Thus, $\tilde{\Phi}$ is a simulation relation from $S\|G$ to $\tilde{S}_{W\uparrow}\|G$. By the definition of $\tilde{\Phi}$, any $((y,x),(\tilde{y},x)) \in \tilde{\Phi}$ satisfies the first and second conditions of a strong simulation relation. To prove the third condition, we consider any $\hat{x} \in X$ and any $\hat{s} \in L(S\|G)$ such that $(y,\hat{x}) \in \xi(Y_0 \times X_0, \hat{s})$ and $(\tilde{y},\hat{x}) \in \tilde{\xi}_{W\uparrow}(\tilde{Y}_{W\uparrow 0} \times X_0, \hat{s})$. Since $((y,x),(\tilde{y},x)) \in \tilde{\Phi}$, we have $\mathcal{R}_y(S) \cap \mathcal{R}_{\tilde{y}}(\tilde{S}_{W\uparrow}) \neq \emptyset$ and

$$\forall(\tilde{x},\tilde{q}) \in W_2^{\tilde{y}} : ((y,\tilde{x}),\tilde{q}) \in \Psi \wedge (y,\tilde{x}) \in \mathit{Reach}(S\|G).$$

It follows that $((y,\hat{x}),(\tilde{y},\hat{x})) \in \tilde{\Phi}$. Then, $\tilde{\Phi}$ is a strong simulation relation from $S\|G$ to $\tilde{S}_{W\uparrow}\|G$. Thus, we have $\tilde{S}_{W\uparrow} \in \widehat{\mathcal{MPS}}(G,R)$. $\square$

Note that, since $\tilde{S}_{W\uparrow} \in \widehat{\mathcal{MPS}}(G,R)$ can be synthesized if G is Σ_{uc}-simulated by R, by Propositions 4.4 and 4.7, we have

$$\mathcal{S}(G,R) \neq \emptyset \Leftrightarrow \widehat{\mathcal{MPS}}(G,R) \neq \emptyset. \tag{4.14}$$

By Theorem 4.2, Proposition 4.7, and the fact that $\mathcal{NS}(G,R) \subseteq \mathcal{S}(G,R)$, NBSR$(\tilde{S}_{W\uparrow},G)$ is a maximal solution to the nonblocking similarity control problem when it is solvable.

Corollary 4.1 We consider the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$. If $\mathcal{NS}(G,R) \neq \emptyset$, then G is Σ_{uc}-simulated by R, and it holds that NBSR$(\tilde{S}_{W\uparrow},G) = \tilde{S}_{W\uparrow}^{NB} \in \mathcal{MPNS}(G,R)$ for the supervisor $\tilde{S}_{W\uparrow} \in \mathcal{A}(\Sigma)$ defined by (4.4).

Proposition 4.8 For the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the following two statements are equivalent:

1. There exists $S \in \widehat{\mathcal{MPS}}(G,R)$ such that NBSR$(S,G) \in \mathcal{A}(\Sigma)$.

2. For any $S' \in \widehat{\mathcal{MPS}}(G,R)$, NBSR$(S',G) \in \mathcal{A}(\Sigma)$.

Proof Since G is Σ_{uc}-simulated by R, it follows from Proposition 4.4 and (4.14) that $\widehat{\mathcal{MPS}}(G,R) \neq \emptyset$. It is sufficient to show 1) $\Rightarrow$ 2). For any $S \in \widehat{\mathcal{MPS}}(G,R)$ such that NBSR$(S,G) \in \mathcal{A}(\Sigma)$, since $S \in \widehat{\mathcal{MPS}}(G,R) \subseteq \mathcal{S}(G,R)$, it follows from Proposition 4.3 that NBSR$(S,G) \in \mathcal{NS}(G,R)$. Then, for any supervisor $S' \in \widehat{\mathcal{MPS}}(G,R)$, by Theorem 4.2, it holds that NBSR$(S',G) \in \mathcal{A}(\Sigma)$.

Finally, we show that the solvability of the nonblocking similarity control problem can be checked by applying the algorithm NBSR to $\tilde{S}_{W\uparrow}$.

Theorem 4.3 For the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$, $\mathcal{NS}(G, R) \neq \emptyset$ if and only if G is Σ_{uc}-simulated by R, and there exists $\hat{S} \in \widehat{\mathcal{MPS}}(G, R)$ such that $\mathrm{NBSR}(\hat{S}, G) \in \mathcal{A}(\Sigma)$.

Proof ($\Leftarrow$) We consider any $\hat{S} \in \widehat{\mathcal{MPS}}(G, R)$ such that $\mathrm{NBSR}(\hat{S}, G) \in \mathcal{A}(\Sigma)$. Since $\hat{S} \in \widehat{\mathcal{MPS}}(G, R) \subseteq \mathcal{S}(G, R)$, $\hat{S}$ satisfies the second condition of admissibility and $\hat{S} \| G \sqsubseteq R$ holds. By Proposition 4.3, we have $\mathrm{NBSR}(\hat{S}, G) \in \mathcal{NS}(G, R) \neq \emptyset$.

($\Rightarrow$) Since $\emptyset \neq \mathcal{NS}(G, R) \subseteq \mathcal{S}(G, R)$, by Proposition 4.4, G is Σ_{uc}-simulated by R. Moreover, by (4.14), there exists $\hat{S} \in \widehat{\mathcal{MPS}}(G, R) \neq \emptyset$. Then, by Theorem 4.2, $\mathrm{NBSR}(\hat{S}, G) \in \mathcal{MPNS}(G, R) \subseteq \mathcal{A}(\Sigma)$. $\square$

By Theorem 4.3 and Propositions 4.7 and 4.8, we have the following corollary.

Corollary 4.2 For the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$, $\mathcal{NS}(G, R) \neq \emptyset$ if and only if G is Σ_{uc}-simulated by R, and $\mathrm{NBSR}(\tilde{S}_{W^\uparrow}, G) = \tilde{S}_{W^\uparrow}^{NB} \in \mathcal{A}(\Sigma)$ holds for the supervisor $\tilde{S}_{W^\uparrow} \in \mathcal{A}(\Sigma)$ defined by (4.4).

By Corollaries 4.1 and 4.2, if the nonblocking similarity control problem is solvable for $G \in \mathcal{A}(\Sigma)$ and $R \in \mathcal{A}(\Sigma)$, then $\mathrm{NBSR}(\tilde{S}_{W^\uparrow}, G) = \tilde{S}_{W^\uparrow}^{NB} \in \mathcal{A}(\Sigma)$ holds, and $\tilde{S}_{W^\uparrow}^{NB} \| G$ is a supremum of $\mathcal{R}_{\sqsubseteq}^{NB}$.

Remark 4.3 By Corollary 4.2, whether or not the nonblocking similarity control problem is solvable can be verified using $\tilde{S}_{W^\uparrow}$. If G is Σ_{uc}-simulated by R and $\mathrm{NBSR}(\tilde{S}_{W^\uparrow}, G) = \tilde{S}_{W^\uparrow}^{NB} \in \mathcal{A}(\Sigma)$ holds, then we have $\mathcal{NS}(G, R) \neq \emptyset$, and Corollary 4.1 guarantees the maximal permissiveness of the output supervisor $\tilde{S}_{W^\uparrow}^{NB}$; otherwise, it holds that $\mathcal{NS}(G, R) = \emptyset$. Since the number $|\tilde{Y}_{W^\uparrow}|$ of states of $\tilde{S}_{W^\uparrow}$ is at most $2^{2|X||Q|+|\Sigma_c \cap \Sigma_{uo}|}$, the computational complexity of computing $\mathrm{NBSR}(\tilde{S}_{W^\uparrow}, G)$ is $O(|\Sigma||X|^3 2^{8|X||Q|+4|\Sigma_c \cap \Sigma_{uo}|})$.

4.4.3 Specialization to Fully Observable Case

The results presented in Subsections 4.4.1 and 4.4.2 can be directly applied to the nonblocking similarity control problem for fully observed systems by letting $\Sigma_{uo} = \emptyset$.

Note that, under full observation, the Definitions 4.3 and 4.4 of state-unmergedness and strong simulation relation, respectively, are written as follows.

Definition 4.5 For the fully observed system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and a supervisor $S = (Y, \Sigma, \beta, Y_0, Y) \in \mathcal{A}(\Sigma)$, S is said to be state-unmerged (with respect to G) if

$$\forall y \in Reach(S), \forall s_1, s_2 \in L_y(S) : \alpha(X_0, s_1) = \alpha(X_0, s_2).$$

Definition 4.6 For the fully observed system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and two supervisor $S_1 = (Y_1, \Sigma, \beta_1, Y_{10}, Y_1) \in \mathcal{A}(\Sigma)$ and $S_2 = (Y_2, \Sigma, \beta_2, Y_{20}, Y_2) \in \mathcal{A}(\Sigma)$, S_2 is said to be strongly more permissive than S_1 (with respect to G) if there exists a simulation relation $\Psi \subseteq (Y_1 \times X) \times (Y_2 \times X)$ from $S_1 \| G$ to $S_2 \| G$ such that, for any $((y_1, x_1), (y_2, x_2)) \in \Psi$, it holds that

- $x_1 = x_2$,
- $L_{y_1}(S_1) \cap L_{y_2}(S_2) \neq \emptyset$, and
- $\forall x \in X, \forall s' \in L(S_1 \| G) : [(y_1, x) \in \xi_1(Y_{10} \times X_0, s') \wedge (y_2, x) \in \xi_2(Y_{20} \times X_0, s')] \Rightarrow ((y_1, x), (y_2, x)) \in \Psi$,

where $\xi_1 : (Y_1 \times X) \times \Sigma \to 2^{Y_1 \times X}$ and $\xi_2 : (Y_2 \times X) \times \Sigma \to 2^{Y_2 \times X}$ are the transition functions of $S_1 \| G$ and $S_2 \| G$, respectively.

Recall that $\tilde{S}_{W^\uparrow} = (\tilde{Y}_{W^\uparrow}, \Sigma, \tilde{\beta}_{W^\uparrow}, \tilde{Y}_{W^\uparrow 0}, \tilde{Y}_{W^\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (4.4) is reduced to $S^\uparrow = (Y^\uparrow, \Sigma, \beta^\uparrow, Y_0^\uparrow, Y^\uparrow) \in \mathcal{A}(\Sigma)$ defined by (4.5) in the fully observed case. Hence, we have the following results.

Corollary 4.3 For the system $G \in \mathcal{A}(\Sigma)$ and the specification $R \in \mathcal{A}(\Sigma)$, $\mathcal{NS}^{fo}(G, R) \neq \emptyset$ if and only if G is Σ_{uc}-simulated by R, and $\mathrm{NBSR}(S^\uparrow, G) = S^{\uparrow NB} \in \mathcal{A}(\Sigma)$ holds for the supervisor $S^\uparrow \in \mathcal{A}(\Sigma)$ defined by (4.5). If $S^{\uparrow NB} \in \mathcal{A}(\Sigma)$, then it holds that $S^{\uparrow NB} \in \mathcal{MPNS}^{fo}(G, R)$.

Remark 4.4 Since the number $|Y^\uparrow|$ of $S^\uparrow$ is bounded by $2^{|X||Q|}$, $\mathrm{NBSR}(S^\uparrow, G)$ can be computed in $O(|\Sigma||X|^3 2^{4|X||Q|})$.

4.5 Illustrative example

In this section, we provide a practical example of the nonblocking similarity control problem, which is solved by the algorithm NBSR and the supervisor $\tilde{S}_{W^\uparrow} \in \mathcal{A}(\Sigma)$ defined by (4.4).

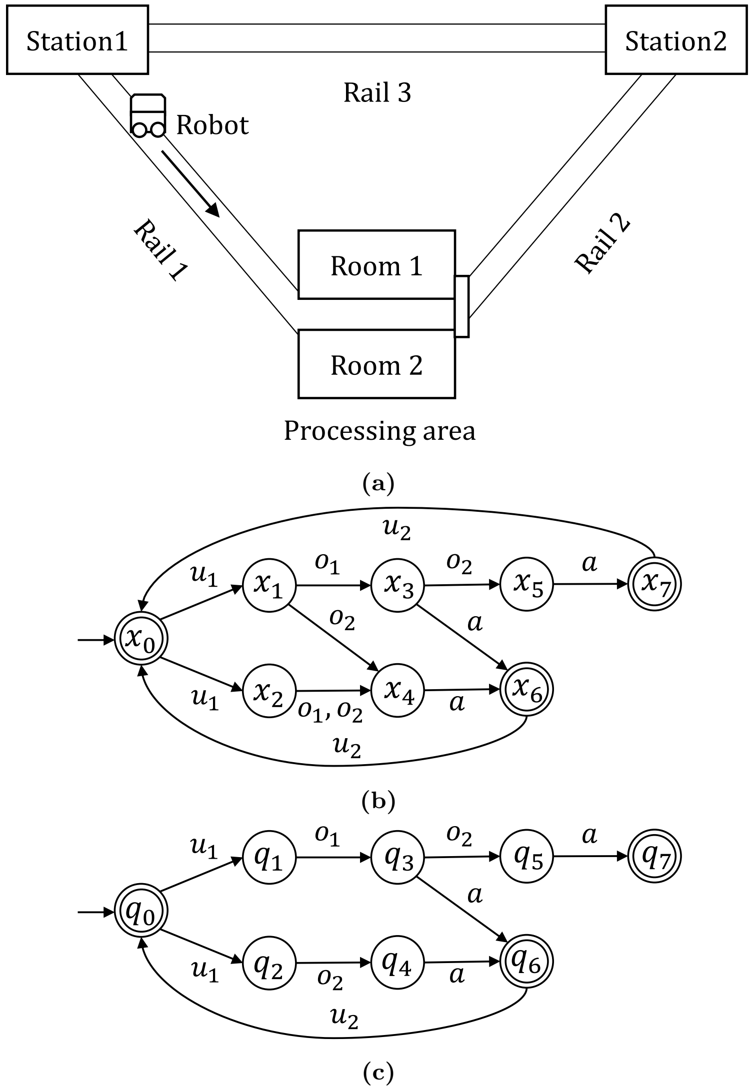


Figure 4.5: (a) A manufacturing system, (b) system model G, and (c) specification R.

For a robot working in a manufacturing system shown in Figure 4.5 (a), which consists of two stations, two processing rooms, and three rails, we consider the nondeterministic model $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ shown in Figure 4.5 (b), where $\Sigma = \Sigma_o = \{o_1, o_2, a, u_1, u_2\}$. There are two kinds of parts A and B in station 1. Initially, the robot is in station 1, and moves to processing area through rail 1 after picking a part (event u_1), where the picked part cannot be identified. The transitions $x_0 \xrightarrow{u_1} x_1$ and $x_0 \xrightarrow{u_1} x_2$ denote that the robot moves to processing area with parts A and B, respectively. In processing area, the robot performs two kinds of operations o_1 and o_2 in rooms 1 and 2, respectively. After performing o_1 or o_2 (states x_3 and x_4), the robot can directly move to station 2 through rail 2 to unload the processed part (event a). Then the robot goes back to station 1 via rail 3 (event u_2). Besides, if o_1 is performed on part A (state x_3), the robot can further process the part by o_2. Then it unloads the processed part in station 2 and moves back to station 1. The events o_1, o_2, and a are controlled by opening or closing the entrances of room 1, room 2, and rail 2, respectively. The robot can freely travel back and forth in rails 1 and 3, that is, $\Sigma_{uc} = \{u_1, u_2\}$. G is marked when the robot is in a station (states x_0, x_6, and x_7).

As the specification, we consider the nondeterministic automaton $R = (Q, \Sigma, \delta, Q_0, Q_m) \in \mathcal{A}(\Sigma)$ shown in Figure 4.5 (c). Suppose that some preparations are needed for supporting the operation of the robot, but the preparations for o_1 and o_2 cannot be done simultaneously. Then, after the occurrence of u_1, regardless of the kind of the picked part, the choice between the operations o_1 and o_2 is prohibited. Besides, if part A is processed by both the two operations o_1 and o_2, then the robot has to stay in station 2 after unloading. This control problem cannot be solved by the conventional language-based supervisory control theory. Besides, the problem of enforcing bisimilarity is not solvable for G and R.

The unique maximal fixpoint $W^{\uparrow}$ of the function $\tilde{F} : 2^{X\times Q} \to 2^{X\times Q}$ defined by (4.3) is obtained as

$$\begin{aligned}\tilde{F}^1(X\times Q) &= X\times Q - \{x_0\}\times\{q_1,q_2,q_3,q_4,q_5,q_6,q_7\} - \{x_6,x_7\}\times\{q_0,q_1,q_2,q_3,q_4,q_5,q_7\}\\ &= \tilde{F}^2(X\times Q)\\ &= W^{\uparrow}.\end{aligned}$$

The supervisor $S^{\uparrow} = (Y^{\uparrow}, \Sigma, \beta^{\uparrow}, Y_0^{\uparrow}, Y^{\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (4.5) is synthesized using $W^{\uparrow}$, whose reachable part is shown in Figure 4.6 (a), where

$$\begin{aligned}&\tilde{y}_0 = \{(x_0,q_0)\}, && \tilde{y}_1 = \{(x_1,q_1),(x_2,q_1)\},\\ &\tilde{y}_2 = \{(x_1,q_1),(x_2,q_2)\}, && \tilde{y}_3 = \{(x_1,q_2),(x_2,q_1)\},\\ &\tilde{y}_4 = \{(x_1,q_2),(x_2,q_2)\}, && \tilde{y}_5 = \{(x_3,q_3),(x_4,q_3)\},\\ &\tilde{y}_6 = \{(x_4,q_4)\}, && \tilde{y}_7 = \{(x_5,q_5)\},\\ &\tilde{y}_8 = \{(x_6,q_6)\}.\end{aligned}$$

Since $S^{\uparrow}\|G$ is blocking, as shown in Figure 4.6 (b), we apply the algorithm NBSR. After two iterations, NBSR outputs the supervisor $S^{\uparrow NB}$ shown in Figure 4.6 (c). Then, by Corollary 4.3, the nonblocking similarity control problem is solvable. The supervised system $S^{\uparrow NB}\|G$ is shown in Figure 4.6 (d). It can be verified that $S^{\uparrow NB}\|G \sqsubseteq R$ holds, and $S^{\uparrow NB}$

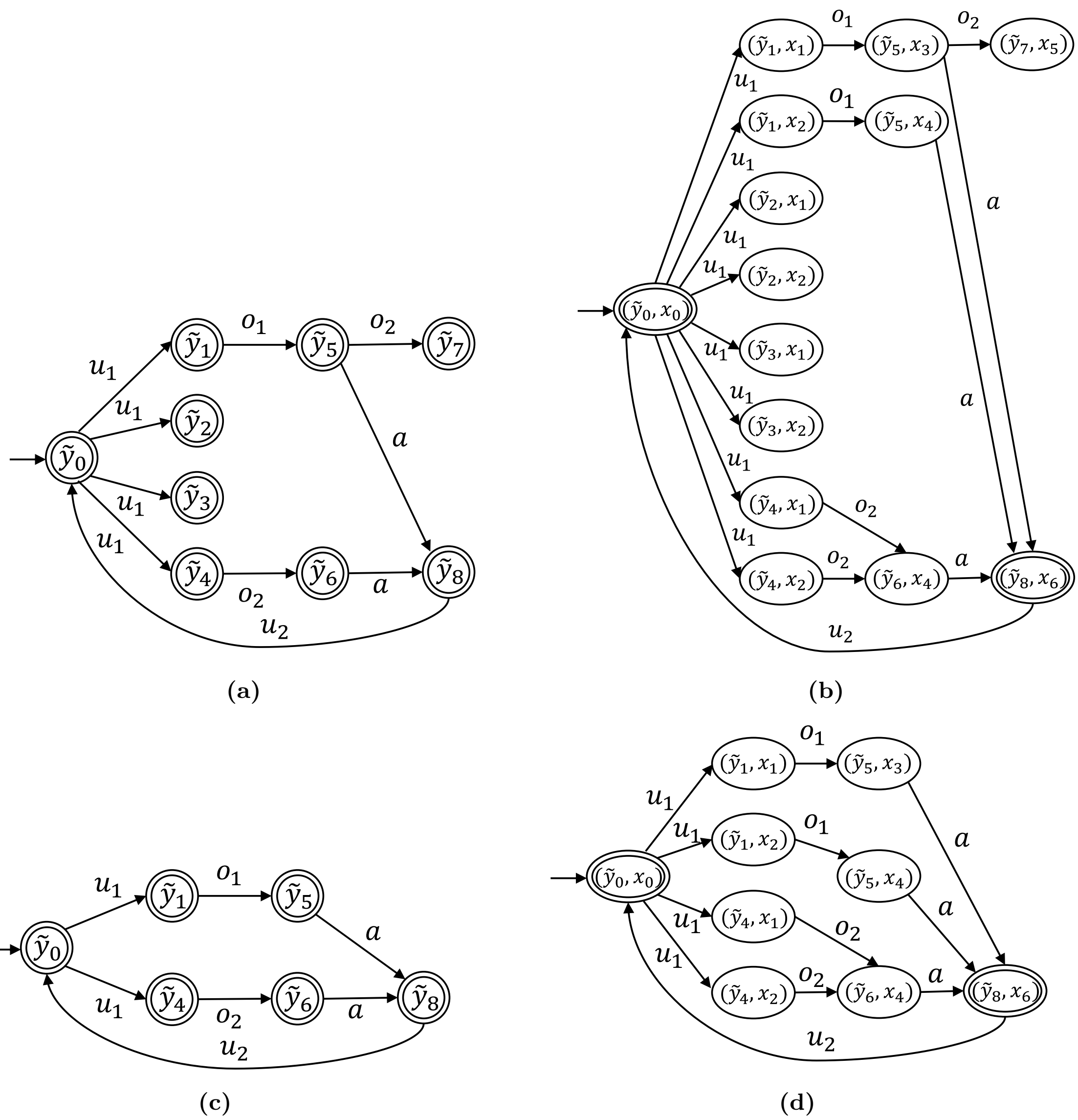


Figure 4.6: (a) Supervisor $S^{\uparrow}$, (b) supervised system $S^{\uparrow} \| G$, (c) supervisor $S^{\uparrow NB}$, and (d) supervised system $S^{\uparrow NB} \| G$.

is admissible and nonblocking. We can confirm that the requirement "o_1 and o_2 are not selectable after u_1" is satisfied by $S^{\uparrow NB}$. In addition, the maximal permissiveness of $S^{\uparrow NB}$ is guaranteed by Corollary 4.3.

4.6 Conclusion

In this chapter, we considered the nonblocking similarity control problem for the system and the specification modeled as nondeterministic automata subject to full as well as partial observation. We proposed an algorithm, named NBSR, for constructing a nonblocking supervisor from a possibly blocking one. Two properties named state-unmergedness and strong maximal permissiveness were identified, which are the key properties for synthesizing a maximally permissive nonblocking supervisor using NBSR. Using a supervisor with these two properties as the input, which is blocking in general, we showed how to check the solvability of the nonblocking similarity control problem and computed a maximal solution when it is solvable. Such a maximal solution forms a supremal feasible nonblocking subspecification of the given specification.

Chapter 5

Modular Similarity Control of Monolithic Systems with Modular Specifications

In this chapter, we consider the similarity control problem where the specification consists of a certain set of elementary specifications, which are coupled by synchronous composition. We call such a specification a modular specification. This control problem can be solved in the monolithic (centralized) way, where a monolithic supervisor is synthesized based on the modular specification. In general, the monolithic synthesis is inefficient due to the notorious exponential state space explosion problem.

We consider the modular similarity control problem, in which the similarity control problem is required to be solved in a modular way without constructing the modular specification. The modular control architecture introduced in [33] for the control of deterministic systems is applied, where, for each elementary specification, an elementary supervisor that enforces similarity is synthesized. We show that the similarity control problem with the modular specification is modularly solvable if it is monolithically solvable. Besides, we show that the elementary supervisors together achieve maximal permissiveness with respect to the modular specification if each of them is maximally permissive with respect to the corresponding elementary specification.

5.1 Problem Formulation

We consider the similarity control problem with a modular specification, where for the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$, the modular specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ is given as the synchronous composition $R = \|_{i \in I} R_i$ of n elementary specifications $R_i = (Q_i, \Sigma, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma)$ with $i \in I = \{1, \ldots, n\}$. Here, the marked state sets are omitted since we do not tackle the issue of nonblockingness in this chapter, but some discussions about nonblocking modular similarity control are given in the end of this chapter. For simplicity, we also assume that the system is fully observed. As discussed later, the results of this chapter can be easily extended to the case of partial observation.

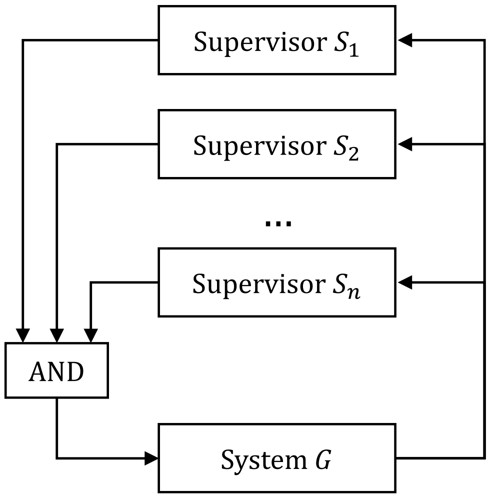


Figure 5.1: Modular control architecture.

Based on G and $R = \|_{i \in I} R_i$, the monolithic supervisor $S^{\uparrow} \in \mathcal{A}(\Sigma)$ defined by (3.10), if exists, can be synthesized as a maximal solution in $\mathcal{MPS}^{fo}(G, R)$ to the similarity control problem. However, R has at most $\prod_{i \in I} |Q_i|$ (reachable) states by the definition of the synchronous composition, which increases exponentially in the number $n = |I|$ of elementary specifications. To avoid directly handling R, we consider the modular control architecture shown in Figure 5.1. Instead of computing $R = \|_{i \in I} R_i$ and then synthesizing a maximally permissive monolithic supervisor in $\mathcal{MPS}^{fo}(G, R)$ for G and R, we individually synthesize n elementary supervisors $S_i = (Y_i, \Sigma, \beta_i, Y_{i,0}) \in \mathcal{A}(\Sigma)$ $(i \in I)$ for G and each of the elementary specifications R_i $(i \in I)$, respectively. Here, elementary supervisors $S_1, \ldots, S_n$ work in parallel, so the supervised system is modeled as $S_1 \| \ldots \| S_n \| G$. Recall that the maximal permissiveness is an important property that we pursue in the supervisor synthesis. Formally, in this chapter, we synthesize a set of Σ_{uc}-admissible elementary supervisors $S_i \in \mathcal{A}(\Sigma)$ $(i \in I)$ which satisfy $S_1 \| \ldots \| S_n \| G \sqsubseteq R = \|_{i \in I} R_i$ and

$$\forall S \in \mathcal{S}^{fo}(G, R) : S \| G \sqsubseteq S_1 \| \ldots \| S_n \| G. \tag{5.1}$$

Here, all the elementary supervisors are assumed to have the same capability on control, which means that the controllability status of each event in Σ is same for different elementary supervisors. In other words, a controllable event $\sigma \in \Sigma_c$ can be disabled by any elementary supervisor.

Remark 5.1 The control mechanism of n elementary supervisors S_i $(i \in I)$ for the (monolithic) system G can be explained as follows. Let $(y_1, \ldots, y_n, x) \in \mathit{Reach}(S_1 \| \ldots \| S_n \| G)$ be the current state of $S_1 \| \ldots \| S_n \| G$. An event $\sigma \in \Sigma$ is possible in $S_1 \| \ldots \| S_n \| G$ if and only if it is possible at x in G, namely, $\alpha(x, \sigma) \neq \emptyset$, and is enabled by all S_i $(i \in I)$, namely

$$\forall i \in I : \beta_i(y_i, \sigma) \neq \emptyset.$$

The eventual enabled event set $\Sigma((y_1, \ldots, y_n))$ is obtained by the intersection

$$\Sigma((y_1, \ldots, y_n)) = \bigcap_{i \in I} \Sigma_i(y_i)$$

of $\Sigma_i(y_i) = \{\sigma \in \Sigma \mid \beta_i(y_i, \sigma) \neq \emptyset\}$.

5.2 Applicability of Modular Control Architecture

In this section, we consider the existence condition for a set of Σ_{uc}-admissible elementary supervisors $S_i \in \mathcal{A}(\Sigma)$ $(i \in I)$ which satisfy $S_1 \| \ldots \| S_n \| G \sqsubseteq R = \|_{i \in I} R_i$. Note that, letting $S_{mod} = \|_{i \in I} S_i \in \mathcal{A}(\Sigma)$, then S_{mod} is Σ_{uc}-admissible and $S_{mod} \| G = (\|_{i \in I} S_i) \| G \simeq S_1 \| \ldots \| S_n \| G \sqsubseteq R$ holds by the definition of the synchronous composition, which further imply $S_{mod} \in \mathcal{S}^{fo}(G, R)$. Hence, $\mathcal{S}^{fo}(G, R) \neq \emptyset$ is a necessary condition for the existence of a set of elementary supervisors which solve the modular similarity control problem.

It was shown in Theorem 1 of [64] that Σ_{uc}-similarity from G to R is a necessary and sufficient condition for $\mathcal{S}^{fo}(G, R) \neq \emptyset$ (see also Proposition 3.2 that is generalized to partial observation). We prove in the following theorem that Σ_{uc}-similarity from G to R is equivalent to that from G to R_i for each $i \in I$.

For each $i \in I$ and each $q \in Q = \prod_{i\in I} Q_i$, we let $\pi_i(q) \in Q_i$ denote the ith element of q, that is, $\pi_i(q) = q_i$ for $q = (q_1, \ldots, q_i, \ldots, q_n) \in Q_1 \times \cdots \times Q_i \times \cdots \times Q_n = Q$.

Theorem 5.1 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the modular specification $R = (Q, \Sigma, \delta, Q_0) = \|_{i\in I} R_i \in \mathcal{A}(\Sigma)$, where $I = \{1, \ldots, n\}$ and $R_i = (Q_i, \Sigma, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma)$ for each $i \in I$, G is Σ_{uc}-simulated by R if and only if, for each $i \in I$, G is Σ_{uc}-simulated by R_i.

Proof ($\Leftarrow$) For each $i \in I$, since G is Σ_{uc}-simulated by R_i, there exists a Σ_{uc}-simulation relation $\Phi_i \subseteq X \times Q_i$ from G to R_i. We show that a relation $\Phi \subseteq X \times Q = X \times (Q_1 \times \cdots \times Q_n)$ defined as

$$\Phi = \{(x, (q_1, \ldots, q_n)) \in X \times Q \mid \forall i \in I : (x, q_i) \in \Phi_i\}$$

is a Σ_{uc}-simulation relation from G to R.

- We consider any $x_0 \in X_0$. For each $i \in I$, since Φ_i is a Σ_{uc}-simulation relation from G to R_i, there exists $q_{i,0} \in Q_{i,0}$ such that $(x_0, q_{i,0}) \in \Phi_i$. Then, we have $(q_{1,0}, \ldots, q_{n,0}) \in Q_{1,0} \times \cdots \times Q_{n,0} = Q_0$. It follows that $(x_0, (q_{1,0}, \ldots, q_{n,0})) \in \Phi$.

- For any $(x, (q_1, \ldots, q_n)) \in \Phi$ and any $\sigma \in \Sigma_{uc}$, we consider any $x' \in \alpha(x, \sigma)$. By the definition of Φ, we have $(x, q_i) \in \Phi_i$ for each $i \in I$. Since Φ_i is a Σ_{uc}-simulation relation from G to R_i, there exists $q_i' \in \delta_i(q_i, \sigma)$ such that $(x', q_i') \in \Phi_i$. Then, we have $(q_1', \ldots, q_n') \in \delta_1(q_1, \sigma) \times \cdots \times \delta_n(q_n, \sigma) = \delta((q_1, \ldots, q_n), \sigma)$. It follows that $(x', (q_1', \ldots, q_n')) \in \Phi$.

Thus, Φ is a Σ_{uc}-simulation relation from G to R.

($\Rightarrow$) Since G is Σ_{uc}-simulated by R, there exists a Σ_{uc}-simulation relation $\Phi' \subseteq X \times Q = X \times (Q_1 \times \cdots \times Q_n)$ from G to R. Then, we show that, for each $i \in I$, the relation $\Phi_i' \subseteq X \times Q_i$ defined as

$$\Phi_i' = \{(x, q_i) \in X \times Q_i \mid \exists q \in Q : (x, q) \in \Phi' \wedge \pi_i(q) = q_i\}$$

is a Σ_{uc}-simulation relation from G to R_i.

- We consider any $x_0 \in X_0$. Since Φ' is a Σ_{uc}-simulation relation from G to R, there exists $q_0 = (q_{1,0}, \ldots, q_{i,0}, \ldots, q_{n,0}) \in Q_0 = Q_{1,0} \times \cdots \times Q_{i,0} \times \cdots \times Q_{n,0}$ such that $(x_0, q_0) = (x_0, (q_{1,0}, \ldots, q_{i,0}, \ldots, q_{n,0})) \in \Phi'$. It follows that $(x_0, q_{i,0}) \in \Phi'_i$.

- For any $(x, q_i) \in \Phi'_i$ and any $\sigma \in \Sigma_{uc}$, we consider any $x' \in \alpha(x, \sigma)$. By the definition of Φ'_i, there exists $q = (q_1, \ldots, q_i, \ldots, q_n) \in Q$ such that $(x, q) \in \Phi'$. Since Φ' is a Σ_{uc}-simulation relation from G to R, there exists $q' = (q'_1, \ldots, q'_i, \ldots, q'_n) \in \delta((q_1, \ldots, q_i, \ldots, q_n), \sigma) = \delta_1(q_1, \sigma) \times \cdots \times \delta_i(q_i, \sigma) \times \cdots \times \delta_n(q_n, \sigma)$ such that $(x', (q'_1, \ldots, q'_i, \ldots, q'_n)) \in \Phi'$. It follows that $(x', q'_i) \in \Phi'_i$.

Thus, for each $i \in I$, Φ'_i is a Σ_{uc}-simulation relation from G to R_i. □

Remark 5.2 Theorem 5.1 implies that Σ_{uc}-similarity from G to R can be modularly verified by checking Σ_{uc}-similarity from G to each R_i ($i \in I$), where the function $F_{G,R_i} : 2^{X \times Q_i} \to 2^{X \times Q_i}$ defined by (3.3) for G and R_i can be used. The computational complexity for obtaining the unique maximal fixpoint $W^{\uparrow}_{G,R_i} \in 2^{X \times Q_i}$ of F_{G,R_i} is $O(|\Sigma_{uc}||X|^3|Q_i|^3)$. Then, the modular verification of the nonemptiness of $\mathcal{S}^{fo}(G, R)$ can be done in $O(|\Sigma_{uc}||X|^3 \sum_{i \in I} |Q_i|^3)$.

On the other hand, the computational complexity for monolithically computing the unique maximal fixpoint $W^{\uparrow}_{G,R} \in 2^{X \times Q}$ of $F : 2^{X \times Q} \to 2^{X \times Q}$ defined by (3.3) for G and $R = \|_{i \in I} R_i$ is $O(|\Sigma_{uc}||X|^3 \prod_{i \in I} |Q_i|^3)$.

5.3 Maximally Permissive Modular Similarity Control with Modular Specification

In this section, we synthesize a set of elementary supervisors that solves the modular similarity control problem and show that they are maximally permissive with respect to R.

5.3.1 Synthesis of Elementary Supervisors

We assume that $\mathcal{S}^{fo}(G, R = \|_{i \in I} R_i) \neq \emptyset$, so that the modular similarity control problem is solvable as discussed before. Then, by Theorem 5.1, for each $i \in I$, G is Σ_{uc}-simulated by R_i.

As an elementary supervisor, we consider the supervisor

$$\bar{S}^{\uparrow} = (Y^{\uparrow}, \Sigma, \bar{\beta}^{\uparrow}, Y_0^{\uparrow}) \in \mathcal{A}(\Sigma) \tag{5.2}$$

defined in [65], that is obtained by modifying the one $S^{\uparrow} = (Y^{\uparrow}, \Sigma, \beta^{\uparrow}, Y_0^{\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (3.10), where for each $W \in Y^{\uparrow}$ and each $\sigma \in \Sigma$, the nondeterministic transition function $\bar{\beta}^{\uparrow} : Y^{\uparrow} \times \Sigma \to 2^{Y^{\uparrow}}$ is defined as

$$\bar{\beta}^{\uparrow}(W,\sigma) = \begin{cases} \beta^{\uparrow}(W,\sigma) - \{W'' \in Y^{\uparrow} \mid \Sigma^{\uparrow}(W'') = \emptyset\}, & \text{if } \exists W' \in \beta^{\uparrow}(W,\sigma) : \Sigma^{\uparrow}(W') \neq \emptyset \\ \beta^{\uparrow}(W,\sigma), & \text{otherwise.} \end{cases}$$

The following proposition shows that $\bar{S}^{\uparrow}$ is an element of $\mathcal{MPS}^{fo}(G,R)$.

Proposition 5.1 (Theorem 3 of [65]) For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, the supervisor $\bar{S}^{\uparrow} = (Y^{\uparrow}, \Sigma, \bar{\beta}^{\uparrow}, Y_0^{\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (5.2) satisfies $\bar{S}^{\uparrow} \in \mathcal{MPS}^{fo}(G,R)$.

Note that, although both $\bar{S}^{\uparrow}$ and $S^{\uparrow}$ have $2^{|X||Q|}$ states and $|\Sigma|2^{2|X||Q|}$ transitions in the worst case, the numbers of states and transitions of $\bar{S}^{\uparrow}$ are lower than those of $S^{\uparrow}$ in general.

For each $i \in I$, we let $\bar{S}_i^{\uparrow} = (Y_i^{\uparrow}, \Sigma, \bar{\beta}_i^{\uparrow}, Y_{i,0}^{\uparrow}) \in \mathcal{A}(\Sigma)$ be the supervisor defined by (5.2) based on the unique maximal fixpoint $W_{G,R_i}^{\uparrow}$ of the function $F_{G,R_i} : 2^{X \times Q_i} \to 2^{X \times Q_i}$ defined by (3.3) for G and R_i, which can be synthesized whenever G is Σ_{uc}-simulated by R_i. By Proposition 5.1, we have $\bar{S}_i^{\uparrow} \in \mathcal{MPS}^{fo}(G,R_i) \subseteq \mathcal{S}^{fo}(G,R_i)$.

We prove in the following theorem that if, for each $i \in I$, the elementary supervisor S_i enforces similarity to its corresponding elementary specification R_i, then they together enforce similarity to the modular specification $R = \|_{i \in I} R_i$.

Theorem 5.2 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the modular specification $R = (Q, \Sigma, \delta, Q_0) = \|_{i \in I} R_i \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, where $I = \{1, \ldots, n\}$ and $R_i = (Q_i, \Sigma, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma)$, we consider n supervisors $S_i = (Y_i, \Sigma, \beta_i, Y_{i,0}) \in \mathcal{S}^{fo}(G,R_i)$ $(i \in I)$. It holds that $S_1 \| \ldots \| S_n \| G \sqsubseteq R$.

Proof For each $i \in I$, we have $S_i \| G \sqsubseteq R_i$ by $S_i \in \mathcal{S}^{fo}(G,R_i)$. Then, there exists a simulation relation $\Psi_i \subseteq (Y_i \times X) \times Q_i$ from $S_i \| G$ to R_i. We show that the relation $\Psi \subseteq (Y_1 \times \cdots \times Y_n \times X) \times Q = (Y_1 \times \cdots \times Y_n \times X) \times (\prod_{i \in I} Q_i)$ defined as

$$\Psi = \{((y_1, \ldots, y_n, x), (q_1, \ldots, q_n)) \in (Y_1 \times \cdots \times Y_n \times X) \times Q \mid \forall i \in I : ((y_i, x), q_i) \in \Psi_i\}$$

is a simulation relation from $S_1 \| \ldots \| S_n \| G$ to R.

Let $\xi_{mod} : (Y_1 \times \cdots \times Y_n \times X) \times \Sigma \to 2^{Y_1 \times \cdots \times Y_n \times X}$ denote the transition function of $S_1 \| \ldots \| S_n \| G$. For each $i \in I$, let $\xi_i : (Y_i \times X) \times \Sigma \to 2^{Y_i \times X}$ denote the transition function of $S_i \| G$.

- We consider any $(y_{1,0}, \ldots, y_{n,0}, x_0) \in Y_{1,0} \times \cdots \times Y_{n,0} \times X_0$. For any $i \in I$, we have $(y_{i,0}, x_0) \in Y_{i,0} \times X_0$. Since Ψ_i is a simulation relation from $S_i \| G$ to R_i, there exists $q_{i,0} \in Q_{i,0}$ such that $((y_{i,0}, x_0), q_{i,0}) \in \Psi_i$. It follows that $(q_{1,0}, \ldots, q_{n,0}) \in Q_{1,0} \times \cdots \times Q_{n,0}$ and $((y_{1,0}, \ldots, y_{n,0}, x_0), (q_{1,0}, \ldots, q_{n,0})) \in \Psi$.
- For any $((y_1, \ldots, y_n, x), (q_1, \ldots, q_n)) \in \Psi$ and any $\sigma \in \Sigma$, we consider any $(y_1', \ldots, y_n', x') \in \xi_{mod}((y_1, \ldots, y_n, x), \sigma) = \beta_1(y_1, \sigma) \times \cdots \times \beta_n(y_n, \sigma) \times \alpha(x, \sigma)$. For each $i \in I$, we have $((y_i, x), q_i) \in \Psi_i$ by the definition of Ψ. It holds that $(y_i', x') \in \beta_i(y_i, \sigma) \times \alpha(x, \sigma) = \xi_i((y_i, x), \sigma)$. Since Ψ_i is a simulation relation from $S_i \| G$ to R_i, there exists $q_i' \in \delta_i(q_i, \sigma)$ such that $((y_i', x'), q_i') \in \Psi_i$. It follows that $(q_1', \ldots, q_n') \in \delta_1(q_1, \sigma) \times \cdots \times \delta_n(q_n, \sigma) = \delta((q_1, \ldots, q_n), \sigma)$ and $((y_1', \ldots, y_n', x'), (q_1', \ldots, q_n')) \in \Psi$.

Hence, we have $S_1 \| \ldots \| S_n \| G \sqsubseteq R$. □

5.3.2 Maximal Permissiveness of Modular Similarity Control

In this subsection, we show that, if each of the elementary supervisors S_i is maximally permissive with respect to G and R_i, namely,

$$\forall i \in I : S_i \in \mathcal{MPS}^{fo}(G, R_i)$$

holds, then so does (5.1). In order to do so, we need the following lemma.

Lemma 5.1 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$, we consider two supervisors $S_1 = (Y_1, \Sigma, \beta_1, Y_{1,0}) \in \mathcal{A}(\Sigma)$ and $S_2 = (Y_2, \Sigma, \beta_2, Y_{2,0}) \in \mathcal{A}(\Sigma)$ such that $S_1 \| G \sqsubseteq S_2 \| G$. Then, there exists a simulation relation $\Phi \subseteq (Y_1 \times X) \times (Y_2 \times X)$ from $S_1 \| G$ to $S_2 \| G$ such that

$$\forall ((y_1, x), (y_2, x')) \in \Phi : x = x'.$$

Proof Let $\xi_1 : (Y_1 \times X) \times \Sigma \to 2^{Y_1 \times X}$ and $\xi_2 : (Y_2 \times X) \times \Sigma \to 2^{Y_2 \times X}$ denote the transition functions of $S_1 \| G$ and $S_2 \| G$, respectively.

Since $S_1 \| G \sqsubseteq S_2 \| G$, there exists a simulation relation $\Phi' \subseteq (Y_1 \times X) \times (Y_2 \times X)$ from $S_1 \| G$ to $S_2 \| G$. We show that the relation $\Phi \subseteq (Y_1 \times X) \times (Y_2 \times X)$ defined as

$$\Phi = \{((y_1, x), (y_2, x)) \in (Y_1 \times X) \times (Y_2 \times X) \mid \exists \bar{x} \in X : ((y_1, x), (y_2, \bar{x})) \in \Phi'\}$$

is a simulation relation from $S_1 \| G$ to $S_2 \| G$.

- We consider any $(y_{1,0}, x_0) \in Y_{1,0} \times X_0$. Since Φ' is a simulation relation from $S_1 \| G$ to $S_2 \| G$, there exists $(y_{2,0}, \bar{x}_0) \in Y_{2,0} \times X_0$ such that $((y_{1,0}, x_0), (y_{2,0}, \bar{x}_0)) \in \Phi'$. Then, we have $(y_{2,0}, x_0) \in Y_{2,0} \times X_0$ and $((y_{1,0}, x_0), (y_{2,0}, x_0)) \in \Phi$.

- For any $((y_1, x), (y_2, x)) \in \Phi$ and any $\sigma \in \Sigma$, we consider any $(y'_1, x') \in \xi_1((y_1, x), \sigma) = \beta_1(y_1, \sigma) \times \alpha(x, \sigma)$. By the definition of Φ, there exists $\bar{x} \in X$ such that $((y_1, x), (y_2, \bar{x})) \in \Phi'$. Since Φ' is a simulation relation from $S_1 \| G$ to $S_2 \| G$, there exists $(y'_2, \bar{x}') \in \xi_2((y_2, \bar{x}), \sigma) = \beta_2(y_2, \sigma) \times \alpha(\bar{x}, \sigma)$ such that $((y'_1, x'), (y'_2, \bar{x}')) \in \Phi'$. Then, we have $(y'_2, x') \in \beta_2(y_2, \sigma) \times \alpha(x, \sigma) = \xi_2((y_2, x), \sigma)$. It follows from $((y'_1, x'), (y'_2, \bar{x}')) \in \Phi'$ and the definition of Φ that $((y'_1, x'), (y'_2, x')) \in \Phi$.

Thus, Φ is a simulation relation from $S_1 \| G$ to $S_2 \| G$. □

Now, we show in the following theorem that maximal permissiveness can be modularly achieved.

Theorem 5.3 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the modular specification $R = (Q, \Sigma, \delta, Q_0) = \|_{i \in I} R_i \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, where $I = \{1, \ldots, n\}$ and $R_i = (Q_i, \Sigma, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma)$, we consider n supervisors $S_i = (Y_i, \Sigma, \beta_i, Y_{i,0}) \in \mathcal{MPS}^{fo}(G, R_i)$ $(i \in I)$. For any $S = (Y, \Sigma, \beta, Y_0) \in \mathcal{S}^{fo}(G, R)$, it holds that $S \| G \sqsubseteq S_1 \| \ldots \| S_n \| G$.

Proof Since $R = \|_{i \in I} R_i$, Proposition 2.1, we have $R \sqsubseteq R_i$ for each $i \in I$. By $S \in \mathcal{S}^{fo}(G, R)$, we have $S \| G \sqsubseteq R$. Since the order $\sqsubseteq$ on $\mathcal{A}(\Sigma)$ is transitive, we have $S \| G \sqsubseteq R_i$, which implies together with the Σ_{uc}-admissibility of S that $S \in \mathcal{S}^{fo}(G, R_i)$. It follows from $S_i \in \mathcal{MPS}^{fo}(G, R_i)$ that $S \| G \sqsubseteq S_i \| G$. Then, by Lemma 5.1, there exists a simulation relation $\Phi_i \subseteq (Y \times X) \times (Y_i \times X)$ from $S \| G$ to $S_i \| G$ such that

$$\forall ((y, x), (y_i, x')) \in \Phi_i : x = x'.$$

We show that the relation $\Phi \subseteq (Y \times X) \times (Y_1 \times \cdots \times Y_n \times X)$ defined as

$$\Phi = \{((y,x),(y_1,\ldots,y_n,x)) \in (Y \times X) \times (Y_1 \times \cdots \times Y_n \times X) \mid \forall i \in I : ((y,x),(y_i,x)) \in \Phi_i\}$$

is a simulation relation from $S\|G$ to $S_1\|\ldots\|S_n\|G$.

Let $\xi : (Y \times X) \times \Sigma \to 2^{Y \times X}$ and $\xi_{mod} : (Y_1 \times \cdots \times Y_n \times X) \times \Sigma \to 2^{Y_1 \times \cdots \times Y_n \times X}$ denote the transition function of $S\|G$ and $S_1\|\ldots\|S_n\|G$, respectively. For each $i \in I$, let $\xi_i : (Y_i \times X) \times \Sigma \to 2^{Y_i \times X}$ denote the transition function of $S_i\|G$.

- We consider any $(y_0, x_0) \in Y_0 \times X_0$. For any $i \in I$, since Φ_i is a simulation relation from $S\|G$ to $S_i\|G$, there exists $(y_{i,0}, x_0) \in Y_{i,0} \times X_0$ such that $((y_0,x_0),(y_{i,0},x_0)) \in \Phi_i$. It follows that $((y_{1,0},\ldots,y_{n,0},x_0)) \in Y_{1,0} \times \cdots \times Y_{n,0} \times X_0$ and $((y_{1,0},\ldots,y_{n,0},x_0),(q_{1,0},\ldots,q_{n,0})) \in \Phi$.
- For any $((y,x),(y_1,\ldots,y_n,x)) \in \Psi$ and any $\sigma \in \Sigma$, we consider any $(y',x') \in \xi((y,x),\sigma)$. For each $i \in I$, we have $((y,x),(y_i,x)) \in \Phi_i$ by the definition of Φ. Since Φ_i is a simulation relation from $S\|G$ to $S_i\|G$, there exists $(y'_i,x') \in \xi_i((y_i,x),\sigma) = \beta_i(y_i,\sigma) \times \alpha(x,\sigma)$ such that $((y',x'),(y'_i,x')) \in \Phi_i$. It follows that $(y'_1,\ldots,y'_n,x') \in \beta_1(y_1,\sigma) \times \cdots \times \beta_n(y_n,\sigma) \times \alpha(x,\sigma) = \xi_{mod}((y_1,\ldots,y_n,x),\sigma)$ and $((y',x'),(y'_1,\ldots,y'_n,x')) \in \Psi$.

Hence, we have $S\|G \sqsubseteq S_1\|\ldots\|S_n\|G$. □

Remark 5.3 Proposition 5.1, Theorems 5.2 and 5.3 imply that elementary supervisors $\bar{S}_i^{\uparrow} \in \mathcal{MPS}^{fo}(G,R_i)$ $(i \in I)$ defined by (5.2) for G and R_i, respectively, are a maximal solution to the modular similarity control problem. In the worst case, $\bar{S}_i^{\uparrow} (i \in I)$ have $\sum_{i\in I} 2^{|X||Q_i|}$ states and $|\Sigma|(\sum_{i\in I} 2^{2|X||Q_i|})$ transitions in total, which are mitigated from those $2^{|X||Q|} = 2^{|X|\prod_{i\in I}|Q_i|}$ and $|\Sigma|2^{2|X||Q|} = |\Sigma|2^{2|X|\prod_{i\in I}|Q_i|}$, respectively, of the monolithic supervisor $\bar{S}^{\uparrow} \in \mathcal{MPS}^{fo}(G,R)$ defined by (5.2) for G and $R = \|_{i\in I} R_i$.

Example 5.1 We consider the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ shown in Figure 5.2 (a) and the elementary specifications $R_1 = (Q_1, \Sigma, \delta_1, Q_{1,0}) \in \mathcal{A}(\Sigma)$ and $R_2 = (Q_2, \Sigma, \delta_2, Q_{2,0}) \in \mathcal{A}(\Sigma)$ shown in Figures 5.2 (b) and (c), respectively, where $\Sigma = \{a,b,c,d,e,f,g\}$ and where $\Sigma_{uc} = \{f,g\}$. The unique maximal fixpoints $W^{\uparrow}_{G,R_1} \subseteq X \times Q_1$ and $W^{\uparrow}_{G,R_2} \subseteq X \times Q_2$ of the functions $F_{G,R_i} : 2^{X \times Q_i} \to 2^{X \times Q_i}$ $(i = 1, 2)$ defined by (3.3) for G and R_i can be obtained as

$$\begin{aligned} F^1_{G,R_1}(X \times Q_1) &= X \times Q_1 - \{x_1\} \times \{q_{1,0}, q_{1,1}, q_{1,2}, q_{1,4}\} - \{x_2\} \times \{q_{1,0}, q_{1,1}, q_{1,2}, q_{1,3}\} \\ &= F^2_{G,R_1}(X \times Q_1) \\ &= W^{\uparrow}_{G,R_1} \end{aligned}$$

and

$$\begin{aligned} F^1_{G,R_2}(X \times Q_2) &= X \times Q_2 - \{x_1\} \times \{q_{2,0}, q_{2,1}, q_{2,2}\} - \{x_2\} \times \{q_{2,0}, q_{2,1}, q_{2,2}, q_{2,3}\} \\ &= F^2_{G,R_2}(X \times Q_2) \\ &= W^{\uparrow}_{G,R_2} \end{aligned}$$

respectively. We have $(x_0, q_{1,0}) \in W^{\uparrow}_{G,R_1}$ and $(x_0, q_{2,0}) \in W^{\uparrow}_{G,R_2}$ for the initial states x_0, $q_{1,0}$, and $q_{2,0}$ of G, R_1, and R_2, respectively. Then, the similarity control problem for G and $R_1 \| R_2$ is monolithically and modularly solvable.

Using $W^{\uparrow}_{G,R_1}$ and $W^{\uparrow}_{G,R_2}$, we synthesize two maximally permissive elementary similarity-enforcing supervisors $\bar{S}^{\uparrow}_1$ and $\bar{S}^{\uparrow}_2$ defined by (5.2) for R_1 and R_2, respectively. The reachable parts of $\bar{S}^{\uparrow}_1$ and $\bar{S}^{\uparrow}_2$ are shown in Figures 5.2 (d) and (e), respectively, where

$$\begin{aligned} y_{1,0} &= \{(x_0, q_{1,0})\}, & y_{1,1} &= \{(x_0, q_{1,1})\}, \\ y_{1,2} &= \{(x_0, q_{1,2})\}, & y_{1,3} &= \{(x_1, q_{1,3})\}, \\ y_{2,0} &= \{(x_0, q_{2,0})\}, & y_{2,1} &= \{(x_0, q_{2,1})\}, \\ y_{2,2} &= \{(x_0, q_{2,2})\}, & y_{2,3} &= \{(x_1, q_{2,3})\}, \\ y_{2,4} &= \{(x_1, q_{2,4}), (x_2, q_{2,4})\}. \end{aligned}$$

The reachable part of the supervised system $\bar{S}^{\uparrow}_1 \| \bar{S}^{\uparrow}_2 \| G$ is shown in Figure 5.3 (a). It can be verified that $\bar{S}^{\uparrow}_1 \| \bar{S}^{\uparrow}_2 \| G$ is simulated by the modular specification $R = R_1 \| R_2$ shown in Figure 5.3 (b). By Proposition 5.1 and Theorem 5.3, $\bar{S}^{\uparrow}_1$ and $\bar{S}^{\uparrow}_2$ together achieve the maximal permissiveness with respect to R.

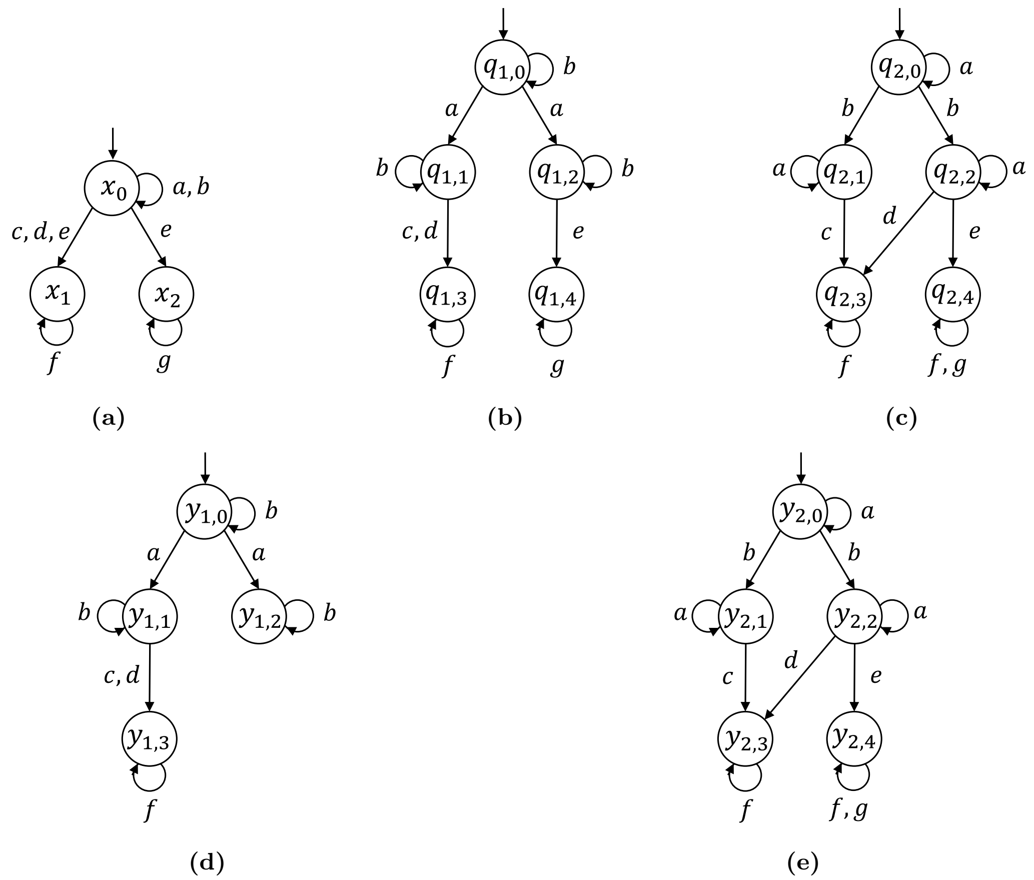
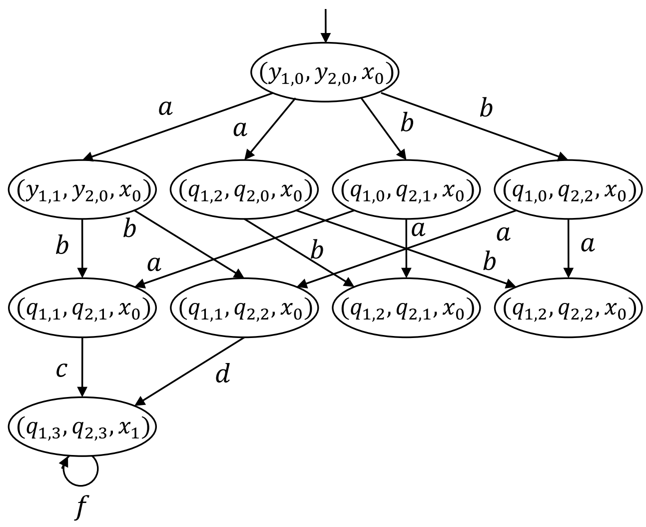


Figure 5.2: (a) System G, (b) elementary specification R_1, (c) elementary specification R_2, (d) elementary supervisor $\bar{S}_1^{\uparrow}$, and (e) elementary supervisor $\bar{S}_2^{\uparrow}$.

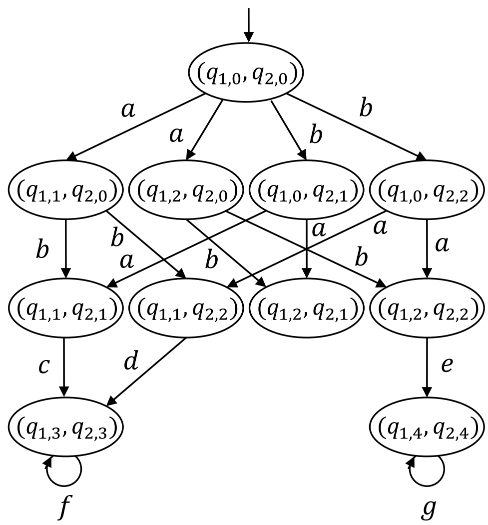
5.4 Extension Discussion

5.4.1 Generalization to Partial Observation

We assumed that the system is fully observed in the beginning of this chapter. The results of this chapter can be straightforwardly extended to the case of partial observation by assuming that the elementary supervisors have the same capability on observation (as a monolithic one does). Note that Theorem 5.1 is applicable under partial observation, which implies together with Proposition 3.2 that there exists a solution to the modular similarity control problem when $\mathcal{S}(G, R = \|_{i\in I} R_i) \neq \emptyset$. Then, G is respectively Σ_{uc}-simulated by R_i ($i \in I$), so that a set of elementary supervisors can be synthesized using the results of Chapter 3. For example, the supervisor $\bar{S}_{W\uparrow} = (Y_{W\uparrow}, \Sigma, \bar{\beta}_{W\uparrow}, Y_{W\uparrow 0}) \in \mathcal{A}(\Sigma)$ defined by (3.25) can be considered. The following corollary that extends Theorems 5.2 and 5.3 identifies a maximal solution to the modular similarity control problem under partial observation.



(a)



(b)

Figure 5.3: (a) Supervised system $\bar{S}_1^{\uparrow} \| \bar{S}_2^{\uparrow} \| G$ and (b) modular specification R.

Corollary 5.1 For the system $G = (X, \Sigma, \alpha, X_0) \in \mathcal{A}(\Sigma)$ and the modular specification $R = (Q, \Sigma, \delta, Q_0) = \|_{i \in I} R_i \in \mathcal{A}(\Sigma)$ such that G is Σ_{uc}-simulated by R, where $I = \{1, \ldots, n\}$ and $R_i = (Q_i, \Sigma, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma)$, we consider n elementary supervisors $S_i = (Y_i, \Sigma, \beta_i, Y_{i,0}) \in \mathcal{MPS}(G, R_i)$ $(i \in I)$. It holds that $S_1 \| \ldots \| S_n \| G \sqsubseteq R$, and for any $S = (Y, \Sigma, \beta, Y_0) \in \mathcal{S}(G, R)$, it holds that $S \| G \sqsubseteq S_1 \| \ldots \| S_n \| G$.

Remark 5.4 For each $i \in I$, let $\bar{S}_{W^{\uparrow}_{G,R_i}} = (Y_{W^{\uparrow}_{G,R_i}}, \Sigma, \bar{\beta}_{W^{\uparrow}_{G,R_i}}, Y_{W^{\uparrow}_{G,R_i} 0}) \in \mathcal{MPS}(G, R_i)$ denote the supervisor defined by (3.25) for G and R_i. Then, Corollary 5.1 guarantees that elementary supervisors $\bar{S}_{W^{\uparrow}_{G,R_i}}$ $(i \in I)$ are a maximal solution to the modular similarity control problem for the partially observed system G and the modular specification

$R = \|_{i \in I} R_i$. The monolithic supervisor $\bar{S}_{W^{\uparrow}_{G,R}} = (Y_{W^{\uparrow}_{G,R}}, \Sigma, \bar{\beta}_{W^{\uparrow}_{G,R}}, Y_{W^{\uparrow}_{G,R}0}) \in \mathcal{MPS}(G, R)$ defined by (3.25) for G and R has at most $2^{2|X||Q|+|\Sigma_c \cap \Sigma_{uo}|} = 2^{2|X|(\prod_{i \in I}|Q_i|)+|\Sigma_c \cap \Sigma_{uo}|}$ states and $|\Sigma|2^{4|X||Q|+2|\Sigma_c \cap \Sigma_{uo}|} = |\Sigma|2^{4|X|(\prod_{i \in I}|Q_i|)+2|\Sigma_c \cap \Sigma_{uo}|}$ transitions. By using the modular control architecture, these upper bound numbers are reduced to $\sum_{i \in I} 2^{2|X||Q_i|+|\Sigma_c \cap \Sigma_{uo}|}$ and $|\Sigma|(\sum_{i \in I} 2^{4|X||Q_i|+2|\Sigma_c \cap \Sigma_{uo}|})$, respectively.

5.4.2 Discussion on Nonblocking Modular Similarity Control

We discuss the imposition of nonblockingness on modular similarity control in this subsection, so that the marking of the system $G = (X, \Sigma, \alpha, X_0, X_m) \in \mathcal{A}(\Sigma)$ and the modular specification $R = (Q, \Sigma, \delta, Q_0, Q_m) = \|_{i \in I} R_i \in \mathcal{A}(\Sigma)$ with $R_i = (Q_i, \Sigma, \delta_i, Q_{i,0}, Q_{i,m}) \in \mathcal{A}(\Sigma)$ is taken into consideration.

Using the algorithm NBSR introduced in Chapter 4, a maximally permissive elementary nonblocking supervisor can be synthesized, if it exists, for each of the elementary specifications. For this purpose, the supervisor $S^{\uparrow} = (Y^{\uparrow}, \Sigma, \beta^{\uparrow}, Y_0^{\uparrow}, Y^{\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (4.5) or the one $\tilde{S}_{W^{\uparrow}} = (\tilde{Y}_{W^{\uparrow}}, \Sigma, \tilde{\beta}_{W^{\uparrow}}, \tilde{Y}_{W^{\uparrow}0}, \tilde{Y}_{W^{\uparrow}}) \in \mathcal{A}(\Sigma)$ defined by (4.4) can be used as the input elementary supervisor, according to whether the system under consideration is fully or partially observed. For each $i \in I$, we assume that $S_i = (Y_i, \Sigma, \beta_i, Y_{i,0}, Y_i) \in \mathcal{NS}(G, R_i)$ itself is nonblocking. However, the supervised system $S_1 \| \ldots \| S_n \| G$ is possibly blocking, due to conflict between the elementary supervisors.

In the modular control of deterministic systems, a nonconflicting property on languages is derived to guarantee that the system supervised by multiple supervisors is nonblocking. Formally, for the deterministic system $G \in \mathcal{A}(\Sigma)$ and deterministic nonblocking supervisors $S_1, S_2 \in \mathcal{A}(\Sigma)$, $S_1 \| S_2 \| G$ is nonblocking if and only if $L_m(S_1 \| G)$ and $L_m(S_2 \| G)$ are nonconflicting, namely,

$$\overline{L_m(S_1 \| G) \cap L_m(S_2 \| G)} = \overline{L_m(S_1 \| G)} \cap \overline{L_m(S_2 \| G)}$$

holds [33]. However, the nonconflicting property is not sufficient to guarantee the nonblockingness of modular control in the nondeterministic setting.

As in the deterministic setting [39,40], a scheme of conflict resolution for nondeterministic systems is needed to achieve the nonblocking behavior.

On the other hand, by computing the synchronous composition $S_{mod} = \|_{i \in I} S_i$ of the elementary supervisors, we can obtain a monolithic supervisor to which the algorithm NBSR can be applied. However, synchronous composition operation dramatically increases the state number of the supervisor (exponential state space explosion). As the complexity of NBSR is polynomial in the size of the input supervisor, such an approach is computationally inefficient.

5.5 Conclusion

In this chapter, we considered the similarity control problem for the monolithic system with the modular specification. To mitigate the computational complexity caused by the exponential state space explosion, the modular control architecture was applied. It was proved that the modular method is applicable whenever the similarity control problem is solvable. For each of the elementary specifications, a maximally permissive similarity-enforcing elementary supervisor was synthesized. We showed that these supervisors together achieve the maximal permissiveness with respect to the modular specification.

Chapter 6

Modular Similarity Control of Composite Systems

In this chapter, we consider the similarity control problem where the composite system under consideration consists of multiple local subsystems, each of which is assigned a nondeterministic local subspecification. As discussed in Chapter 5, the monolithic synthesis suffers from a severe computational burden.

We provide a new condition for the applicability of the modular control architecture, which is weaker than the one used in [68,69]. To modularly solve the similarity control problem, we synthesize a local supervisor for each subsystem and its corresponding local subspecification, so that the entire specification is modularly achieved in the sense that the supervised system is simulated by the specification. Moreover, we prove that the same permissiveness as a maximally permissive monolithic supervisor can be achieved by these local supervisors.

6.1 Problem Formulation

We consider the similarity control problem of the composite system $G = (X, \Sigma, \alpha, X_0) = \|_{i\in I} G_i \in \mathcal{A}(\Sigma)$ given as the synchronous composition of n nondeterministic subsystems $G = (X_i, \Sigma_i, \alpha_i, X_{i,0}) \in \mathcal{A}(\Sigma_i)$ ($i \in I = \{1, \ldots, n\}$). For each $i \in I$, G_i is assigned a nondeterministic local subspecification $R_i = (Q_i, \Sigma_i, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma_i)$. The entire specification $R = (Q, \Sigma, \delta, Q_0) \in \mathcal{A}(\Sigma)$ is denoted as the synchronous composition of all these local subspecifications $R = \|_{i\in I} R_i$. We do not tackle the issue of nonblockingness in this chapter, so the marked state sets are dropped. Besides, we assume that all the events of the composite system are observable.

For a subsystem $G_i \in \mathcal{A}(\Sigma_i)$ and its corresponding local subspecification $R_i \in \mathcal{A}(\Sigma_i)$, the event set Σ_i is partitioned into controllable and uncontrollable event sets $\Sigma_{i,c}$ and $\Sigma_{i,uc}$, respectively. This means that a local supervisor $S_i \in \mathcal{A}(\Sigma_i)$ for G_i has to be $\Sigma_{i,uc}$-admissible (with respect to G_i). On the other hand, for the monolithic control of the entire $G = \|_{i\in I} G_i \in \mathcal{A}(\Sigma)$ and $R = \|_{i\in I} R_i \in \mathcal{A}(\Sigma)$, a controllable event of G is controllable in at least one subsystem. Namely, the set Σ_c of controllable events is given as

$$\Sigma_c = \bigcup_{i\in I} \Sigma_{i,c}. \tag{6.1}$$

Then, the uncontrollable event set Σ_{uc} is obtained as

$$\Sigma_{uc} = \Sigma - \Sigma_c. \tag{6.2}$$

When the monolithic synthesis approach is used to solve the similarity control problem for G and R, the supervisor $\bar{S}^{\uparrow} = (Y^{\uparrow}, \Sigma, \bar{\beta}^{\uparrow}, Y_0^{\uparrow}) \in \mathcal{MPS}^{fo}(G,R)$ defined by (5.2) can be used. Since G and R have at most $\prod_{i\in I} |X_i|$ and $\prod_{i\in I} |Q_i|$ (reachable) states, respectively, $\bar{S}^{\uparrow}$ has at most $2^{|X||Q|} = 2^{\prod_{i\in I}|X_i||Q_i|}$ states and $|\Sigma| 2^{2|X||Q|} = |\Sigma| 2^{2\prod_{i\in I}|X_i||Q_i|}$ transitions in the worst case.

The subsystem G_i supervised by a local supervisor $S_i = (Y_i, \Sigma, \beta_i, Y_{i,0}) \in \mathcal{A}(\Sigma_i)$ is denoted by $S_i \| G_i$. Then, $\|_{i\in I}(S_i \| G_i)$ models the composite system G under the control of a set of local supervisors S_i $(i \in I)$.

Formally, in this chapter, we synthesize $\Sigma_{i,uc}$-admissible local supervisors $S_i \in \mathcal{A}(\Sigma)$ $(i \in I)$ which satisfy $\|_{i\in I}(S_i\|G_i) \sqsubseteq R = \|_{i\in I} R_i$ and

$$\forall S \in \mathcal{S}^{fo}(G,R) : S\|G \sqsubseteq \|_{i\in I}(S_i\|G_i). \tag{6.3}$$

Remark 6.1 The control mechanism of n local supervisor S_i $(i \in I)$ for the composite system G can be explained as follows. Let $((y_1,x_1),\ldots,(y_n,x_n)) \in \mathit{Reach}(\|_{i\in I}(S_i\|G_i))$ be the current state of $\|_{i\in I}(S_i\|G_i)$. An event $\sigma \in \Sigma$ is possible if and only if, for each $i \in I(\sigma) = \{i \in I \mid \sigma \in \Sigma_i\}$, it is possible at x_i in G_i and enabled at y_i by S_i.

6.2 Applicability of Modular Control for Composite Systems

In this section, we consider the existence condition for $\Sigma_{i,uc}$-admissible local supervisors $S_i \in \mathcal{A}(\Sigma)$ $(i \in I)$ which satisfy $\|_{i\in I}(S_i \| G_i) \sqsubseteq R = \|_{i\in I} R_i$.

Given $G = \|_{i\in I} G_i \in \mathcal{A}(\Sigma)$ and $R = \|_{i\in I} R_i \in \mathcal{A}(\Sigma)$, where the controllability status is defined as (6.1) and (6.2), it was shown in [68] that, under the assumption

$$\forall i, j \in I : i \neq j \Rightarrow \Sigma_{i,uc} \cap \Sigma_j = \emptyset, \tag{6.4}$$

the Σ_{uc}-similarity from G to R is equivalent to the $\Sigma_{i,uc}$-similarity from G_i to R_i for each $i \in I$.

Proposition 6.1 (Theorem 1 of [68]) For the system $G = (X, \Sigma, \alpha, X_0) = \|_{i\in I} G_i \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) = \|_{i\in I} R_i \in \mathcal{A}(\Sigma)$, where $I = \{1, \ldots, n\}$, $G_i = (X_i, \Sigma_i, \alpha_i, X_{i,0}) \in \mathcal{A}(\Sigma_i)$, and $R_i = (Q_i, \Sigma_i, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma_i)$, we assume that the condition (6.4) holds. Then, G is Σ_{uc}-simulated by R if and only if G_i is $\Sigma_{i,uc}$-simulated by R_i for each $i \in I$.

Here, instead of (6.4), we make the following assumptions

$$\forall i, j \in I : i \neq j \Rightarrow \Sigma_{i,uc} \cap \Sigma_{j,c} = \emptyset, \tag{6.5}$$

and

$$\forall x = (x_1, \ldots, x_n) \in Reach(G), \forall \sigma \in \Sigma_{uc} : [\exists i \in I : \alpha_i(x_i, \sigma) \neq \emptyset] \Rightarrow \alpha(x, \sigma) \neq \emptyset. \tag{6.6}$$

The assumption (6.5) means that an event that is uncontrollable in a subsystem cannot be controllable in other subsystems. This assumption is widely used in decentralized supervisory control, see for example [35]. The assumption (6.6) denotes that, for any reachable state of G, any uncontrollable event which is feasible in at least one subsystem is globally feasible.

Note that (6.4) is stronger than (6.5) and (6.6), namely, if (6.4) holds, then (6.5) and (6.6) hold, but the converse does not necessarily hold. We provide an example where (6.5) and (6.6) hold but (6.4) is not satisfied.

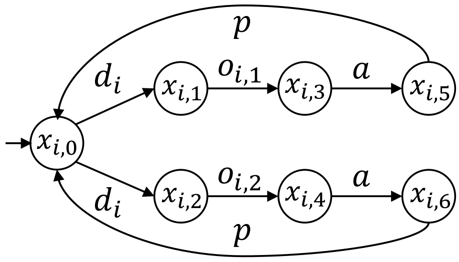


Figure 6.1: Subsystem G_i of $G = G_1 \| G_2$.

Example 6.1 We consider a manufacturing system consisting of an input buffer, an output buffer, a robot arm, and two processing robots. This system can be modeled as the synchronous composition of the subsystems $G_i \in \mathcal{A}(\Sigma_i)$ $(i = 1, 2)$ shown in Figure 6.1. The robot arm nondeterministically picks one of two kinds of parts from the input buffer and then drops it to the ith processing robot (event d_i). Depending on the kind of the part the robot received, it transits to one of the two states $x_{i,1}$ and $x_{i,2}$, at which it performs operations $o_{i,1}$ and $o_{i,2}$ to process the part, respectively. The two robots assemble their processed parts (event a) after procession. Finally, the arm moves the assembled product to the output buffer (event p). Hence, the event set Σ_i of G_i is given as $\Sigma_i = \{d_i, o_{i,1}, o_{i,2}, a, p\}$ for $i = 1, 2$. Suppose that the behavior of the arm is uncontrollable, that is, $\Sigma_{i,uc} = \{d_i, p\}$. Since $p \in \Sigma_{1,uc} \cap \Sigma_2$, (6.4) is not satisfied. On the other hand, the system $G = G_1 \| G_2$ satisfies (6.5) and (6.6).

If any uncontrollable event is not shared by multiple subsystems, the conditions (6.5) and (6.6) are satisfied. Besides, in Example 6.1, the common uncontrollable event $p \in \Sigma_{uc}$ exists and is triggered by the execution of the common controllable event $a \in \Sigma_c$. In such a case, (6.6) is also satisfied. Finding a way to further relax the assumption (6.6) constitutes a challenge for future investigations.

When (6.4) holds (which can be easily verified), the solvability of the similarity control problem can be modularly verified as suggested by Proposition 6.1, so that the computational complexity is reduced from $O(|\Sigma_{uc}| \prod_{i \in I} |X_i|^3 |Q_i|^3)$ of the monolithic verification to $O(\sum_{i \in I} |\Sigma_{i,uc}||X_i|^3|Q_i|^3)$. We show in the following theorem that this modular verification is also valid when (6.5) and (6.6) hold.

Theorem 6.1 For the system $G = (X, \Sigma, \alpha, X_0) = \|_{i \in I} G_i \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) = \|_{i \in I} R_i \in \mathcal{A}(\Sigma)$, where $I = \{1, \ldots, n\}$, $G_i = (X_i, \Sigma_i, \alpha_i, X_{i,0}) \in \mathcal{A}(\Sigma_i)$, and $R_i = (Q_i, \Sigma_i, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma_i)$, we assume that the conditions (6.5) and (6.6) hold. Then, G is Σ_{uc}-simulated by R if and only if G_i is $\Sigma_{i,uc}$-simulated by R_i for each $i \in I$.

Proof ($\Leftarrow$) By Propositions 5 and 6 of [68], G is Σ_{uc}-simulated by R if G_i is $\Sigma_{i,uc}$-simulated by R_i for each $i \in I$.

($\Rightarrow$) Since G is Σ_{uc}-simulated by R, there exists a Σ_{uc}-simulation relation $\Phi \subseteq X \times Q$ from G to R. For each $i \in I$, we show that the relation $\Phi_i \subseteq X_i \times Q_i$ defined as

$$\Phi_i = \{(x_i, q_i) \in X_i \times Q_i \mid \exists (x, q) \in \Phi : x \in \mathit{Reach}(G) \wedge \pi_i(x) = x_i \wedge \pi_i(q) = q_i\}$$

is a $\Sigma_{i,uc}$-simulation relation from G_i to R_i, where $\pi_i(x) = x_i$ and $\pi_i(q) = q_i$ for each $i \in I$, each $x = (x_1, \ldots, x_i, \ldots, x_n) \in \prod_{i \in I} X_i = X$, and each $q = (q_1, \ldots, q_i, \ldots, q_n) \in \prod_{i \in I} Q_i = Q$.

- For any $x_{i,0} \in X_{i,0}$, there exists $x_0 \in X_0 = \prod_{i \in I} X_{i,0} \subseteq \mathit{Reach}(G)$ such that $\pi_i(x_0) = x_{i,0}$. Since Φ is a Σ_{uc}-simulation relation from G to R, there exists $q \in Q_0 = \prod_{i \in I} Q_{i,0}$ such that $(x_0, q_0) \in \Phi$. Letting $q_{i,0} = \pi_i(q_0)$, we have $q_{i,0} \in Q_{i,0}$ and $(x_{i,0}, q_{i,0}) \in \Phi_i$.
- For any $(x_i, q_i) \in \Phi_i$ and any $\sigma \in \Sigma_{i,uc}$, we consider any $x_i' \in \alpha_i(x_i, \sigma)$. By (6.5), we have $\sigma \in \Sigma - \bigcup_{i \in I} \Sigma_{i,c} = \Sigma_{uc}$. It follows from the definition of Φ_i that there exists $(x, q) \in \Phi$ such that $x \in \mathit{Reach}(G)$, $\pi_i(x) = x_i$, and $\pi_i(q) = q_i$. By $x_i' \in \alpha_i(x_i, \sigma) \neq \emptyset$ and (6.6), we have $\alpha(x, \sigma) \neq \emptyset$. Since $\sigma \in \Sigma_{i,uc} \subseteq \Sigma_i$ and $x_i' \in \alpha_i(x_i, \sigma)$, by the definition of the synchronous composition, there exists $x' \in \alpha(x, \sigma) \subseteq \mathit{Reach}(G)$ such that $\pi_i(x') = x_i'$. Since Φ is a Σ_{uc}-simulation relation from G to R, there exists $q' \in \delta(q, \sigma)$ such that $(x', q') \in \Phi$. Letting $q_i' = \pi_i(q')$, since $\sigma \in \Sigma_i$, we have $q_i' \in \delta_i(q_i, \sigma)$. It follows that $(x_i', q_i') \in \Phi_i$.

Hence, G_i is $\Sigma_{i,uc}$-simulated by R_i. $\square$

If (6.4) does not hold, it is worth further verifying (6.5) and (6.6). Although (6.6) requires the construction of G, whose computational complexity is $O(|\Sigma| \prod_{i \in I} |X_i|^2)$, it is generally lower than that $O(|\Sigma_{uc}| \prod_{i \in I} |X_i|^3 |Q_i|^3)$ of the monolithic verification.

6.3 Maximal Permissiveness

In this section, we show that maximal permissiveness can be achieved by modular control as long as (6.5) and (6.6) hold. The following proposition just says that, if each of the local supervisors enforces similarity for the local subsystem and its subspecification, then the entire specification is satisfied by modular control.

Proposition 6.2 Let the system $G = \|_{i\in I} G_i \in \mathcal{A}(\Sigma)$ and the specification $R = \|_{i\in I} R_i \in \mathcal{A}(\Sigma)$ be such that G is Σ_{uc}-simulated by R, where $I = \{1, \ldots, n\}$ and $G_i, R_i \in \mathcal{A}(\Sigma_i)$. We assume that the conditions (6.5) and (6.6) hold. For each $i \in I$, we consider any local supervisor $S_i \in \mathcal{S}^{fo}(G_i, R_i)$. Then, it holds that $\|_{i\in I}(S_i \| G_i) \sqsubseteq R$.

Proof For each $i \in I$, $S_i \in \mathcal{S}^{fo}(G_i, R_i)$ satisfies $S_i \| G_i \sqsubseteq R_i$. It follows from Corollary 1 of [68] that $\|_{i\in I}(S_i \| G_i) \sqsubseteq \|_{i\in I} R_i = R$. □

We consider the supervisor $S^{\uparrow} = (Y^{\uparrow}, \Sigma, \beta^{\uparrow}, Y_0^{\uparrow}) \in \mathcal{A}(\Sigma)$ defined by (3.10). For each $i \in I$, if $G_i \in \mathcal{A}(\Sigma_i)$ is $\Sigma_{i,uc}$-simulated by $R_i \in \mathcal{A}(\Sigma_i)$, the local supervisor $S_i^{\uparrow} = (Y_i^{\uparrow}, \Sigma_i, \beta_i^{\uparrow}, Y_{i,0}^{\uparrow}) \in \mathcal{A}(\Sigma_i)$ defined by (3.10) is synthesized using the unique maximal fixpoint $W^{\uparrow}_{G_i,R_i} \in 2^{X_i \times Q_i}$ of the function $F_{G_i,R_i} : 2^{X_i \times Q_i} \to 2^{X_i \times Q_i}$ defined by (3.3) for G_i and R_i. These local supervisors have at most $\sum_{i\in I} 2^{|X_i||Q_i|}$ states and $\sum_{i\in I} |\Sigma_i| 2^{2|X_i||Q_i|}$ transitions in total, which are much fewer than $2^{\prod_{i\in I}|X_i||Q_i|}$ states and $|\Sigma| 2^{2\prod_{i\in I}|X_i||Q_i|}$ transitions, respectively, of the monolithic counterpart $S^{\uparrow}$.

Now, we show a nice property of $S_i^{\uparrow}$, which is useful for proving that the maximal permissiveness with respect to the entire specification can be achieved by these local supervisors $S_i^{\uparrow} (i \in I)$. In order to do so, we introduce the following operation.

Definition 6.1 For an arbitrary subset $\Sigma' \subseteq \Sigma$ and any automaton $G' = (X', \Sigma', \alpha', X_0') \in \mathcal{A}(\Sigma')$ over it, we define the Σ-complemented automaton $C^{\Sigma}(G') \in \mathcal{A}(\Sigma)$ of G' as $C^{\Sigma}(G') = (X', \Sigma, \alpha^{\Sigma}, X_0')$, where the nondeterministic transition function $\alpha^{\Sigma} : X' \times \Sigma \to 2^{X'}$ is defined as

$$\alpha^{\Sigma}(x, \sigma) = \begin{cases} \alpha'(x, \sigma), & \text{if } \sigma \in \Sigma' \\ \{x\}, & \text{if } \sigma \in \Sigma - \Sigma' \end{cases}$$

for each $x \in X'$ and each $\sigma \in \Sigma$.

For $G' \in \mathcal{A}(\Sigma')$, the event set of its Σ-complemented automaton $C^{\Sigma}(G') \in \mathcal{A}(\Sigma)$ is extended to $\Sigma \supseteq \Sigma'$ by adding self-loops by the events in $\Sigma - \Sigma'$ to all states. Note that, for an arbitrary set of automata $G_i \in \mathcal{A}(\Sigma_i)$ $(i \in I)$ with $\bigcup_{i\in I} \Sigma_i = \Sigma$, it is trivial that

$$\|_{i\in I} G_i = \|_{i\in I} C^{\Sigma}(G_i) \in \mathcal{A}(\Sigma).$$

Then, for each $i \in I$, we consider the Σ-complemented subsystem $C^{\Sigma}(G_i) \in \mathcal{A}(\Sigma)$ and its Σ-complemented local subspecification $C^{\Sigma}(R_i) \in \mathcal{A}(\Sigma)$. We let $\Sigma_{i,uc}$ be the uncontrollable event set of $C^{\Sigma}(G_i)$, so the rest events in $\Sigma - \Sigma_{i,uc} (= \Sigma_{i,c} \cup (\Sigma - \Sigma_i))$ are controllable.

Now, we prove in the following lemma that $C^{\Sigma}(S_i^{\uparrow}) \in \mathcal{A}(\Sigma)$ is an element of $\mathcal{MPS}^{fo}(C^{\Sigma}(G_i), C^{\Sigma}(R_i))$.

Lemma 6.1 For each $i \in I = \{1, \ldots, n\}$, let the subsystem $G_i \in \mathcal{A}(\Sigma_i)$ and the subspecification $R_i \in \mathcal{A}(\Sigma_i)$ be such that G_i is $\Sigma_{i,uc}$-simulated by R_i. Then, the supervisor $S_i^{\uparrow} = (Y_i^{\uparrow}, \Sigma_i, \beta_i^{\uparrow}, Y_{i,0}^{\uparrow}) \in \mathcal{A}(\Sigma_i)$ defined by (3.10) (for G_i and R_i) satisfies $C^{\Sigma}(S_i^{\uparrow}) \in \mathcal{MPS}^{fo}(C^{\Sigma}(G_i), C^{\Sigma}(R_i))$, where $\Sigma = \bigcup_{i\in I} \Sigma_i$.

Proof For each $i \in I$, to prove $C^{\Sigma}(S_i^{\uparrow}) \in \mathcal{MPS}^{fo}(C^{\Sigma}(G_i), C^{\Sigma}(R_i))$, we show ${}^{\Sigma}S_i^{\uparrow} = C^{\Sigma}(S_i^{\uparrow})$, where ${}^{\Sigma}S_i^{\uparrow} = ({}^{\Sigma}Y_i^{\uparrow}, \Sigma, {}^{\Sigma}\beta_i^{\uparrow}, {}^{\Sigma}Y_{i,0}^{\uparrow}) \in \mathcal{A}(\Sigma)$ is the supervisor defined by (3.10) for $C^{\Sigma}(G_i)$ and $C^{\Sigma}(R_i)$.

Since G_i is $\Sigma_{i,uc}$-simulated by R_i and $\Sigma_{i,uc}$ is also the uncontrollable event set of $C^{\Sigma}(G_i)$, $C^{\Sigma}(G_i)$ is $\Sigma_{i,uc}$-simulated by $C^{\Sigma}(R_i)$. Moreover, we have $W^{\uparrow}_{G_i,R_i} = W^{\uparrow}_{C^{\Sigma}(G_i),C^{\Sigma}(R_i)}$, where $W^{\uparrow}_{G_i,R_i}$ and $W^{\uparrow}_{C^{\Sigma}(G_i),C^{\Sigma}(R_i)}$ are the unique maximal fixpoints of F_{G_i,R_i} and $F_{C^{\Sigma}(G_i),C^{\Sigma}(R_i)}$ defined by (3.3), respectively. Since all the transitions of $C^{\Sigma}(G_i)$ and $C^{\Sigma}(R_i)$ by the events in $\Sigma - \Sigma_i$ are deterministic self-loops at each of their states, by the definition of ${}^{\Sigma}\beta_i^{\uparrow}$, so are the transitions of ${}^{\Sigma}S_i^{\uparrow}$ by the events in $\Sigma - \Sigma_i$, which are defined at all the states of ${}^{\Sigma}S_i^{\uparrow}$. In addition, we have ${}^{\Sigma}Y_i^{\uparrow} = Y_i^{\uparrow}$, ${}^{\Sigma}Y_{i,0}^{\uparrow} = Y_{i,0}^{\uparrow}$, and ${}^{\Sigma}\beta_i^{\uparrow}(W, \sigma) = \beta_i^{\uparrow}(W, \sigma)$ for each $W \in {}^{\Sigma}Y_i^{\uparrow} = Y_i^{\uparrow}$ and each $\sigma \in \Sigma_i$. Hence, we have ${}^{\Sigma}S_i^{\uparrow} = C^{\Sigma}(S_i^{\uparrow})$. By Proposition 3.4, we have $C^{\Sigma}(S_i^{\uparrow}) = {}^{\Sigma}S_i^{\uparrow} \in \mathcal{MPS}^{fo}(C^{\Sigma}(G_i), C^{\Sigma}(R_i))$. $\square$

Now, we prove the maximal permissiveness of the modular similarity control.

Theorem 6.2 Let the system $G = (X, \Sigma, \alpha, X_0) = \|_{i\in I} G_i \in \mathcal{A}(\Sigma)$ and the specification $R = (Q, \Sigma, \delta, Q_0) = \|_{i\in I} R_i \in \mathcal{A}(\Sigma)$ be such that G is Σ_{uc}-simulated by R, where $I =$

$\{1,\ldots,n\}$, $G_i = (X_i, \Sigma_i, \alpha_i, X_{i,0}) \in \mathcal{A}(\Sigma_i)$, and $R_i = (Q_i, \Sigma_i, \delta_i, Q_{i,0}) \in \mathcal{A}(\Sigma_i)$. We assume that the conditions (6.5) and (6.6) hold. For each $i \in I$, we consider the supervisor $S_i^{\uparrow} = (Y_i^{\uparrow}, \Sigma_i, \beta_i^{\uparrow}, Y_{i,0}^{\uparrow}) \in \mathcal{A}(\Sigma_i)$ defined by (3.10). Then, $S\|G \sqsubseteq \|_{i\in I}(S_i^{\uparrow}\|G_i)$ holds for an arbitrary $S = (Y, \Sigma, \beta, Y_0) \in \mathcal{S}^{fo}(G,R)$.

Proof By the definition of the Σ-complemented automaton, we have $\|_{i\in I}(S_i^{\uparrow}\|G_i) = \|_{i\in I}(C^{\Sigma}(S_i^{\uparrow})\|C^{\Sigma}(G_i))$. By Proposition 2.1, it is sufficient to show that $S\|G \sqsubseteq C^{\Sigma}(S_i^{\uparrow})\|C^{\Sigma}(G_i)$ holds for each $i \in I$.

For each $i \in I$, by the definition of the synchronous composition and the Σ-complemented automaton, we have

$$
\begin{aligned}
&S\|G = S\|(\|_{i\in I}G_i) \simeq \\
&\quad (S\|G_1\|\ldots\|G_{i-1}\|G_{i+1}\|\ldots\|G_n)\|G_i = \\
&\quad\quad (S\|G_1\|\ldots\|G_{i-1}\|G_{i+1}\|\ldots\|G_n)\|C^{\Sigma}(G_i).
\end{aligned}
$$

That is, $S\|G$ can be seen as the Σ-complemented subsystem $C^{\Sigma}(G_i) = (X_i, \Sigma, \alpha_i^{\Sigma}, X_{i,0}) \in \mathcal{A}(\Sigma)$ controlled by the virtual supervisor $\tilde{S}_i = (\tilde{Y}_i, \Sigma, \tilde{\beta}_i, \tilde{Y}_{i,0}) = S\|G_1\|\ldots\|G_{i-1}\|G_{i+1}\|\ldots\|G_n \in \mathcal{A}(\Sigma)$. Now, for each $i \in I$, we show that $\tilde{S}_i \in \mathcal{S}^{fo}(C^{\Sigma}(G_i), C^{\Sigma}(R_i))$.

To show that $\tilde{S}_i$ is $\Sigma_{i,uc}$-admissible with respect to $C^{\Sigma}(G_i)$, we consider any $((y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n), x_i) \in \mathit{Reach}(\tilde{S}_i\|C^{\Sigma}(G_i))$ and any $\sigma \in \Sigma_{i,uc}$ such that $\alpha_i^{\Sigma}(x_i, \sigma) \neq \emptyset$. By the definition of the synchronous composition, we have $(x_1, \ldots, x_i, \ldots, x_n) \in \mathit{Reach}(G)$ and $(y, (x_1, \ldots, x_i, \ldots, x_n)) \in \mathit{Reach}(S\|G)$. It follows from $\sigma \in \Sigma_{i,uc} \subseteq \Sigma_i$ and the definition of the Σ-complemented automaton that $\alpha_i(x_i, \sigma) = \alpha_i^{\Sigma}(x_i, \sigma) \neq \emptyset$. By (6.5), we have $\sigma \in \Sigma - \bigcup_{i\in I}\Sigma_{i,c} = \Sigma_{uc}$. Then, it follows from (6.6) that $\alpha((x_1, \ldots, x_i, \ldots, x_n), \sigma) \neq \emptyset$, which further implies that $\alpha_j(x_j, \sigma) \neq \emptyset$ holds for each $j \in I(\sigma)$. Since $S \in \mathcal{S}^{fo}(G,R)$, it is Σ_{uc}-admissible with respect to G. It follows that $\beta(y, \sigma) \neq \emptyset$. Hence, we have $\tilde{\beta}_i((y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n), \sigma) \neq \emptyset$ by the definition of the synchronous composition.

We next show $\tilde{S}_i\|C^{\Sigma}(G_i) \sqsubseteq C^{\Sigma}(R_i)$. By Proposition 2.1, we have $R = \|_{i\in I}R_i = \|_{i\in I}C^{\Sigma}(R_i) \sqsubseteq C^{\Sigma}(R_i)$. It follows from $S \in \mathcal{S}^{fo}(G,R)$ that $S\|G \sqsubseteq R$. Since $\tilde{S}_i\|C^{\Sigma}(G_i) \simeq S\|G$, we have $\tilde{S}_i\|C^{\Sigma}(G_i) \simeq S\|G \sqsubseteq R \sqsubseteq C^{\Sigma}(R_i)$. By the transitivity of $\sqsubseteq$, we have

$\tilde{S}_i \| C^\Sigma(G_i) \sqsubseteq C^\Sigma(R_i)$. Hence, it holds that $\tilde{S}_i \in \mathcal{S}^{fo}(C^\Sigma(G_i), C^\Sigma(R_i))$.

By Lemma 6.1, $C^\Sigma(S_i^\uparrow) \in \mathcal{MPS}^{fo}(C^\Sigma(G_i), C^\Sigma(R_i))$ holds for each $i \in I$. It follows that $S\|G \simeq \tilde{S}_i\|C^\Sigma(G_i) \sqsubseteq C^\Sigma(S_i^\uparrow)\|C^\Sigma(G_i)$. By the transitivity of $\sqsubseteq$, we have $S\|G \sqsubseteq C^\Sigma(S_i^\uparrow)\|C^\Sigma(G_i)$. Thus, $S\|G \sqsubseteq \|_{i\in I}(C^\Sigma(S_i^\uparrow)\|C^\Sigma(G_i)) = \|_{i\in I}(S_i^\uparrow\|G_i)$ holds by Proposition 2.1. □

Remark 6.2 Since $S^\uparrow \in \mathcal{MPS}^{fo}(G,R) \subseteq \mathcal{S}^{fo}(G,R)$, we have $S^\uparrow\|G \sqsubseteq \|_{i\in I}(S_i^\uparrow\|G_i)$ as shown in Theorem 6.2 under the assumptions (6.5) and (6.6). Letting $S' = \|_{i\in I}S_i^\uparrow$, we have $S'\|G \simeq \|_{i\in I}(S_i^\uparrow\|G_i)$. Moreover, it can be easily proved that $S' \in \mathcal{S}^{fo}(G,R)$, which implies together with $S^\uparrow \in \mathcal{MPS}^{fo}(G,R)$ that $\|_{i\in I}(S_i^\uparrow\|G_i) \simeq S'\|G \sqsubseteq S^\uparrow\|G$. This means that $\|_{i\in I}(S_i^\uparrow\|G_i) \sim S^\uparrow\|G$, namely, they are simulation equivalent (under (6.5) and (6.6)). However, they are not necessarily bisimilar to each other.

Remark 6.3 In [69], to solve the modular similarity control problem under the assumption (6.4), local supervisors were synthesized according to the synthesis method of [64]. However, the maximal permissiveness is not guaranteed, as opposed to Theorem 6.2.

Example 6.2 We consider the system $G \in \mathcal{A}(\Sigma)$ that consists of two subsystems $G_1 = (X_1, \Sigma_1, \alpha_1, X_{1,0}) \in \mathcal{A}(\Sigma_1)$ and $G_2 = (X_2, \Sigma_2, \alpha_2, X_{2,0}) \in \mathcal{A}(\Sigma_2)$ shown in Figures 6.2 (a), where for $i = 1, 2$, $\Sigma_i = \{a, b_i, c_i, d, e_i\}$ and $\Sigma_{i,uc} = \{a, e_i\}$. The local subspecifications $R_1 = (Q_1, \Sigma_1, \delta_1, Q_{1,0}) \in \mathcal{A}(\Sigma_1)$ and $R_2 = (Q_2, \Sigma_2, \delta_2, Q_{2,0}) \in \mathcal{A}(\Sigma_2)$ for G_1 and G_2, respectively, are given as shown in Figures 6.2 (b) and (c), respectively. Although (6.4) does not hold, it can be verified that the conditions (6.5) and (6.6) are satisfied. For $i = 1, 2$, the unique maximal fixpoints $W_{G_i,R_i}^\uparrow$ of F_{G_i,R_i} defined by (3.3) for G_i and R_i are computed as

$$\begin{aligned} F_{G_1,R_1}^1(X_1 \times Q_1) &= X_1 \times Q_1 - \{x_{1,0}\} \times \{q_{1,1}, q_{1,2}, q_{1,3}, q_{1,4}\} - \\ &\qquad \{x_{1,3}, x_{1,4}\} \times \{q_{1,0}, q_{1,1}, q_{1,2}, q_{1,4}\} \\ &= F_{G_1,R_1}^2(X_1 \times Q_1) \\ &= W_{G_1,R_1}^\uparrow \end{aligned}$$

and

$$
\begin{aligned}
F^1_{G_2,R_2}(X_2 \times Q_2) &= X_2 \times Q_2 - \{x_{2,0}\} \times \{q_{2,1}, q_{2,2}, q_{2,3}, q_{2,4}\} - \\
&\qquad \{x_{2,3}, x_{2,4}\} \times \{q_{2,0}, q_{2,1}, q_{2,2}\} \\
&= F^2_{G_2,R_2}(X_2 \times Q_2) \\
&= W^{\uparrow}_{G_2,R_2}.
\end{aligned}
$$

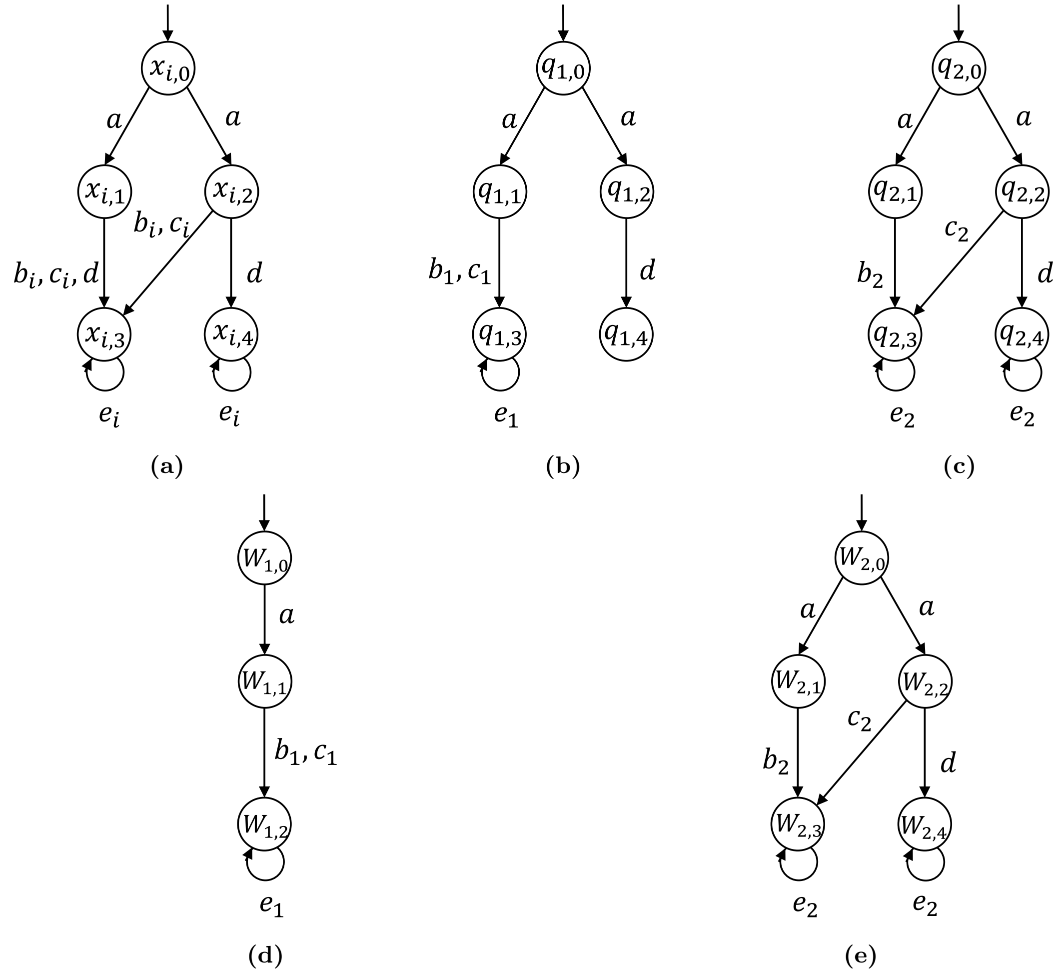


Figure 6.2: (a) Subsystems $G_i (i = 1, 2)$, (b) subspecification R_1, (c) subspecification R_2, (d) local supervisor $S_1^{\uparrow}$, and (e) local supervisor $S_1^{\uparrow}$.

We have $(x_{1,0}, q_{1,0}) \in W^{\uparrow}_{G_1,R_1}$ and $(x_{2,0}, q_{2,0}) \in W^{\uparrow}_{G_2,R_2}$ for the initial states $x_{1,0}$, $q_{1,0}$, $x_{2,0}$ and $q_{2,0}$ of G_1, R_1, G_2, and R_2, respectively. By Proposition 3.3, G_i is $\Sigma_{i,uc}$-simulated by R_i for $i = 1, 2$. In addition, by Theorem 6.1, the similarity control problem is solvable. Then, the local supervisors $S_1^{\uparrow}$ and $S_2^{\uparrow}$ defined by (3.10) can be synthesized, whose reachable parts are shown in Figures 6.2 (d) and (e), respectively, where

$$W_{1,0} = \{(x_{1,0}, q_{1,0})\}, \qquad W_{1,1} = \{(x_{1,1}, q_{1,1}), (x_{1,2}, q_{1,1})\},$$

$$W_{1,2} = \{(x_{1,3}, q_{1,3})\},$$

$$W_{2,0} = \{(x_{2,0}, q_{2,0})\}, \qquad W_{2,1} = \{(x_{2,1}, q_{2,1}), (x_{2,2}, q_{2,1})\},$$

$$W_{2,2} = \{(x_{2,1}, q_{2,2}), (x_{2,2}, q_{2,2})\}, \qquad W_{2,3} = \{(x_{2,3}, q_{2,3})\},$$

$$W_{2,4} = \{(x_{2,3}, q_{2,4}), (x_{2,4}, q_{2,4})\}.$$

Since (6.5) and (6.6) are satisfied, Proposition 6.2 ensures that $S_1^{\uparrow}$ and $S_2^{\uparrow}$ solve the modular similarity control problem, and the maximal permissiveness under the modular control scheme is guaranteed by Theorem 6.2.

Remark 6.4 As opposed to the modular similarity control problem of monolithic systems with modular specifications studied in Chapter 5, the results of this chapter cannot be straightforwardly extended to the case of partial observation, even though an observability version of the condition (6.4) defined as

$$\forall i, j \in I : i \neq j \Rightarrow \Sigma_{i,uo} \cap \Sigma_j = \emptyset$$

is imposed, where $\Sigma_{i,uo}$ is the unobservable event set of the ith subsystem. Here, the observable and unobservable event sets of the entire system are given as $\Sigma_o = \bigcup_{i \in I} \Sigma_{i,o}$ and $\Sigma_{uo} = \Sigma - \Sigma_o$, respectively.

6.4 Conclusion

In this chapter, we considered the modular similarity control problem for nondeterministic DESs that consist of multiple subsystems, each of which is assigned a local subspecification. To apply the modular control architecture, we made two assumptions that (i) the local controllability status of any event is the same in different subsystems and that (ii) any uncontrollable event that is locally feasible in at least one subsystem is globally feasible in the composite system. These assumptions are weaker than the one used in [69] which requires that any uncontrollable event is not shared by multiple subsystems. Under these assumptions, we showed that maximal permissiveness can be achieved by a set of local supervisors synthesized using the existing method of [65].

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this dissertation, we studied the maximally permissive similarity control problems for nondeterministic DESs in several settings. Given a system and a (safety) specification both modeled as NFAs, a similarity control problem is a feasible choice when a supervisor that enforces bisimilarity does not exist. Since the specification is given as the upper bound of behavior, synthesizing a maximal solution is an important subject in the similarity control problem.

For the monolithic similarity control problem, we first solved the similarity control problem subject to partial observation. We developed a synthesis method of a maximally permissive similarity-enforcing supervisor that works under partial observation. We also showed that a certain set of redundant states of the synthesized supervisor can be eliminated without reducing its permissiveness.

Then, we considered the nonblocking similarity control problem, which guarantees an important property of liveness of the controlled system. To impose nonblockingness, we proposed an algorithm that computes a nonblocking supervisor from a possibly blocking one by iteratively removing certain states. This algorithm is applicable for the similarity-enforcing supervisors synthesized for both fully and partially observed systems. We identified a class of supervisors as candidates for input supervisors of the proposed algorithm, so that a maximal solution to the nonblocking similarity control problem, if exists, is obtained as the output. In addition, we derived a necessary and sufficient condition for the existence of a solution to the nonblocking similarity control problem.

Next, we studied the modular similarity control problem, which deals with the main obstacle of exponential state space explosion arising in the supervisory control of large-scale systems. For the similarity control problem of the monolithic system and the modular specification that consists of multiple elementary specifications, we showed that the similarity control problem with modular specification is modularly solvable whenever it is monolithically solvable. We showed that, in order to achieve the maximal permissiveness with respect to the modular specification, it is sufficient to synthesize a set of elementary supervisors, each of which is maximally permissive with respect to the corresponding elementary specification.

Finally, we synthesized a maximal solution to the similarity control problem of the composite system, which consists of multiple local subsystems, each of which is assigned a nondeterministic local subspecification. A solution that is not necessarily maximal is obtained in the existing research [69], under the assumption that any uncontrollable event uniquely belongs to one subsystem. The problem is modularly solved under a new weaker assumption, while the maximal permissiveness is guaranteed.

By solving the above-mentioned similarity control problems, this study contributes to the construction of supervisory control theory of DESs. Since supervisory control theory constitutes the fundamental framework of formal-method-based control system design, the results of this thesis contribute to the construction of control system design theory based on formal method.

7.2 Directions for Future Work

The results of Chapters 3 and 4 are about enforcing safety (with respect to simulation relation) and nonblockingness properties to the given system and specification. However, there are many other properties in the context of supervisory control, for example, opacity, diagnosability, and detectability. A nondeterministic supervisor that enforces these properties could be a direction of future work.

In Section 3.5, we have introduced a technique to eliminate certain redundant states without reducing the permissiveness. Such a technique is useful in the modular control, which potentially reduces a large number of redundant branches (states). However, this technique cannot identify all redundant states. It is important to construct a maximally permissive supervisor with the fewest states.

For the modular similarity control problem, there are several directions of future work. (i) As discussed in Subsection 5.4.2, a nonconflicting property that guarantees nonblockingness of the system supervised by multiple supervisors is required for nondeterministic control. (ii) The local subspecifications generally span a couple of subsystems in practice, so a research that deals with such a situation is expected. (iii) The application of other control structures such as hierarchical structure in similarity control constitutes an important direction of future work.

References

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

[2] A. Hellgren, M. Fabian, and B. Lennartson, "Modular implementation of discrete event systems as sequential function charts applied to an assembly cell," in *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01) (Cat. No.01CH37204)*, pp. 453–458, 2001.

[3] R. J. Leduc and W. M. Wonham, "Discrete event systems modeling and control of a manufacturing testbed," in *Proceedings of 1995 Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 793–796, 1995.

[4] B. Brandin and F. Charbonnier, "The supervisory control of the automated manufacturing system of the aip," in *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, pp. 319–324, 1994.

[5] P. N. Pena, T. A. Costa, R. S. Silva, and R. H. Takahashi, "Control of flexible manufacturing systems under model uncertainty using supervisory control theory and evolutionary computation schedule synthesis," *Information Sciences*, vol. 329, pp. 491–502, 2016. Special issue on Discovery Science.

[6] H. Liao, Y. Wang, J. Stanley, S. Lafortune, S. Reveliotis, T. Kelly, and S. Mahlke, "Eliminating concurrency bugs in multithreaded software: A new approach based on discrete-event control," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2067–2082, 2013.

[7] A. Auer, J. Dingel, and K. Rudie, "Concurrency control generation for dynamic threads using discrete-event systems," *Science of Computer Programming*, vol. 82, pp. 22–43, 2014.

[8] F. Atampore, J. Dingel, and K. Rudie, "Automated service composition via supervisory control theory," in *Proceedings of 2016 13th International Workshop on Discrete Event Systems (WODES)*, pp. 28–35, 2016.

[9] M. Seidl, "Systematic controller design to drive high-load call centers," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 2, pp. 216–223, 2006.

[10] S. T. Forschelen, J. M. van de Mortel-Fronczak, R. Su, and J. E. Rooda, "Application of supervisory control theory to theme park vehicles," *Discrete Event Dynamic Systems*, vol. 22, no. 4, pp. 511–540, 2012.

[11] R. J. Theunissen, M. Petreczky, R. R. Schiffelers, D. A. van Beek, and J. E. Rooda, "Application of supervisory control synthesis to a patient support table of a magnetic resonance imaging scanner," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 20–32, 2014.

[12] B. C. Rawlings, B. Christenson, J. M. Wassick, and B. E. Ydstie, "Supervisor synthesis to satisfy safety and reachability requirements in chemical process control," *IFAC Proceedings Volumes*, vol. 47, no. 2, pp. 195–200, 2014.

[13] B. van der Sanden, M. Reniers, M. Geilen, T. Basten, J. Jacobs, J. Voeten, and R. Schiffelers, "Modular model-based supervisory controller design for wafer logistics in lithography machines," in *Proceedings of 2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pp. 416–425, 2015.

[14] T. Korssen, V. Dolk, J. van de Mortel-Fronczak, M. Reniers, and M. Heemels, "Systematic model-based design and implementation of supervisors for advanced driver assistance systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 533–544, 2018.

[15] F. F. Reijnen, M. A. Goorden, J. M. van de Mortel-Fronczak, and J. E. Rooda, "Modeling for supervisor synthesis – a lock-bridge combination case study," *Discrete Event Dynamic Systems*, vol. 30, no. 3, pp. 499–532, 2020.

[16] L. Moormann, M. A. Goorden, J. M. van de Mortel-Fronczak, W. J. Fokkink, P. Maessen, and J. E. Rooda, "Efficient validation of supervisory controllers using symmetry reduction," *IFAC-PapersOnLine*, vol. 53, no. 4, pp. 288–295, 2020.

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

[18] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE Transactions on Automatic Control*, vol. 33, no. 3, pp. 249–260, 1988.

[19] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Information Sciences*, vol. 44, no. 3, pp. 173–198, 1988.

[20] H. Cho and S. I. Marcus, "On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation," *Mathematics of Control, Signals and Systems*, vol. 2, no. 1, pp. 47–69, 1989.

[21] R. D. Brandt, V. Garg, R. Kumar, F. Lin, S. I. Marcus, and W. M. Wonham, "Formulas for calculating supremal controllable and normal sublanguages," *Systems & Control Letters*, vol. 15, no. 2, pp. 111–117, 1990.

[22] J. Komenda and J. H. van Schuppen, "Control of discrete-event systems with partial observations using coalgebra and coinduction," *Discrete Event Dynamic Systems*, vol. 15, no. 3, pp. 257–315, 2005.

[23] H. Cho and S. I. Marcus, "Supremal and maximal sublanguages arising in supervisor synthesis problems with partial observations," *Mathematical systems theory*, vol. 22, no. 3, pp. 177–211, 1989.

[24] M. Heymann and F. Lin, "On-line control of partially observed discrete event systems," *Discrete Event Dynamic Systems*, vol. 4, no. 3, pp. 221–236, 1994.

[25] N. B. Hadj-Alouane, S. Lafortune, and F. Lin, "Centralized and distributed algorithms for on-line synthesis of maximal control policies under partial observation," *Discrete Event Dynamic Systems*, vol. 6, no. 4, pp. 379–427, 1996.

[26] T. Ushio, "On-line control of discrete event systems with a maximally controllable and observable sublanguage," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E82-A, no. 9, pp. 1965–1970, 1999.

[27] S. Takai and T. Ushio, "Effective computation of an $L_m(G)$-closed, controllable, and observable sublanguage arising in supervisory control," *Systems & Control Letters*, vol. 49, no. 3, pp. 191–200, 2003.

[28] K. Cai, R. Zhang, and W. M. Wonham, "Relative observability of discrete-event systems and its supremal sublanguages," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 659–670, 2015.

[29] X. Yin and S. Lafortune, "Synthesis of maximally permissive supervisors for partially-observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1239–1254, 2016.

[30] K. Inan, "Nondeterministic supervision under partial observations," in *11th International Conference on Analysis and Optimization of Systems Discrete Event Systems*, pp. 39–48, 1994.

[31] R. Kumar, S. Jiang, C. Zhou, and W. Qiu, "Polynomial synthesis of supervisor for partially observed discrete-event systems by allowing nondeterminism in control," *IEEE Transactions on Automatic Control*, vol. 50, no. 4, pp. 463–475, 2005.

[32] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems.* Springer, 3rd ed., 2021.

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

[34] Y. Willner and M. Heymann, "Supervisory control of concurrent discrete-event systems," *International Journal of Control*, vol. 54, no. 5, pp. 1143–1169, 1991.

[35] S. Jiang and R. Kumar, "Decentralized control of discrete event systems with specializations to local control and concurrent systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 5, pp. 653–660, 2000.

[36] M. H. de Queiroz and J. E. R. Cury, "Modular supervisory control of large scale discrete event systems," in *Discrete Event Systems: Analysis and Control*, pp. 103–110, 2000.

[37] J. Komenda, J. H. van Schuppen, B. Gaudin, and H. Marchand, "Supervisory control of modular systems with global specification languages," *Automatica*, vol. 44, no. 4, pp. 1127–1134, 2008.

[38] W. M. Wonham, K. Cai, and K. Rudie, "Supervisory control of discrete-event systems: A brief history," *Annual Reviews in Control*, vol. 45, pp. 250–256, 2018.

[39] K. Wong, J. Thistle, R. Malhamé, and H.-H. Hoang, "Supervisory control of distributed systems: Conflict resolution," *Discrete Event Dynamic Systems*, vol. 10, no. 1, pp. 131–186, 2000.

[40] Y.-L. Chen, S. Lafortune, and F. Lin, "Design of nonblocking modular supervisors using event priority functions," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 432–452, 2000.

[41] R. C. Hill, D. M. Tilbury, and S. Lafortune, "Modular supervisory control with equivalence-based abstraction and covering-based conflict resolution," *Discrete Event Dynamic Systems*, vol. 20, no. 1, pp. 139–185, 2010.

[42] M. A. Shayman and R. Kumar, "Supervisory control of nondeterministic systems with driven events via prioritized synchronization and trajectory models," *SIAM Journal on Control and Optimization*, vol. 33, no. 2, pp. 469–497, 1995.

[43] R. Kumar and M. A. Shayman, "Nonblocking supervisory control of nondeterministic systems via prioritized synchronization," *IEEE Transactions on Automatic Control*, vol. 41, no. 8, pp. 1160–1175, 1996.

[44] S. Jiang and R. Kumar, "Supervisory control of nondeterministic discrete-event systems with driven events via masked prioritized synchronization," *IEEE Transactions on Automatic Control*, vol. 47, no. 9, pp. 1438–1449, 2002.

[45] A. Overkamp, "Supervisory control using failure semantics and partial specifications," *IEEE Transactions on Automatic Control*, vol. 42, no. 4, pp. 498–510, 1997.

[46] M. Heymann and F. Lin, "Discrete-event control of nondeterministic systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 3–17, 1998.

[47] A. Arnold, A. Vincent, and I. Walukiewicz, "Games for synthesis of controllers with partial observation," *Theoretical Computer Science*, vol. 303, no. 1, pp. 7–34, 2003.

[48] S. Pinchinat and J.-B. Raclet, "Supervisory control problems for nondeterministic discrete-event systems: A logical approach," in *IFAC Proceedings of 16th IFAC World Congress*, pp. 67–72, 2005.

[49] S. Jiang and R. Kumar, "Supervisory control of discrete event systems with CTL* temporal logic specifications," *SIAM Journal on Control and Optimization*, vol. 44, no. 6, pp. 2079–2103, 2006.

[50] A. C. van Hulst, M. A. Reniers, and W. J. Fokkink, "Maximally permissive controlled system synthesis for non-determinism and modal logic," *Discrete Event Dynamic Systems*, vol. 27, no. 1, pp. 109–142, 2017.

[51] R. Milner, *A Calculus of Communicating Systems.* Springer-Verlag, 1980.

[52] R. Milner, *Communication and Concurrency.* Prentice-Hall, 1989.

[53] C. Zhou, R. Kumar, and S. Jiang, "Control of nondeterministic discrete-event systems for bisimulation equivalence," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 754–765, 2006.

[54] C. Zhou and R. Kumar, "A small model theorem for bisimilarity control under partial observation," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 93–97, 2007.

[55] S. Takai, "Bisimilarity enforcing supervisory control of nondeterministic discrete event systems with nondeterministic specifications," *Automatica*, vol. 108, p. 108470, 2019.

[56] H. Farhat, "Control of nondeterministic systems for bisimulation equivalence under partial information," *IEEE Transactions on Automatic Control*, vol. 65, no. 12, pp. 5437–5443, 2020.

[57] Y. Sun and H. Lin, "Bisimilarity enforcing supervisory control of nondeterministic discrete event systems," in *2012 American Control Conference (ACC)*, pp. 6102–6107, 2012.

[58] K. Kimura and S. Takai, "Bisimilarity control of nondeterministic discrete event systems under event and state observations," *IEICE Transactions on Information and Systems*, vol. E97-D, no. 5, pp. 1140–1148, 2014.

[59] C. Zhou and R. Kumar, "Bisimilarity enforcement for discrete event systems using deterministic control," *IEEE Transactions on Automatic Control*, vol. 56, no. 12, pp. 2986–2991, 2011.

[60] Y. Sun, H. Lin, and B. M. Chen, "Bisimilarity enforcing supervisory control for deterministic specifications," *Automatica*, vol. 50, no. 1, pp. 287–290, 2014.

[61] K. Shimatani and S. Takai, "Deterministic supervisors for bisimilarity control of partially observed nondeterministic discrete event systems with deterministic specifications," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E104-A, no. 2, pp. 438–446, 2021.

[62] C. Zhou and R. Kumar, "Control of nondeterministic discrete event systems for simulation equivalence," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pp. 340–349, 2007.

[63] K. Kimura and S. Takai, "Maximally permissive similarity enforcing supervisors for nondeterministic discrete event systems under event and state observations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E97-A, no. 7, pp. 1500–1507, 2014.

[64] N. Kushi and S. Takai, "Synthesis of similarity enforcing supervisors for nondeterministic discrete event systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1457–1464, 2018.

[65] S. Takai, "Synthesis of maximally permissive supervisors for nondeterministic discrete event systems with nondeterministic specifications," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 3197–3204, 2021.

[66] H. Yamada and S. Takai, "Nonblocking similarity control of nondeterministic discrete event systems under event and state observations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E101-A, no. 2, pp. 328–337, 2018.

[67] J. Li and S. Takai, "Maximally permissive nonblocking supervisors for similarity control of nondeterministic discrete event systems under event and state observations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E102-A, no. 2, pp. 399–403, 2019.

[68] M. Hoshino and S. Takai, "Decentralized similarity control of composite nondeterministic discrete event systems with local specifications," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100-A, no. 2, pp. 395–405, 2017.

[69] I. Okubo and S. Takai, "Decentralized supervisor synthesis for composite nondeterministic discrete event systems with local specifications," in *Proceedings of the SICE Annual Conference 2018*, pp. 1793–1796, 2018.

[70] S. Takai, "Maximally permissive supervisory control of nondeterministic discrete event systems with nondeterministic specifications," in *Proceedings of 57th IEEE Conference on Decision and Control*, pp. 3975–3980, 2018.

[71] A. Tarski, "A lattice-theoretical fixpoint theorem and its applications," *Pacific Journal of Mathematics*, vol. 5, no. 2, pp. 285–309, 1955.

Publications

Journals

1. J. Li and S. Takai, "Maximally permissive nonblocking supervisors for similarity control of nondeterministic discrete event systems under event and state observations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E102-A, no. 2, pp. 399–403, 2019.

2. J. Li and S. Takai, "Synthesis of maximally permissive supervisors for similarity control of partially observed nondeterministic discrete event systems," *Automatica*, vol. 135, article no. 109978, pp. 1–13, 2022.

3. J. Li and S. Takai, "Modular similarity control of nondeterministic discrete event systems with modular specifications," *IEEE Control System Letters*, vol. 6, pp. 1358–1363, 2022.

4. J. Li and S. Takai, "Maximally permissive modular similarity control of composite nondeterministic discrete event systems," *IEEE Control System Letters*, vol. 6, pp. 2305–2310, 2022.

5. J. Li and S. Takai, "Maximally permissive supervisors for nonblocking similarity control of nondeterministic discrete event systems," *IEEE Transactions on Automatic Control*, accepted for publication.

International Conferences

1. J. Li and S. Takai, "Maximally permissive similarity-enforcing supervisors for nondeterministic discrete event systems under partial observation," in *Proceedings of the 58th IEEE Conference on Decision and Control*, Nice, France, pp. 1037–1042, 2019.

2. J. Li and S. Takai, "Maximally permissive nonblocking similarity control of nondeterministic discrete event systems," in *Proceedings of the 59th IEEE Conference on Decision and Control*, Jeju Island, Republic of Korea, pp. 73–78, 2020.

Domestic Conferences

1. J. Li and S. Takai, "Maximally permissive nonblocking supervisors for similarity control of nondeterministic discrete event systems," *IEICE Technical Report*, vol. 117, no. 506, MSS2017-91, Toyonaka, Osaka, pp. 73-76, 2018.

2. J. Li and S. Takai, “Similarity Control of Nondeterministic Discrete Event Systems under Partial Observation,” *IEICE Technical Report*, vol. 118, no. 384, MSS2018-57, Naha, Okinawa, pp. 19-23, 2019.

3. J. Li and S. Takai, “Maximally permissive similarity-enforcing supervisors for nondeterministic discrete event systems with modular specifications,” In *Proceedings of the 63rd Japan Joint Automatic Control Conference*, pp. 81–84, 2020.

Maximally Permissive Similarity Control of Nondeterministic Discrete Event Systems

２０２２年６月

李　京倫