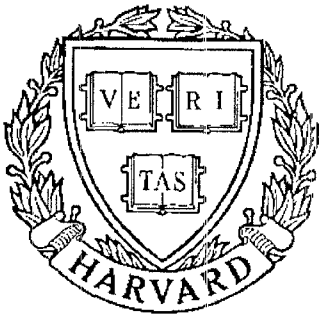
TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER



Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Supervisory Control of Nondeterministic Systems with Driven Events via Prioritized Synchronization and Trajectory Models

by M.A. Shayman and R. Kumar

TR 92-113

Supervisory Control of Nondeterministic Systems with Driven Events via Prioritized Synchronization and Trajectory Models [1]

Mark A. Shayman
Department of Electrical Engineering and
Systems Research Center
University of Maryland
College Park, MD 20742
Email: shayman@src.umd.edu

Ratnesh Kumar
Department of Electrical Engineering
University of Kentucky
Lexington, KY 40506-0046
Email: kumar@engr.uky.edu

October 21, 1992

[1]This research was supported in part by Center for Robotics and Manufacturing, University of Kentucky, in part by by the National Science Foundation under the Engineering Research Centers Program Grant CDR-8803012, the Minta Martin Fund for Aeronautical Research, and the General Research Board at the University of Maryland.

Abstract

We study the supervisory control of nondeterministic discrete event dynamical systems (DEDS's) with driven events in the setting of *prioritized synchronization* and *trajectory models* introduced by Heymann. Prioritized synchronization captures the notions of controllable, uncontrollable, and driven events in a natural way, and we use it for constructing supervisory controllers. The trajectory model is used for characterizing the behavior of nondeterministic DEDS's since it is a sufficiently detailed model (in contrast to the less detailed language or failures models), and serves as a *language congruence* with respect to the operation of prioritized synchronization. We obtain results concerning controllability and observability in this general setting.

Keywords: discrete event systems, supervisory control, nondeterministic automata, driven events, prioritized synchronization, trajectory models

AMS (MOS) subject classifications: 68Q75, 93B25, 93C83

1 Introduction

Supervisory control of discrete event dynamical systems (DEDS's) was introduced by Ramadge and Wonham [17]. In this approach, the behavior of a DEDS, called the plant, is described by its language, the collection of all possible sequences of events (traces) that it can generate. The task is to design a controller, called a supervisor, which, based on the observation of the sequence of events, disables some of the *controllable* events so that the language generated by the controlled plant either equals a prespecified desired language, called a target language, or remains confined to a prespecified range of languages. Various extensions of this basic problem such as control under partial observation, decentralized and modular control, hierarchical control, and optimal control have also been studied. Refer to [18] and references therein for an overview of research in this area.

Most of the research on supervisory control of DEDS's assumes that the plant can be modeled as a *deterministic* system [8]. In other words, given a state of the system, and an event that occurs in that state, the state reached after the occurrence of the event is uniquely known. Such an assumption is not satisfied whenever unmodeled dynamics, partial observation, or inherent nondeterminism is present. It is not possible to avoid these complications in a realistic setting. Hence the assumption of a deterministic plant is quite strong. In this paper, we relax this assumption, and consider the control of a *nondeterministic* plant [8, 16, 7, 9, 5], which offers a more reasonable and realistic model of DEDS's.

A *modeling framework* m over a finite event set Σ is an equivalence relation on all DEDS's representable as state machines, with arbitrary state space, having ϵ-transitions and event set Σ. We identify m with the projection π_m which maps each state machine $\mathcal{P}$ to its equivalence class or *model* $\pi_m(\mathcal{P})$. If the equivalence class of $\mathcal{P}$ is uniquely characterized by an attribute which is common to its members, we will freely identify $\pi_m(\mathcal{P})$ with this attribute.

We say that a modeling framework π_m is *more detailed* than another modeling framework π_n if the equivalence relation π_m *refines* the equivalence relation π_n. Obviously, it is desirable to use the least detailed modeling framework which is sufficient for the design task at hand. A complex system is generally synthesized by combining simpler systems using various types of interconnections. Since specifications for the logical behavior of a DEDS are typically given in terms of the *language* of the system, a basic requirement is that the modeling framework should contain sufficient detail so that if the models for each subsystem are known, then the language of the interconnected system is uniquely determined. A modeling framework with such a property for a given class of admissible interconnections is referred to as a *language congruence* [5].

The language modeling framework associates to a system its language, the collection of all possible finite traces which are executable. Thus, the language model of a system is a subset of Σ^*, the set of all finite strings of events in Σ including ϵ, the zero length string. For deterministic systems and deterministic operators such as *strict synchronous composition* (SSC), the language modeling framework is a language congruence. If operators which introduce nondeterminism (e.g., internal choice, event internalization) are admissible, then the language modeling framework is no longer a language congruence and a more detailed

modeling framework such as the *failures model* introduced by Hoare [7] must be used in order to have a language congruence. The failures model consists of the set of all *failures* of the system–pairs (s, Σ') where s is a trace and $\Sigma' \subseteq \Sigma$ is a refusal set with the property that if the environment restricts the possible events to Σ', the system can deadlock following execution of s. Thus a failures model is a subset of $\Sigma^* \times 2^{\Sigma}$.[1]

In the work of Kumar, Garg and Marcus [11], control design is accomplished by constructing a supervisor which operates in strict synchronization with the plant. In the work of Balemi et al. [1], the set of events Σ is partitioned into two disjoint subsets–*commands* which are generated by the supervisor and sent to the plant, and *responses* which are generated by the plant and sent to the supervisor. It is required that the plant and supervisor be *mutually receptive*, which means that the plant executes every command generated by the supervisor and the supervisor executes every response generated by the plant. Thus, this design also requires that every event be executed synchronously.

There are several reasons to consider control designs which do not require complete synchronization between the plant and supervisor. Uncontrollable events are generated spontaneously by the plant and the supervisor is not permitted to interfere with their execution. Consequently, there is no a priori reason to assume that the supervisor needs to "track" every such event by undergoing a transition synchronously with the plant. Also, certain uncontrollable events in the plant may not be sensed and hence are invisible to the supervisor. It is unrealistic to require the supervisor to execute such events synchronously.

In many applications, it is not realistic to expect (or require) the plant to respond synchronously to every event generated by the supervisor. (Such events are referred to as *forceable* [4], *driven* [5] or *command* [1] events in the literature.) By permitting the supervisor to place commands which are not executed by the plant, nondeterminism in the plant may be resolved and performance improved. For example, not every piece of equipment in a factory will trigger an alarm upon breakdown. Breakdown may only be discovered when an action is requested by the supervisor and not executed by the plant. Thus, the unsensed state of the plant is determined by a synchronization failure.

Another motivation for relaxing the requirement of strict synchronization comes from systems in which a single supervisor controls more than one plant. For example, in a walking machine, there could be separate modules (viewed here as plants) which perform motion control and vision control respectively. At a higher-level, there could be a single supervisor which controls and coordinates the two modules. Some of the commands issued by the supervisor may apply to both the modules, while others may be relevant to only one of them and should be ignored by the other.[2]

Heymann [5] has proposed a type of interconnection, called *prioritized synchronous composition* (PSC), which relaxes the synchronization requirements on the plant and supervisor.

[1] For simplicity, we ignore the possibility of divergence.

[2] This problem can sometimes be addressed by assigning different event alphabets to each process and only requiring synchronization for events in the intersection of the alphabets. However, this is inadequate for applications in which synchronization requirements are naturally state-dependent. See [10] for a different approach using state-dependent (or trace-dependent) alphabets.

A priority set of events is associated with each system. An event is executable in the PSC of two systems only if it is executable in the system(s) whose priority set(s) contain that event. It is shown in [5, Example 7] that two systems with the same failures model may yield different languages when composed in prioritized synchrony with a fixed system. Thus, if PSC is included as an admissible interconnection operator, a more detailed modeling framework than the failures model is required to serve as a language congruence. One such modeling framework, called the *trajectory model*, is proposed by Heymann [5] and Heymann-Meyer [6]. The trajectory model of a system consists of the set of all *trajectories*–finite sequences of the type $\Sigma_0(\sigma_1, \Sigma_1)\ldots(\sigma_k, \Sigma_k)$, where $\sigma_1 \ldots \sigma_k$ is the trace executed by the system, while $\Sigma_j \subseteq \Sigma$ $(j = 0, \ldots, k)$ is a refusal set, a set of events which can result in deadlock if presented to the system by the environment at the indicated point in the trajectory. Thus, a trajectory model is a subset of $2^\Sigma \times (\Sigma \times 2^\Sigma)^*$ and refines the failures model by including the intermediate refusal sets.

Although we use the trajectory model for describing the behavior of a nondeterministic plant, it is assumed that the desired specification is given only in terms of a language model (as in [17]), and not in terms of a trajectory model. This is a reasonable assumption, for in most applications, we are only interested in the sequences of events that a system can execute, and not in the events that the system may "refuse" to execute after execution of a certain event in a certain event sequence. Hence we address the following supervisory control problem:

> Given (i) a partition $\Sigma = \Sigma_c \cup \Sigma_u \cup \Sigma_d$ of the event set into subsets of controllable, uncontrollable and driven events, (ii) a nondeterministic plant with trajectory model $P \subseteq 2^\Sigma \times (\Sigma \times 2^\Sigma)^*$, whose priority set is $A = \Sigma_c \cup \Sigma_u$, (iii) a target language $K \subseteq \Sigma^*$; design a supervisor–another trajectory model, denoted $S \subseteq 2^\Sigma \times (\Sigma \times 2^\Sigma)^*$–whose priority set is $B = \Sigma_c \cup \Sigma_d$, such that the language of the PSC of P and S equals K.

The interconnection of the plant and the supervisor by PSC results in disabling of some of the controllable events and forcing of some of the driven events, while never preventing any of the uncontrollable events from occurring in the plant. Thus we investigate the supervisory control of DEDS's in the general setting of trajectory models and PSC, as opposed to language models and SSC studied by Kumar, Garg and Marcus [11].

We obtain a necessary and sufficient condition for the existence of a supervisor for the general problem with driven events, and also provide a technique for synthesizing a supervisor. For ease of implementation, we design supervisors which are deterministic. We also address the control problem when some of the uncontrollable events are not observed by the supervisor. While the primary goal of this paper is to obtain necessary and sufficient conditions for the control of nondeterministic systems with driven events, a secondary goal is to provide a rigorous mathematical foundation for the theory of trajectory models and PSC, and to resolve certain ambiguities concerning their properties which exist in the literature.

The organization of this paper is as follows: In Section 2, the trajectory model of a nondeterministic state machine (NSM) with ϵ-moves is defined and its properties derived

from those of NSM's. An algorithm to construct a canonical NSM from a given trajectory model is presented and its correctness proven. In Section 3, the PSC of NSM's is defined and it is shown that this induces a PSC operation on trajectory models. It is also proven that the trajectory modeling framework is a language congruence relative to PSC. Properties of the PSC of trajectory models are described in Section 4, and the technique of *augmentation* is introduced. In Section 5, the supervisory control problem with driven events under both complete and partial observation is solved.

2 Trajectory Model

A plant, or a DEDS to be controlled, is modeled as an NSM with ϵ-moves. Letting $\mathcal{P}$ denote an NSM, it is defined to be the four tuple [8]:

$$\mathcal{P} := (X_{\mathcal{P}}, \Sigma, \delta_{\mathcal{P}}, x^0_{\mathcal{P}}),$$

where $X_{\mathcal{P}}$ denotes the state space of $\mathcal{P}$, Σ denotes the event set of $\mathcal{P}$, $\delta_{\mathcal{P}} : X_{\mathcal{P}} \times \Sigma \cup \{\epsilon\} \rightarrow 2^{X_{\mathcal{P}}}$ denotes the nondeterministic[3] transition function of $\mathcal{P}$, and $x^0_{\mathcal{P}} \in X_{\mathcal{P}}$ denotes the initial state of $\mathcal{P}$. A triple $(x_1, \sigma, x_2) \in X_{\mathcal{P}} \times (\Sigma \cup \{\epsilon\}) \times X_{\mathcal{P}}$ is called a transition in $\mathcal{P}$ if $x_2 \in \delta_{\mathcal{P}}(x_1, \sigma)$. A transition (x_1, ϵ, x_2) is referred to as a *silent* transition. We assume that the plant cannot undergo an unbounded sequence of silent transitions.

2.1 Language or Trace Model

As mentioned in Section 1, although trajectory models are used for describing the behaviors of nondeterministic systems, language or trace models are used for describing the desired or target specifications. Hence in this subsection we define the language model of the plant.

We first define the ϵ-closure of a state, which is the set of states reached by executing a finite sequence of "silent" transitions.

Definition 1 The *ϵ-closure* map, $\epsilon^*_{\mathcal{P}} : X_{\mathcal{P}} \rightarrow 2^{X_{\mathcal{P}}}$, is recursively defined to be:

$\forall x \in X_{\mathcal{P}}$:

- $x \in \epsilon^*_{\mathcal{P}}(x)$,
- $x' \in \epsilon^*_{\mathcal{P}}(x) \Rightarrow \delta_{\mathcal{P}}(x', \epsilon) \subseteq \epsilon^*_{\mathcal{P}}(x)$.

Using the definition of ϵ-closure, we extend the definition of the transition function from events to traces as follows:

[3]The transition function $\delta_{\mathcal{P}}$ is deterministic if and only if it is of the type, $\delta_{\mathcal{P}} : X_{\mathcal{P}} \times \Sigma \rightarrow X_{\mathcal{P}}$, in which (i) there are no transitions labeled ϵ, and (ii) given a state and an event, either a unique state is reached upon execution of that event in that state, or that event is undefined in that state.

Definition 2 The extension of the transition function to traces, denoted $\delta_{\mathcal{P}}^* : X_{\mathcal{P}} \times \Sigma^* \rightarrow 2^{X_{\mathcal{P}}}$, is defined inductively on the length of the traces as:

$\forall x \in X_{\mathcal{P}}$:

- $\delta_{\mathcal{P}}^*(x, \epsilon) = \epsilon_{\mathcal{P}}^*(x)$,
- $\forall s \in \Sigma^*, \sigma \in \Sigma : \delta_{\mathcal{P}}^*(x, s\sigma) = \epsilon_{\mathcal{P}}^*(\delta_{\mathcal{P}}(\delta_{\mathcal{P}}^*(x, s), \sigma))$,

where in the last equality, the transition map is extended to $\delta_{\mathcal{P}} : 2^{X_{\mathcal{P}}} \times \Sigma \cup \{\epsilon\} \rightarrow 2^{X_{\mathcal{P}}}$, and the ϵ-closure map is extended to $\epsilon_{\mathcal{P}}^* : 2^{X_{\mathcal{P}}} \rightarrow 2^{X_{\mathcal{P}}}$ in the natural way.

The set of states reached by executing a string $s \in \Sigma^*$ from a state $x \in X_{\mathcal{P}}$ is given by the set $\delta_{\mathcal{P}}^*(x, s)$. It is clear that if $\mathcal{P}$ is deterministic, then the extension of the transition function to traces is also a deterministic partial map $\delta_{\mathcal{P}}^* : X_{\mathcal{P}} \times \Sigma^* \rightarrow X_{\mathcal{P}}$. (It is a partial map since it is generally defined only on a subset of $X_{\mathcal{P}} \times \Sigma^*$.)

The preceding definition can be used to obtain the language or trace model for the plant $\mathcal{P}$, denoted $\pi_l(\mathcal{P}) \subseteq \Sigma^*$, as follows:

$$\pi_l(\mathcal{P}) := \{s \in \Sigma^* \mid \delta_{\mathcal{P}}^*(x_{\mathcal{P}}^0, s) \neq \emptyset\}.$$

Note that the following properties, L1 and L2, are satisfied by the language model $\pi_l(\mathcal{P})$ of the plant:

L1 (nonemptiness): $\epsilon \in \pi_l(\mathcal{P}) \Rightarrow \pi_l(\mathcal{P}) \neq \emptyset$,

L2 (prefix closure): $s \in \pi_l(\mathcal{P}), t \in \Sigma^*$ such that $t < s \Rightarrow t \in \pi_l(\mathcal{P})$,

where the notation $t < s$ is used to denote that t is a *proper prefix* of s. Given a language model satisfying properties L1 and L2, standard algorithms are available for constructing a canonical state machine having the same language model. If K is a nonempty subset of Σ^*, $\overline{K}$ denotes the prefix-closure of K–i.e.,

$$\overline{K} = \{t \in \Sigma^* | \ \exists s \in K \text{ such that } t \leq s\}.$$

2.2 Trajectory Model of a Nondeterministic State Machine

As discussed in Section 1, language models are not adequate for characterizing the behavior of nondeterministic systems. Hence, we next define the trajectory model for the plant $\mathcal{P}$. We first need to define the refusal map, and extend the transition function from events to trajectories.

Definition 3 The *refusal* map, $\Re_{\mathcal{P}} : X_{\mathcal{P}} \rightarrow 2^{\Sigma}$, is defined as:

$$\forall x \in X_{\mathcal{P}} : \Re_{\mathcal{P}}(x) = \Sigma - \bigcup_{x' \in \epsilon_{\mathcal{P}}^*(x)} \Sigma_{\mathcal{P}}(x'),$$

where $\Sigma_{\mathcal{P}}(x') = \{\sigma \in \Sigma \mid \delta_{\mathcal{P}}(x', \sigma) \neq \emptyset\}$.

Thus the refusal map defines, at each state, a set of events such that the system "refuses" to execute any of the events belonging to that set at that state. An event $\sigma \in \Sigma$ belongs to the refusal set of a state $x \in X_{\mathcal{P}}$ if and only if it is undefined at each state belonging to the ϵ-closure of x.

Next we define the extension of the transition function from events to trajectories.

Definition 4 The extension of the transition function to trajectories, denoted $\delta_{\mathcal{P}}^{\pi_t} : X_{\mathcal{P}} \times (2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*) \rightarrow 2^{X_{\mathcal{P}}}$, is defined inductively on the length of the trajectories as:

$\forall x \in X_{\mathcal{P}}$:

- $\forall \Sigma' \subseteq \Sigma : \delta_{\mathcal{P}}^{\pi_t}(x, \Sigma') = \{x' \in \epsilon_{\mathcal{P}}^*(x) \mid \Sigma' \subseteq \Re_{\mathcal{P}}(x')\}$,
- $\forall e \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*, \sigma \in \Sigma, \Sigma' \subseteq \Sigma$:
  $\delta_{\mathcal{P}}^{\pi_t}(x, e(\sigma, \Sigma')) = \{x' \in \epsilon_{\mathcal{P}}^*(\delta_{\mathcal{P}}(\delta_{\mathcal{P}}^{\pi_t}(x, e), \sigma)) \mid \Sigma' \subseteq \Re_{\mathcal{P}}(x')\}$.

A state $x' \in X_{\mathcal{P}}$ is reached by executing a "zero-length" trajectory $\Sigma' \subseteq \Sigma$ from a state $x \in X_{\mathcal{P}}$ if (i) x' belongs to the epsilon-closure of x, and (ii) the refusal set of x' contains Σ'. A state $x' \in X_{\mathcal{P}}$ is reached by executing a trajectory $e(\sigma, \Sigma') \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$ from a state $x \in X_{\mathcal{P}}$ if (i) x' belongs to the epsilon-closure of a state reached by executing the event σ from a state reached after executing the trajectory e from x, and (ii) the refusal set of x' contains Σ'. It is clear that if $\mathcal{P}$ is deterministic, then the extension of the transition function to trajectories is also a deterministic partial map $\delta_{\mathcal{P}}^{\pi_t} : X_{\mathcal{P}} \times (2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*) \rightarrow X_{\mathcal{P}}$.

A trajectory $e \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$ can be written as $e = \Sigma_0(e)(\sigma_1(e), \Sigma_1(e)) \ldots (\sigma_n(e), \Sigma_n(e))$ for some $n \in \mathcal{N}$, where $\Sigma_i(e) \subseteq \Sigma$ for each $0 \leq i \leq n$, and $\sigma_j(e) \in \Sigma$ for each $1 \leq j \leq n$. We call n the length of e, and denote it as $|e| = n$. $\Sigma_i(e)$ is called the *ith* refusal set of e, and $\sigma_j(e)$ the *jth* event of e. For each $0 \leq i \leq |e|$, we use e^i to denote the prefix of length i of e, i.e., $e^i := \Sigma_0(e) \ldots (\sigma_i(e), \Sigma_i(e))$. Given two (distinct) trajectories $e, f \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$, we say that f is (strictly) *dominated* by e, denoted $f \sqsubset e$, (equivalently, e *dominates* f, denoted $e \sqsupset f$), if $|f| = |e| := n$, $\sigma_j(f) = \sigma_j(e)$ for each $1 \leq j \leq n$, and $\Sigma_i(f) \subseteq \Sigma_i(e)$, for each $0 \leq i \leq n$.

Based on the above extension of the transition function from events to trajectories, we define the trajectory model of the plant $\mathcal{P}$, which we denote as $\pi_t(\mathcal{P})$:

$$\pi_t(\mathcal{P}) := \{e \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^* \mid \delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e) \neq \emptyset\}.$$

We refer to the elements of this set as the *trajectories of* $\mathcal{P}$. A trajectory $e \in \pi_t(\mathcal{P})$ is said to be a *dominant* trajectory of $\mathcal{P}$ if there exists no other trajectory $f \in \pi_t(\mathcal{P})$ such that $e \sqsubset f$. Furthermore, e is said to be a *maximal* trajectory if it is dominant, and there exists no other trajectory $g \in \pi_t(\mathcal{P})$ such that $e < g$.

Example 1 Consider a system $\mathcal{P}$ that deadlocks, i.e., cannot execute any transition, at its initial state. Then $\pi_t(\mathcal{P}) = \{\Sigma' \mid \Sigma' \subseteq \Sigma\}$. I.e., the trajectory model of $\mathcal{P}$ consists of the zero length trajectories Σ'. We use $\Delta_{\Sigma} := \{\Sigma' \mid \Sigma' \subseteq \Sigma\}$, to denote the trajectory model of the *deadlock* system. Given $\sigma \in \Sigma$ and a trajectory model $T \subseteq 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$, we use $\sigma \rightarrow T$ to

denote the system that first executes the event σ and then follows a trajectory in T. In other words, $\sigma \rightarrow T := \{\Sigma'(\sigma, e) \mid \Sigma' \subseteq \Sigma - \{\sigma\}, e \in T\}$. $\sigma \rightarrow T$ is called the *σ-prefix* operation on the trajectory model T. Given trajectory models $T_1, T_2 \subseteq 2^\Sigma \times (\Sigma \times 2^\Sigma)^*$, we use $T_1 \oplus T_2$ to denote the system that nondeterministically chooses to execute trajectories either in T_1 or in T_2. $T_1 \oplus T_2$ is called the *internal choice* between T_1 and T_2, and $T_1 \oplus T_2 := T_1 \cup T_2$. If $\sigma_1, \sigma_2 \in \Sigma$ with $\sigma_1 \neq \sigma_2$, the *external choice* between the trajectory models $\sigma_1 \rightarrow T_1$ and $\sigma_2 \rightarrow T_2$ is defined to be the trajectory model

$$(\sigma_1 \rightarrow T_1) + (\sigma_2 \rightarrow T_2) := \{e \in (\sigma_1 \rightarrow T_1) \cup (\sigma_2 \rightarrow T_2) | \ e^0 \in (\sigma_1 \rightarrow T_1) \cap (\sigma_2 \rightarrow T_2)\}.$$

This is a process which initially makes a deterministic choice between σ_1 and σ_2. If σ_i is executed, then the remainder of the trajectory is in T_i.

Remark 1 There is a subtle but important difference in the meaning of the refusal sets in a trajectory model as opposed to those in an NSM. In the NSM, $\Re_\mathcal{P}(x)$ represents events that *must* be refused at the state x if offered by the environment. In contrast, the refusal set $\Sigma_i(e)$ in the trajectory e represents a set of events which *can* be refused if offered by the environment following execution of the previous fragment of the trajectory. The reason for this is that the *trajectory fragment* does not uniquely determine the state of the NSM due to nondeterminism, unless e is a maximal trajectory. (Refer to Algorithm 1.)

It follows from the definition of the trajectory model $\pi_t(\mathcal{P})$ that it satisfies the following five properties, denoted T1, T2, T3, T4, and T5:

Proposition 1 The trajectory model $\pi_t(\mathcal{P})$ of an NSM $\mathcal{P}$ satisfies the following properties:

T1 (nonemptiness): $\emptyset \in \pi_t(\mathcal{P}) \Rightarrow \pi_t(\mathcal{P}) \neq \emptyset$,

T2 (prefix closure): $\forall e \in \pi_t(\mathcal{P}), f \in 2^\Sigma \times (\Sigma \times 2^\Sigma)^* : f < e \Rightarrow f \in \pi_t(\mathcal{P})$,

T3 (dominance closure): $\forall e \in \pi_t(\mathcal{P}), f \in 2^\Sigma \times (\Sigma \times 2^\Sigma)^* : f \sqsubset e \Rightarrow f \in \pi_t(\mathcal{P})$,

T4 (refusal of infeasible): $\forall e \in 2^\Sigma \times (\Sigma \times 2^\Sigma)^*, 0 \leq i \leq |e|, \sigma \in \Sigma :$
$e^i(\sigma, \emptyset) \notin \pi_t(\mathcal{P}) \Rightarrow e^{i-1}(\sigma_i(e), \Sigma_i(e) \cup \{\sigma\}) \ldots (\sigma_{|e|}, \Sigma_{|e|}) \in \pi_t(\mathcal{P})$,

T5 (Persistence of refused): $\forall e \in 2^\Sigma \times (\Sigma \times 2^\Sigma)^*, 0 \leq i \leq |e|, \sigma \in \Sigma :$
$\sigma \in \Sigma_i(e) \Rightarrow \sigma_{i+1}(e) \neq \sigma$.

Proof: T1, T2 and T5 follow immediately from the definition of the trajectory model. To prove T3, we note that a straightforward induction on trajectory length shows that if $f \sqsubset e$, then $\delta^{\pi_t}_\mathcal{P}(x^0_\mathcal{P}, e) \subseteq \delta^{\pi_t}_\mathcal{P}(x^0_\mathcal{P}, f)$, which immediately yields T3. It remains to prove T4. Fix i, and suppose that $e^i(\sigma, \emptyset) \notin \pi_t(\mathcal{P})$. Then $\delta_\mathcal{P}(\delta^{\pi_t}_\mathcal{P}(x^0_\mathcal{P}, e^i), \sigma) = \emptyset$. Since $\epsilon^*_\mathcal{P}(\delta^{\pi_t}_\mathcal{P}(x^0_\mathcal{P}, e^i)) = \delta^{\pi_t}_\mathcal{P}(x^0_\mathcal{P}, e^i)$, this implies that $\sigma \in \Re_\mathcal{P}(x)$, $\forall x \in \delta^{\pi_t}_\mathcal{P}(x^0_\mathcal{P}, e^i)$. It follows immediately that if $\bar{e}$ is obtained from e by replacing $\Sigma_i(e)$ with $\Sigma_i(e) \cup \{\sigma\}$, then $\delta^{\pi_t}_\mathcal{P}(x^0_\mathcal{P}, \bar{e}) = \delta^{\pi_t}_\mathcal{P}(x^0_\mathcal{P}, e)$ which implies that $\bar{e} \in \pi_t(\mathcal{P})$. □

Remark 2 In contrast to [6] where the properties of the trajectory model are defined axiomatically, we regard the NSM as the fundamental object and *derive* the properties of the trajectory model from the properties of NSM's.

If e is any trajectory which has the property that $\sigma \in \Sigma_i(e)$ whenever $e^i(\sigma, \emptyset) \notin \pi_t(\mathcal{P})$, then we say that e is *saturated.* Note that in T4, if e is a prefix of a maximal trajectory and $e(\sigma, \emptyset) \notin \pi_t(P)$, then $\sigma \in \Sigma_i(e)$. Thus, a prefix of a maximal trajectory is saturated. It is important to distinguish between saturated trajectories and dominant trajectories. A dominant trajectory is trivially saturated. However, while a prefix of a saturated trajectory is saturated, a prefix of a dominant trajectory is not necessarily dominant. In the special case of a deterministic process, every saturated trajectory is dominant, so the sets of saturated and dominant trajectories are identical.

Example 2 Let $\Sigma = \{a\}$, Δ_Σ the deadlock process with alphabet Σ, and let $P = \Delta_\Sigma \oplus (a \rightarrow \Delta_\Sigma)$. Then $e = \emptyset(a, \{a\})$ is dominant (and hence also saturated). The prefix $e^0 = \emptyset$ is saturated but is not dominant since P also contains the length 0 trajectory $e' = \{a\}$.

Definition 5 Given a NSM $\mathcal{P}$, the *dominant trajectory model,* denoted $\pi_{tdom}(\mathcal{P})$, of $\mathcal{P}$ is defined to be:

$$\pi_{tdom}(\mathcal{P}) := \{e \in \pi_t(\mathcal{P}) \mid \nexists f \in \pi_t(\mathcal{P}) \text{ such that } e \sqsubset f\}.$$

Similarly, the *saturated trajectory model,* denoted $\pi_{tsat}(\mathcal{P})$, of $\mathcal{P}$ is defined to be:

$$\pi_{tsat}(\mathcal{P}) := \{e \in \pi_t(\mathcal{P}) \mid \ e \text{ is saturated } \}.$$

It is clear that $\pi_{tdom}(\mathcal{P}) \subseteq \pi_{tsat}(\mathcal{P}) \subseteq \pi_t(\mathcal{P})$.

Remark 3 Since a trajectory model satisfies T3, it can be uniquely determined from its dominant trajectories or from its saturated trajectories. However, if there is no upper bound on the lengths of the trajectories in $\pi_t(\mathcal{P})$, then $\pi_t(\mathcal{P})$ is not necessarily determined by its subset of maximal trajectories. For example, if $\mathcal{P}$ is the deterministic process with language $\{a\}^*$, then the set of maximal trajectories is empty whereas $\pi_t(\mathcal{P})$ is obviously not empty.

Since a trajectory model is a more detailed model than a language model, we can obtain the language model from a trajectory model by "projecting" the trajectories onto the set Σ^*. Formally,

Definition 6 The *trace* map from trajectories to traces, denoted $tr : 2^\Sigma \times (\Sigma \times 2^\Sigma)^* \rightarrow \Sigma^*$, is defined inductively on the length of the trajectories as:

- $\forall \Sigma' \subseteq \Sigma : tr(\Sigma') = \epsilon$,
- $\forall e \in 2^\Sigma \times (\Sigma \times 2^\Sigma)^*, \sigma \in \Sigma, \Sigma' \subseteq \Sigma : tr(e(\sigma, \Sigma')) = tr(e)\sigma$.

It is clear that $tr(\pi_t(\mathcal{P})) = \pi_l(\mathcal{P})$, where the trace operator is extended to the set of trajectory models in the natural way. Given a trajectory model $T \subseteq 2^\Sigma \times (\Sigma \times 2^\Sigma)^*$ satisfying properties T1-T5, we use $L(T) := tr(T)$ to denote the language model associated with T.

2.3 Construction of Canonical State Machine

In this subsection we develop an algorithm for constructing a canonical state machine for any given trajectory model satisfying T1-T5. Given a trajectory model, $P \subseteq 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$, we have shown above that it is equivalent, in detail of description, to its saturated trajectory subset P_{sat}. We use the set of saturated trajectories for the construction of the canonical state machine. Given a finite number of event sets $\Sigma_1, \ldots, \Sigma_n \subseteq \Sigma$ for some $n \in \mathcal{N}$, we use the notation $min(\Sigma_1, \ldots, \Sigma_n)$ to denote the collection of minimal sets from among the given n sets, i.e.,

$$min(\Sigma_1, \ldots, \Sigma_n) := \{\Sigma_i, 1 \leq i \leq n \mid \nexists j \text{ such that } 1 \leq j \leq n; j \neq i; \Sigma_j \subset \Sigma_i\}$$

Lemma 1 Let P be a trajectory model.

(a) P_{sat} contains a unique minimal 0-length trajectory $\Sigma^0_{min} := \{\sigma' \in \Sigma |\ \sigma' \notin L(P)\}$.

(b) If $e \in P_{sat}$ and $e(\sigma, \emptyset) \in P$, then the family $\{\Sigma' \subseteq \Sigma |\ e(\sigma, \Sigma') \in P_{sat}\}$ has a unique minimal element given by

$$\Sigma^{(e,\sigma)}_{min} := \{\sigma' \in \Sigma |\ e(\sigma, \emptyset)(\sigma', \emptyset) \notin P\}.$$

Proof: The proof of (a) is similar to that of (b), so we include only the latter. Since $e(\sigma, \emptyset) \in P$, repeated application of T4 yields $e(\sigma, \Sigma^{(e,\sigma)}_{min}) \in P$. Since e is saturated, in order to show that $e(\sigma, \Sigma^{(e,\sigma)}_{min})$ is saturated, it suffices to show that if $e(\sigma, \Sigma^{(e,\sigma)}_{min})(\sigma', \emptyset) \notin P$, then $\sigma' \in \Sigma^{(e,\sigma)}_{min}$. Suppose $\sigma' \notin \Sigma^{(e,\sigma)}_{min}$. Then $e(\sigma, \emptyset)(\sigma', \emptyset) \in P$. By repeated application of T4, it follows that $e(\sigma, \Sigma^{(e,\sigma)}_{min})(\sigma', \emptyset) \in P$, contradiction. Thus, $e(\sigma, \Sigma^{(e,\sigma)}_{min}) \in P_{sat}$.

Finally, suppose $e(\sigma, \Sigma') \in P_{sat}$ and $\sigma' \in \Sigma^{(e,\sigma)}_{min}$–i.e., $e(\sigma, \emptyset)(\sigma', \emptyset) \notin P$. By T3, it follows that $e(\sigma, \Sigma')(\sigma', \emptyset) \notin P$. Since $e(\sigma, \Sigma')$ is saturated, this implies that $\sigma' \in \Sigma'$, so $\Sigma^{(e,\sigma)}_{min} \subseteq \Sigma'$. □

Algorithm 1 Given $P \subseteq 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$, construct a nondeterministic state machine (with ϵ-moves) $\mathcal{P} := (X_{\mathcal{P}}, \Sigma, \delta_{\mathcal{P}}, x^0_{\mathcal{P}})$, where

- $X_{\mathcal{P}} = P_{sat}$ is the state space of $\mathcal{P}$,
- $x^0_{\mathcal{P}} = \Sigma^0_{min}$ is the initial state of $\mathcal{P}$,
- $\delta_{\mathcal{P}} : X_{\mathcal{P}} \times \Sigma \cup \{\epsilon\} \rightarrow 2^{X_{\mathcal{P}}}$ is the nondeterministic transition function of $\mathcal{P}$ defined as follows:

  1. $\forall e \in P_{sat}$, $\sigma \in \Sigma$:

$$\delta_{\mathcal{P}}(e, \sigma) = \begin{cases} e(\sigma, \Sigma^{(e,\sigma)}_{min}) & \text{if } e(\sigma, \emptyset) \in P \\ \emptyset & \text{otherwise} \end{cases}$$

2. (a) $\forall \Sigma' \subseteq \Sigma$ such that $\Sigma' \in P_{sat}$:

$$\delta_{\mathcal{P}}(\Sigma', \epsilon) = min\{\Sigma'' \subseteq \Sigma |\ \Sigma'' \in P_{sat},\ \Sigma' \subset \Sigma''\}$$

(b) $\forall e(\sigma, \Sigma') \in P_{sat}$:

$$\delta_{\mathcal{P}}(e(\sigma, \Sigma'), \epsilon) = \{e(\sigma, \Sigma'') | \cdot \Sigma'' \in min\{\hat{\Sigma} \subseteq \Sigma |\ e(\sigma, \hat{\Sigma}) \in P_{sat},\ \Sigma' \subset \hat{\Sigma}\}\}$$

Algorithm 1 provides a procedure for constructing a canonical NSM $\mathcal{P}$ for a given trajectory model P. The state space of $\mathcal{P}$ equals P_{sat}, the set of saturated trajectories of P, and the initial state of $\mathcal{P}$ is the minimal 0-length saturated trajectory Σ^0_{min} of P. The state reached by executing a non-epsilon event $\sigma \in \Sigma$ from a state $e \in P_{sat}$ equals the minimal saturated trajectory of the type $e(\sigma, \Sigma')$ dominating $e(\sigma, \emptyset)$. The set of states reached by executing the epsilon event from a 0-length trajectory $\Sigma' \in P_{sat} = X_{\mathcal{P}}$ equals the set of minimal 0-length saturated trajectories dominating Σ'. Also, the set of states reached by executing the epsilon event from a trajectory $e(\sigma, \Sigma') \in P_{sat} = X_{\mathcal{P}}$ equals the set of minimal saturated trajectories of the type $e(\sigma, \Sigma'')$ dominating $e(\sigma, \Sigma')$.

Note that the canonical NSM constructed using Algorithm 1 has as many states as the number of saturated trajectories. The notion of Nerode equivalence [8] can be easily extended to the set of trajectories, and "minimal" NSM's can be constructed for given trajectory models.

Remark 4 A construction which bears some similarity to Algorithm 1 was informally described in [6, Algorithm 12.1]. However, a proof to show that the trajectory model of the canonical NSM equals P was omitted in that reference. There is also an important difference between the two algorithms. The construction in [6, Algorithm 12.1] is based on prefixes of dominant trajectories. In contrast, Algorithm 1 is based on saturated trajectories. The use of saturated trajectories for the states has the advantage of avoiding the need to introduce certain "auxiliary states" as is the case when prefixes of dominant trajectories are used. This advantage arises because the saturated trajectories satisfy the properties described in Lemma 1.

We now prove the correctness of Algorithm 1–i.e., that the trajectory model of the canonical NSM coincides with the given trajectory model.

Proposition 2 $\pi_t(\mathcal{P}) = P$, where $\mathcal{P}$ is as constructed in Algorithm 1.

Proof: We begin by showing that

$$\forall\ e = \bar{e}(\sigma, \Sigma') \in P_{sat},\ \ \Re_{\mathcal{P}}(e) = \Sigma'. \tag{1}$$

It follows from the definition of $\delta_{\mathcal{P}}$ that $\sigma' \in \Re_{\mathcal{P}}(e)$ if and only if

$$f(\sigma', \emptyset) \notin P, \quad \forall\ f = \bar{e}(\sigma, \Sigma'') \in P_{sat} \text{ such that } \Sigma' \subseteq \Sigma''.$$

If $\sigma' \in \Sigma'$, then $\sigma' \in \Sigma''$ for all such Σ''. By T5, $f(\sigma', \emptyset) \notin P$, so $\sigma' \in \Re_{\mathcal{P}}(e)$. Thus, $\Sigma' \subseteq \Re_{\mathcal{P}}(e)$. On the other hand, if $\sigma' \notin \Sigma'$, then since $e \in P_{sat}$, it follows that $e(\sigma', \emptyset) \in P$, so $\sigma' \notin \Re_{\mathcal{P}}(e)$. Thus, $\Re_{\mathcal{P}}(e) \subseteq \Sigma'$, proving (1).

Next, we claim that

$$\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e) = \{f \in P_{sat} |\ e \sqsubseteq f\}, \quad \forall\, e \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*. \tag{2}$$

We prove (2) by induction on $|e|$. Let $e = \Sigma'$, a length-0 trajectory. Using the definition of $\delta_{\mathcal{P}}$ and (1) gives

$$\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, \Sigma') = \{\Sigma'' \in \epsilon_{\mathcal{P}}^*(x_{\mathcal{P}}^0) |\ \Sigma' \subseteq \Re_{\mathcal{P}}(\Sigma'')\} = \{\Sigma'' \in P_{sat} |\ \Sigma' \subseteq \Sigma''\}.$$

This establishes (2) in the length-0 case.

For the induction step, let $e = \bar{e}(\sigma, \Sigma') \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$. Using the induction hypothesis on $\bar{e}$, (1), and the fact that P_{sat} is prefix-closed gives

$$\begin{aligned}
\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e) &= \{f \in \epsilon_{\mathcal{P}}^*(\delta_{\mathcal{P}}(\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, \bar{e}), \sigma)) |\ \Sigma' \subseteq \Re_{\mathcal{P}}(f)\} \\
&= \{f \in \epsilon_{\mathcal{P}}^*(\delta_{\mathcal{P}}(\{\bar{f} \in P_{sat} |\ \bar{e} \sqsubseteq \bar{f}\}, \sigma)) |\ \Sigma' \subseteq \Re_{\mathcal{P}}(f)\} \\
&= \{\bar{f}(\sigma, \Sigma'') \in P_{sat} |\ \bar{e} \sqsubseteq \bar{f},\ \Sigma' \subseteq \Sigma''\} \\
&= \{f \in P_{sat} |\ e \sqsubseteq f\}.
\end{aligned}$$

This completes the induction step and establishes (2).

If $e \in P_{sat}$, (2) implies that $e \in \delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e)$. Thus, $\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e)$ is nonempty, so $e \in \pi_t(\mathcal{P})$. Hence, $P_{sat} \subseteq \pi_t(\mathcal{P})$. Since every trajectory in P is dominated by a saturated trajectory and $\pi_t(\mathcal{P})$ satisfies T3, this implies that $P \subseteq \pi_t(\mathcal{P})$.

On the other hand, if $e \in \pi_t(\mathcal{P})$, then $\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e)$ is nonempty, so there exists $f \in P_{sat}$ which dominates e. Since P satisfies T3, this implies that $e \in P$, so $\pi_t(\mathcal{P}) \subseteq P$, which completes the proof. □

The following result is an immediate consequence of the proof of Proposition 2.

Corollary 1 If P is a trajectory model with canonical NSM $\mathcal{P}$, then

$$\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e) = \{f \in P_{sat} |\ e \sqsubseteq f\}, \quad \forall e \in P.$$

The following result is an immediate consequence of Propositions 1 and 2.

Theorem 1 Let $P \subseteq 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$. Then P is the trajectory model of a nondeterministic state machine (with ϵ-moves) if and only if P satisfies properties T1-T5.

2.4 Deterministic Trajectory Models

Recall that a state machine $\mathcal{P}$ is deterministic if and only if its transition function is a partial map $\delta_{\mathcal{P}} : X_{\mathcal{P}} \times \Sigma \to X_{\mathcal{P}}$. I.e., there are no ϵ-transitions and $\delta_{\mathcal{P}}(x, \sigma)$ is either empty or contains exactly one element.

Definition 7 Let $P \subseteq 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$ satisfy T1-T5. P is called a *deterministic trajectory model* if and only if there exists a deterministic state machine $\mathcal{P}$ such that $\pi_t(\mathcal{P}) = P$.

For any NSM $\mathcal{P}$, the language model can be obtained from the trajectory model via the trace operator since $\pi_l(\mathcal{P}) = tr(\pi_t(\mathcal{P}))$. In the special case when the system $\mathcal{P}$ is deterministic, the trajectory model can be recovered from the language model. Consequently, for deterministic systems, the language model is equivalent in detail of description to the trajectory model. The language model can be used to compute the trajectory model as described below. First consider the definition of the inverse operation of the trace map.

Definition 8 Let K be a nonempty prefix-closed subset of Σ^*. The *trajectory* map from traces to trajectories for the language model K, denoted $trj_K : K \rightarrow 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$, is defined inductively on the length of the traces of K as follows:

- $trj_K(\epsilon) = \{\sigma \in \Sigma \mid \sigma \notin K\}$,
- $\forall s \in K$, $\sigma \in \Sigma$ such that $s\sigma \in K : trj_K(s\sigma) = trj_K(s)(\sigma, \{\sigma' \in \Sigma \mid s\sigma\sigma' \notin K\})$.

Lemma 2 Let $\mathcal{P}$ be an NSM with language model $K := \pi_l(\mathcal{P})$. Then

(a) $trj_K(K) \subseteq \pi_t(\mathcal{P})$

(b) If $\mathcal{P}$ is deterministic, then

$$trj_K(K) = \pi_{tsat}(\mathcal{P}) = \pi_{tdom}(\mathcal{P})$$

Proof: Let $s \in K$ be a trace of length r. If $r = 0$, then $s = \epsilon$; otherwise let $s = \sigma_1\sigma_2 \ldots \sigma_r$. Let s^i denote the length-i prefix of s, and define $\hat{\Sigma}_i = \{\sigma \in \Sigma |\ s^i\sigma \notin K\}$. Set

$$e = trj_K(s) = \hat{\Sigma}_0(\sigma_1, \hat{\Sigma}_1) \ldots (\sigma_r, \hat{\Sigma}_r).$$

Since $s \in \pi_l(\mathcal{P}) = tr(\pi_t(\mathcal{P}))$, it follows from T3 that the trajectory $\emptyset(\sigma_1, \emptyset) \ldots (\sigma_r, \emptyset) \in \pi_t(\mathcal{P})$. By repeated application of T4, this implies that $e \in \pi_t(\mathcal{P})$, proving (a).

Now assume that $\mathcal{P}$ is deterministic. To prove (b), it suffices to show that e is the unique trajectory in $\pi_{tsat}(\mathcal{P})$ with $tr(e) = s$. Since every dominant trajectory is saturated and there exists a dominant trajectory with trace s, this implies that e is also the unique trajectory in $\pi_{tdom}(\mathcal{P})$ with $tr(e) = s$. Also, since there must exist a saturated trajectory with trace s, it suffices to show that if $f \in \pi_{tsat}(\mathcal{P})$ with $tr(f) = s$, then $f = e$. We use induction on $r = |e|$ to prove this together with the assertion that

$$\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e) = \delta_{\mathcal{P}}^*(x_{\mathcal{P}}^0, tr(e)) \tag{3}$$

Note that since $\mathcal{P}$ is deterministic, $\epsilon_{\mathcal{P}}^*(x) = x$, and given any $w \in K$, there exists a unique $x_w \in \delta_{\mathcal{P}}^*(x_{\mathcal{P}}^0, w)$. Furthermore, $\Re_{\mathcal{P}}(x_w) = \Sigma - \Sigma_{\mathcal{P}}(x_w) = \{\sigma \in \Sigma |\ w\sigma \notin K\}$.

If $r = 0$, then $f = \Sigma_0$ with

$$\Sigma_0 \subseteq \Re_{\mathcal{P}}(x_{\mathcal{P}}^0) = \Sigma - \Sigma_{\mathcal{P}}(x_{\mathcal{P}}^0) = \hat{\Sigma}_0.$$

Since f is saturated, $\hat{\Sigma}_0 \subseteq \Sigma_0$, so $f = e$. Also,

$$\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e) = x_{\mathcal{P}}^0 = \delta_{\mathcal{P}}^*(x_{\mathcal{P}}^0, \epsilon) = \delta_{\mathcal{P}}^*(x_{\mathcal{P}}^0, tr(e)),$$

as required.

For the induction step, express e and f as $e = \bar{e}(\sigma_r, \hat{\Sigma}_r)$, $f = \bar{f}(\sigma_r, \Sigma_r)$. Since the prefix of a saturated trajectory is saturated, $\bar{f} \in \pi_{tsat}(\mathcal{P})$. Therefore, by induction hypothesis, we may assume that $\bar{f} = \bar{e}$. Using (3) applied to $\bar{e}$, it follows that

$$\begin{aligned}
\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, f) &= \{x \in \epsilon_{\mathcal{P}}^*(\delta_{\mathcal{P}}(\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, \bar{e}), \sigma_r))|\ \Sigma_r \subseteq \Re_{\mathcal{P}}(x)\} && (4)\\
&= \{x \in \delta_{\mathcal{P}}(\delta_{\mathcal{P}}^*(x_{\mathcal{P}}^0, tr(\bar{e})), \sigma_r)|\ \Sigma_r \subseteq \Re_{\mathcal{P}}(x)\} && (5)\\
&= \{x \in \delta_{\mathcal{P}}^*(x_{\mathcal{P}}^0, tr(e))|\ \Sigma_r \subseteq \Re_{\mathcal{P}}(x)\} && (6)\\
&= \begin{cases} x_s & \text{if } \Sigma_r \subseteq \Re_{\mathcal{P}}(x_s) \\ \emptyset & \text{otherwise} \end{cases} && (7)
\end{aligned}$$

Since f is a trajectory of $\mathcal{P}$, $\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, f)$ is nonempty, so

$$\Sigma_r \subseteq \Re_{\mathcal{P}}(x_s) = \Sigma - \Sigma_{\mathcal{P}}(x_s) = \hat{\Sigma}_r.$$

Since f is saturated, $\hat{\Sigma}_r \subseteq \Sigma_r$, so $f = e$. Also, by replacing f by e and Σ_r by $\hat{\Sigma}_r$ in the string of equalities (4)-(7), we get

$$\delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e) = x_s = \delta_{\mathcal{P}}^*(x_{\mathcal{P}}^0, tr(e)).$$

This completes the induction step. □

Proposition 3 Let K be a nonempty prefixed-closed sublanguage of Σ^*, and let

$$det(K) := \{f \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*|\ \exists e \in trj_K(K) \text{ such that } f \sqsubseteq e\}.$$

Then

(a) $det(K)$ is a deterministic trajectory model.

(b) If P is any trajectory model with $L(P) = K$, then $det(K) \subseteq P$, with equality if and only if P is deterministic.

Proof: By a standard result, there exists a deterministic state machine $\mathcal{Q}$ such that $\pi_l(\mathcal{Q}) = K$. Setting $Q = \pi_t(\mathcal{Q})$ gives $L(Q) = K$. Since $\mathcal{Q}$ is deterministic, it follows from Lemma 2 that $trj_K(K) = \pi_{tdom}(\mathcal{Q})$, which implies that $det(K) = Q$. Thus, $det(K)$ is a deterministic trajectory model.

Let P be any trajectory model with $L(P) = K$. By Proposition 2, there exists a state machine $\mathcal{P}$ such that $\pi_t(\mathcal{P}) = P$. By Lemma 2, $trj_K(K) \subseteq P$, so $det(K) \subseteq P$. If P is deterministic, then we can take $\mathcal{P}$ to be deterministic, so Lemma 2 implies that $trj_K(K) = \pi_{tdom}(\mathcal{P})$, and hence $det(K) = P$. On the other hand, if $det(K) = P$, then P is deterministic by (a). □

Remark 5 It follows from Proposition 3 that given a nonempty prefix-closed language K, there is a unique deterministic trajectory model with language K. Furthermore, this trajectory model $det(K)$ can be constructed from K by applying the map trj_K and taking dominance closure. This trajectory model is the unique minimal element (with respect to inclusion) of the family of trajectory models having language K.

3 Prioritized Synchronous Composition

In this section, we define the PSC of two NSM's (with ϵ-moves), which induces a PSC operation on trajectory models. We also prove that the trajectory modeling framework is a language congruence with respect to PSC. Our definition of the PSC of NSM's is more general than the one in [5], since the "silent" transitions, i.e., transitions labeled ϵ, were not included. As discussed in Section 1, a priority set is associated with a system. This means that for an event which belongs to the priority set of a system to occur in the PSC with another system, the former system must participate.

Definition 9 Let $\mathcal{P} = (X_\mathcal{P}, \Sigma, \delta_\mathcal{P}, x^0_\mathcal{P})$ and $\mathcal{Q} = (X_\mathcal{Q}, \Sigma, \delta_\mathcal{Q}, x^0_\mathcal{Q})$ be two NSM's (with ϵ-moves). Let $A, B \subseteq \Sigma$ be the priority sets of $\mathcal{P}, \mathcal{Q}$ respectively. Then the PSC of $\mathcal{P}$ and $\mathcal{Q}$, denoted $\mathcal{P}\ {}_A\|_B\ \mathcal{Q}$, is another NSM defined as:

$$\mathcal{P}\ {}_A\|_B\ \mathcal{Q} := \mathcal{R} := (X_\mathcal{R}, \Sigma, \delta_\mathcal{R}, x^0_\mathcal{R}),$$

where $X_\mathcal{R} = X_\mathcal{P} \times X_\mathcal{Q}, x^0_\mathcal{R} = (x^0_\mathcal{P}, x^0_\mathcal{Q})$, and the transition function $\delta_\mathcal{R} : X_\mathcal{R} \times \Sigma \cup \{\epsilon\} \to 2^{X_\mathcal{R}}$ is defined as:

$\forall x_r = (x_p, x_q) \in X_\mathcal{R}, \sigma \in \Sigma :$

$$\delta_\mathcal{R}(x_r, \sigma) = \begin{cases} \delta_\mathcal{P}(x_p, \sigma) \times \delta_\mathcal{Q}(x_q, \sigma) & \text{if } \delta_\mathcal{P}(x_p, \sigma) \neq \emptyset, \delta_\mathcal{Q}(x_q, \sigma) \neq \emptyset \\ \delta_\mathcal{P}(x_p, \sigma) \times \{x_q\} & \text{if } \delta_\mathcal{P}(x_p, \sigma) \neq \emptyset, \sigma \in \Re_\mathcal{Q}(x_q), \sigma \notin B \\ \{x_p\} \times \delta_\mathcal{Q}(x_q, \sigma) & \text{if } \delta_\mathcal{Q}(x_q, \sigma) \neq \emptyset, \sigma \in \Re_\mathcal{P}(x_p), \sigma \notin A \\ \emptyset & \text{otherwise} \end{cases}$$

$\forall x_r = (x_p, x_q) \in X_\mathcal{R},$

$$\delta_\mathcal{R}(x_r, \epsilon) = [\delta_\mathcal{P}(x_p, \epsilon) \cup \{x_p\}] \times [\delta_\mathcal{Q}(x_q, \epsilon) \cup \{x_q\}] - \{(x_p, x_q)\}$$

Thus, if an event is executable in the current states of both $\mathcal{P}$ and $\mathcal{Q}$, then it can be executed in $\mathcal{R}$, in which case both $\mathcal{P}$ and $\mathcal{Q}$ change their states synchronously according to their respective transitions. An event can be executed asynchronously by one of the systems if it is executable by that system and is not in the priority set of, nor can be executed in any state in the epsilon-closure of the current state of the other system. In this case, a state transition occurs in one system while no state change occurs in the other system. The hidden transitions–i.e., those labeled by ϵ–can occur either synchronously or asynchronously. It is clear that an event in $A \cap B$ occurs only synchronously. Such synchronous execution

is not required for events that do not belong to $A \cap B$. However, if an event that does not belong to $A \cap B$ is defined at states $x_p \in X_{\mathcal{P}}$ and $x_q \in X_{\mathcal{Q}}$, then it occurs synchronously at state $x_r = (x_p, x_q) \in X_{\mathcal{R}}$. Synchronous execution of such events is called *broadcast synchronization.*

Remark 6 If $A = B = \Sigma$, then an event is executable in the composed system if and only if it is executable in both systems. Thus this case corresponds to SSC. In contrast, if $A = B = \emptyset$, then an event is executable in the composed system if and only if it is executable in either of the systems. This corresponds to an interleaving composition of the systems modified by the requirement that events which are executable by both systems are executed synchronously.

If $\mathcal{P}$ represents an uncontrolled plant, $\mathcal{Q}$ a supervisor, and $\mathcal{P}\ {}_A\|_B\ \mathcal{Q}$ the controlled plant or the closed loop system, then (i) $A \cap B$ is the set of strict synchronization events and can be used to represent the set of *controllable* events; (ii) $A - B$ is the set of priority events only of $\mathcal{P}$ and can be used to represent the set of *uncontrollable* events; (iii) $B - A$ is the set of priority events only of $\mathcal{Q}$ and can be used to represent the set of *driven* events; (iv) $\Sigma - (A \cup B)$ is assumed to be empty, for events in $\Sigma - A \cup B$ belong neither to the priority set of the plant nor to that of the supervisor.

To simplify future notation, we define for any two event sets $\Sigma', \Sigma'' \subseteq \Sigma$:

$$S_{A,B}(\Sigma', \Sigma'') := [\Sigma' \cap \Sigma''] \cup [A \cap \Sigma'] \cup [B \cap \Sigma''].$$

The following lemma gives two useful properties of the PSC of NSM's. It is a straightforward consequence of the definition of PSC.

Lemma 3 If $\mathcal{R} = \mathcal{P}\ {}_A\|_B\ \mathcal{Q}$ and $x_r = (x_p, x_q) \in X_{\mathcal{R}}$, then

(a) $\epsilon^*_{\mathcal{R}}(x_r) = \epsilon^*_{\mathcal{P}}(x_p) \times \epsilon^*_{\mathcal{Q}}(x_q)$,

(b) $\Re_{\mathcal{R}}(x_r) = S_{A,B}(\Re_{\mathcal{P}}(x_p), \Re_{\mathcal{Q}}(x_q))$.

In other words, a state $x'_r = (x'_p, x'_q) \in X_{\mathcal{R}}$ belongs to the epsilon-closure of $x_r = (x_p, x_q)$ if and only if x'_p (respectively, x'_q) belongs to epsilon-closure of x_p (respectively, x_q). Also, an event is refused in $\mathcal{P}\ {}_A\|_B\ \mathcal{Q}$ if and only if either it is refused in both $\mathcal{P}$ and $\mathcal{Q}$, or it belongs to the priority set of $\mathcal{P}$ and is refused in $\mathcal{P}$, or it belongs to the priority set of $\mathcal{Q}$ and is refused in $\mathcal{Q}$.

We next consider the trajectory model of the PSC of two systems, and obtain its relation to the trajectory models of the component systems. Using the definition of $\mathcal{P}\ {}_A\|_B\ \mathcal{Q}$ and that of its refusal map $\Re_{\mathcal{P}\ {}_A\|_B\ \mathcal{Q}}$, the trajectory model $\pi_t(\mathcal{P}\ {}_A\|_B\ \mathcal{Q})$ is easily obtained from its definition developed in the previous subsection. In order to obtain the relationship between $\pi_t(\mathcal{P})$, $\pi_t(\mathcal{Q})$ and $\pi_t(\mathcal{P}\ {}_A\|_B\ \mathcal{Q})$, we first define the PSC of a pair of trajectories. Although we use the same notation $e_p\ {}_A\|_B\ e_q$ as is used in [6], our definition is *not* precisely the same as the one in this reference.

Definition 10 Let $e_p \in \pi_t(\mathcal{P})$ and $e_q \in \pi_t(\mathcal{Q})$. Then the PSC of e_p and e_q (with respect to $\mathcal{P}$ and $\mathcal{Q}$), denoted $e_p \ {}_A\|_B \ e_q$, is defined inductively on $|e_p| + |e_q|$ as follows:

$\forall \Sigma_p, \Sigma_q \subseteq \Sigma$ such that $\Sigma_p \in \pi_t(\mathcal{P}), \Sigma_q \in \pi_t(\mathcal{Q}) : \Sigma_p \ {}_A\|_B \ \Sigma_q := \{\Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)\}$,

$\forall e_p \in \pi_t(\mathcal{P}); e_q \in \pi_t(\mathcal{Q}); \sigma_p, \sigma_q \in \Sigma; \Sigma_p, \Sigma_q \subseteq \Sigma$ such that $e_p(\sigma_p, \Sigma_p) \in \pi_t(\mathcal{P}), e_q(\sigma_q, \Sigma_q) \in \pi_t(\mathcal{Q})$:
$e_p(\sigma_p, \Sigma_p) \ {}_A\|_B \ e_q(\sigma_q, \Sigma_q) := T_1 \cup T_2 \cup T_3$, where

$$T_1 = \begin{cases} \{e(\sigma_p, \Sigma') \mid e \in e_p \ {}_A\|_B \ e_q(\sigma_q, \Sigma_q); \Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)\} & \text{if } \sigma_p \notin B \text{ and} \\ & e_q(\sigma_q, \Sigma_q)(\sigma_p, \emptyset) \notin \pi_t(\mathcal{Q}) \\ \emptyset & \text{otherwise} \end{cases}$$

$$T_2 = \begin{cases} \{e(\sigma_q, \Sigma') \mid e \in e_p(\sigma_p, \Sigma_p) \ {}_A\|_B \ e_q; \Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)\} & \text{if } \sigma_q \notin A \text{ and} \\ & e_p(\sigma_p, \Sigma_p)(\sigma_q, \emptyset) \notin \pi_t(\mathcal{P}) \\ \emptyset & \text{otherwise} \end{cases}$$

$$T_3 = \begin{cases} \{e(\sigma, \Sigma') \mid e \in e_p \ {}_A\|_B \ e_q; \Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)\} & \text{if } \sigma_p = \sigma_q := \sigma \\ \emptyset & \text{otherwise} \end{cases}$$

Thus the PSC of two zero length trajectories $\Sigma_p \in \pi_t(\mathcal{P})$ and $\Sigma_q \in \pi_t(\mathcal{Q})$, which correspond to initial refusal sets of $\pi_t(\mathcal{P})$ and $\pi_t(\mathcal{Q})$ respectively, is obtained by computing $S_{A,B}(\Sigma_p, \Sigma_q)$ which corresponds to an initial refusal set of $\pi_t(\mathcal{P} \ {}_A\|_B \ \mathcal{Q})$. Next the PSC of two trajectories $e_p(\sigma_p, \Sigma_p) \in \pi_t(\mathcal{P})$ and $e_q(\sigma_q, \Sigma_q) \in \pi_t(\mathcal{Q})$ is obtained by considering these three possible cases: (i) a trajectory belonging to $e_p \ {}_A\|_B \ e_q(\sigma_q, \Sigma_q)$ has already been executed in the composed system, and at this point, σ_p is executable in $\mathcal{P}$ (indicated by $e_p(\sigma_p, \Sigma_p) \in \pi_t(\mathcal{P})$), the occurrence of σ_p cannot be blocked by $\mathcal{Q}$ (indicated by $\sigma_p \notin B$), and $\mathcal{Q}$ cannot participate in the occurrence of σ_p (indicated by $e_q(\sigma_q, \Sigma_q)(\sigma_p, \emptyset) \notin \pi_t(\mathcal{Q})$); (ii) a trajectory belonging to $e_p(\sigma_p, \Sigma_p) \ {}_A\|_B \ e_q$ has already been executed in the composed system, and at this point, σ_q is executable in $\mathcal{Q}$, and $\mathcal{P}$ can neither block the occurrence of σ_q, nor it can participate in the occurrence of σ_q; (iii) $\sigma_p = \sigma_q := \sigma$; a trajectory belonging to $e_p \ {}_A\|_B \ e_q$ has already been executed in the composed system, and at this point, σ is executable in both $\mathcal{P}$ and $\mathcal{Q}$.

Remark 7 It is clear from Definition 10 that if $A = B = \Sigma$, which corresponds to the case of SSC, then an initial refusal set of $\pi_t(\mathcal{P} \ {}_A\|_B \ \mathcal{Q})$ equals the union of an initial refusal set of $\pi_t(\mathcal{P})$ and an initial refusal set of $\pi_t(\mathcal{Q})$, since $S_{\Sigma,\Sigma}(\Sigma_p, \Sigma_q) = \Sigma_p \cup \Sigma_q$. Also, note that the sets $T_1 = T_2 = \emptyset$ since the conditions "$\sigma_p \notin B$" and "$\sigma_q \notin A$" both evaluate to "false". Hence the PSC of $e_p(\sigma_p, \Sigma_p) \in \pi_t(\mathcal{P})$ and $e_q(\sigma_q, \Sigma_q) \in \pi_t(\mathcal{Q})$ is nonempty if and only if the set T_3 is nonempty, which requires that $\sigma_p = \sigma_q$. Using induction, it can be easily concluded that the SSC of trajectories $e_p \in \pi_t(\mathcal{P})$ and $e_q \in \pi_t(\mathcal{Q})$ is a nonempty set if and only if $tr(e_p) = tr(e_q)$, in which case, $tr(e_p \ {}_\Sigma\|_\Sigma \ e_q) = tr(e_p) = tr(e_q)$, and the ith refusal set of any trajectory in $e_p \ {}_\Sigma\|_\Sigma \ e_q$ is any subset of the union of the ith refusal set of e_p and the ith refusal set of e_q, for each $0 \leq i \leq |e_p| = |e_q|$.

We can extend the definition of the PSC of a pair of trajectories to the PSC of the trajectory models. With a slight abuse of notation, we use the same symbol $_A\|_B$ for the PSC of the NSM's $\mathcal{P}, \mathcal{Q}$ and for the PSC of their corresponding trajectory models $\pi_t(\mathcal{P}), \pi_t(\mathcal{Q})$.

Definition 11 The PSC of the trajectory models $\pi_t(\mathcal{P}), \pi_t(\mathcal{Q})$ is defined to be

$$\pi_t(\mathcal{P}) \ {}_A\|_B \ \pi_t(\mathcal{Q}) := \{e_p \ {}_A\|_B \ e_q \mid e_p \in \pi_t(\mathcal{P}), e_q \in \pi_t(\mathcal{Q})\}.$$

The following result shows that the trajectory model of the PSC of NSM's is the PSC of their corresponding trajectory models. Equivalently, it states that the PSC operation on NSM's induces a PSC operation on trajectory models, and the induced operation is precisely the one described in Definition 11.

Theorem 2 For any NSM's $\mathcal{P}, \mathcal{Q}$,

$$\pi_t(\mathcal{P} \ {}_A\|_B \ \mathcal{Q}) = \pi_t(\mathcal{P}) \ {}_A\|_B \ \pi_t(\mathcal{Q})$$

.

Proof: Let $\mathcal{R} = \mathcal{P} \ {}_A\|_B \ \mathcal{Q}$. First we show that

$$\pi_t(\mathcal{R}) \subseteq \pi_t(\mathcal{P}) \ {}_A\|_B \ \pi_t(\mathcal{Q}) \tag{8}$$

We prove by induction on trajectory length that if $e \in \pi_t(\mathcal{R})$ and $x_r = (x_p, x_q) \in \delta_{\mathcal{R}}^{\pi_t}(x_{\mathcal{R}}^0, e)$, then there exist $e_p \in \pi_t(\mathcal{P})$, $e_q \in \pi_t(\mathcal{Q})$ such that

(i) the final refusal sets of e_p, e_q are $\Re_{\mathcal{P}}(x_p)$, $\Re_{\mathcal{Q}}(x_q)$ respectively;

(ii) $e \in e_p \ {}_A\|_B \ e_q$;

(iii) $x_r \in \delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, e_p) \times \delta_{\mathcal{Q}}^{\pi_t}(x_{\mathcal{Q}}^0, e_q)$

Consider a 0-length trajectory $e = \Sigma' \in \pi_t(\mathcal{R})$. Then there exists $x_r = (x_p, x_q) \in \epsilon_{\mathcal{R}}^*(x_{\mathcal{R}}^0)$ such that $\Sigma' \subseteq \Re_{\mathcal{R}}(x_r)$. Lemma 3 implies that $x_p \in \epsilon_{\mathcal{P}}^*(x_{\mathcal{P}}^0)$, $x_q \in \epsilon_{\mathcal{Q}}^*(x_{\mathcal{Q}}^0)$, $\Sigma' \subseteq S_{A,B}(\Re_{\mathcal{P}}(x_p), \Re_{\mathcal{Q}}(x_q))$. Setting $e_p = \Re_{\mathcal{P}}(x_p)$, $e_q = \Re_{\mathcal{Q}}(x_q)$, it follows that (i),(ii),(iii) are satisfied.

For the induction step, consider a trajectory $e = \bar{e}(\sigma, \Sigma') \in \pi_t(\mathcal{R})$. Then there exist $\bar{x}_r = (\bar{x}_p, \bar{x}_q) \in \delta_{\mathcal{R}}^{\pi_t}(x_{\mathcal{R}}^0, \bar{e})$, $x_r' = (x_p', x_q') \in \delta_{\mathcal{R}}(\bar{x}_r, \sigma)$, $x_r = (x_p, x_q) \in \epsilon_{\mathcal{R}}^*(x_r')$ such that $\Sigma' \subseteq \Re_{\mathcal{R}}(x_r)$. By induction hypothesis, there exist $\bar{e}_p \in \pi_t(\mathcal{P})$, $\bar{e}_q \in \pi_t(\mathcal{Q})$ with final refusal sets $\Re_{\mathcal{P}}(\bar{x}_p)$, $\Re_{\mathcal{Q}}(\bar{x}_q)$ respectively such that

$$\bar{e} \in \bar{e}_p \ {}_A\|_B \ \bar{e}_q, \quad \bar{x}_p \in \delta_{\mathcal{P}}^{\pi_t}(x_{\mathcal{P}}^0, \bar{e}_p), \quad \bar{x}_q \in \delta_{\mathcal{Q}}^{\pi_t}(x_{\mathcal{Q}}^0, \bar{e}_q)$$

Since σ is executable in $\bar{x}_r$, it follows from Definition 9 that there are three cases:

(a) $\delta_{\mathcal{P}}(\bar{x}_p, \sigma) \neq \emptyset$, $\delta_{\mathcal{Q}}(\bar{x}_q, \sigma) \neq \emptyset$

(b) $\delta_{\mathcal{P}}(\bar{x}_p, \sigma) \neq \emptyset$, $\sigma \in \Re_{\mathcal{Q}}(\bar{x}_q)$, $\sigma \notin B$

(c) $\delta_Q(\bar{x}_q, \sigma) \neq \emptyset$, $\sigma \in \Re_{\mathcal{P}}(\bar{x}_p)$, $\sigma \notin A$.

By symmetry, it suffices to consider cases (a) and (b).

In case (a), $x'_p \in \delta_{\mathcal{P}}(\bar{x}_p, \sigma)$, $x'_q \in \delta_{\mathcal{Q}}(\bar{x}_q, \sigma)$. Setting $e_p = \bar{e}_p(\sigma, \Re_{\mathcal{P}}(x_p))$, $e_q = \bar{e}_q(\sigma, \Re_{\mathcal{Q}}(x_q))$ and using the fact that

$$\Sigma' \subseteq \Re_{\mathcal{R}}(x_r) = S_{A,B}(\Re_{\mathcal{P}}(x_p), \Re_{\mathcal{Q}}(x_q)), \tag{9}$$

it follows easily that $e_p \in \pi_t(\mathcal{P})$, $e_q \in \pi_t(\mathcal{Q})$ and conditions (i),(ii),(iii) are satisfied.

In case (b), $x'_p \in \delta_{\mathcal{P}}(\bar{x}_p, \sigma)$, $x'_q = \bar{x}_q$. Set $e_p = \bar{e}_p(\sigma, \Re_{\mathcal{P}}(x_p))$ and let e_q be the trajectory obtained from $\bar{e}_q$ by replacing its final refusal set $\Re_{\mathcal{Q}}(\bar{x}_q)$ with the set $\Re_{\mathcal{Q}}(x_q)$. (Since $x_q \in \epsilon^*_{\mathcal{Q}}(\bar{x}_q)$, the new final refusal set will contain the old final refusal set.) Then $e_p \in \pi_t(\mathcal{P})$, $e_q \in \pi_t(\mathcal{Q})$ and conditions (i),(iii) are clearly satisfied. It follows from Definition 10 that

$$\bar{e} \in \bar{e}_p \;{}_A\|_B\; \bar{e}_q \subseteq \bar{e}_p \;{}_A\|_B\; e_q.$$

Since $\sigma \in \Re_{\mathcal{Q}}(\bar{x}_q) \subseteq \Re_{\mathcal{Q}}(x_q)$, it follows from property T5 that $e_q(\sigma, \emptyset) \notin \pi_t(\mathcal{Q})$. Since $\sigma \notin B$ and (9) holds, it follows from Definition 10 that condition (ii) is satisfied. This completes the induction step and establishes (8).

It remains to show that

$$\pi_t(\mathcal{P}) \;{}_A\|_B\; \pi_t(\mathcal{Q}) \subseteq \pi_t(\mathcal{R}) \tag{10}$$

We prove by induction on $|e_p| + |e_q|$ that if $e \in e_p \;{}_A\|_B\; e_q$ with $e_p \in \pi_t(\mathcal{P})$, $e_q \in \pi_t(\mathcal{Q})$, then

$$\delta^{\pi_t}_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p) \times \delta^{\pi_t}_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q) \subseteq \delta^{\pi_t}_{\mathcal{R}}(x^0_{\mathcal{R}}, e) \tag{11}$$

Since the set on the left side is nonempty by assumption, this implies that the set on the right side is nonempty–i.e., that $e \in \pi_t(\mathcal{R})$.

Let $e_p = \Sigma_p$, $e_q = \Sigma_q$ be 0-length trajectories of $\mathcal{P}$, $\mathcal{Q}$ respectively, and let $x_p \in \delta^{\pi_t}_{\mathcal{P}}(x^0_{\mathcal{P}}, \Sigma_p)$, $x_q \in \delta^{\pi_t}_{\mathcal{Q}}(x^0_{\mathcal{Q}}, \Sigma_q)$. Then

$$x_p \in \epsilon^*_{\mathcal{P}}(x^0_{\mathcal{P}}), \quad x_q \in \epsilon^*_{\mathcal{Q}}(x^0_{\mathcal{Q}}), \quad \Sigma_p \subseteq \Re_{\mathcal{P}}(x_p), \quad \Sigma_q \subseteq \Re_{\mathcal{Q}}(x_q).$$

Let $x_r = (x_p, x_q)$. Then $x_r \in \epsilon^*_{\mathcal{R}}(x^0_{\mathcal{R}})$. It follows from Definition 10 and Lemma 3 that $e = \Sigma'$ with

$$\Sigma' \subset S_{A,B}(\Sigma_p, \Sigma_q) \subseteq S_{A,B}(\Re_{\mathcal{P}}(x_p), \Re_{\mathcal{Q}}(x_q)) = \Re_{\mathcal{R}}(x_r).$$

This shows that $x_r \in \delta^{\pi_t}_{\mathcal{R}}(x^0_{\mathcal{R}}, \Sigma')$, so (11) holds in the 0-length case.

For the induction step, write $e_p = \bar{e}_p(\sigma_p, \Sigma_p)$, $e_q = \bar{e}_q(\sigma_q, \Sigma_q)$, and suppose $e = \bar{e}(\sigma, \Sigma') \in e_p \;{}_A\|_B\; e_q$. It follows from Definition 10 that there are three cases to consider:

(d) $\sigma = \sigma_p = \sigma_q$, $\bar{e} \in \bar{e}_p \;{}_A\|_B\; \bar{e}_q$, $\Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)$

(e) $\sigma = \sigma_p$, $\sigma \notin B$, $e_q(\sigma, \emptyset) \notin \pi_t(\mathcal{Q})$, $\bar{e} \in \bar{e}_p \;{}_A\|_B\; e_q$, $\Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)$

(f) $\sigma = \sigma_q$, $\sigma \notin A$, $e_p(\sigma, \emptyset) \notin \pi_t(\mathcal{P})$, $\bar{e} \in e_p \;{}_A\|_B\; \bar{e}_q$, $\Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)$

By symmetry, it suffices to consider cases (d),(e).

For case (d), let $x_r = (x_p, x_q) \in \delta^{\pi_t}_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p) \times \delta^{\pi_t}_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q)$. Then $\Sigma_p \subseteq \Re_{\mathcal{P}}(x_p)$, $\Sigma_q \subseteq \Re_{\mathcal{Q}}(x_q)$, so

$$\Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q) \subseteq S_{A,B}(\Re_{\mathcal{P}}(x_p), \Re_{\mathcal{Q}}(x_q)) = \Re_{\mathcal{R}}(x_r). \tag{12}$$

Since $x_p \in \delta^{\pi_t}_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p)$, $x_q \in \delta^{\pi_t}_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q)$, there exist $\bar{x}_p \in \delta^{\pi_t}_{\mathcal{P}}(x^0_{\mathcal{P}}, \bar{e}_p)$, $\bar{x}_q \in \delta^{\pi_t}_{\mathcal{Q}}(x^0_{\mathcal{Q}}, \bar{e}_q)$, $x'_p \in \delta_{\mathcal{P}}(\bar{x}_{\mathcal{P}}, \sigma)$, $x'_q \in \delta_{\mathcal{Q}}(\bar{x}_{\mathcal{Q}}, \sigma)$ such that $x_p \in \epsilon^*_{\mathcal{P}}(x'_p)$, $x_q \in \epsilon^*_{\mathcal{Q}}(x'_q)$. Let $\bar{x}_r = (\bar{x}_p, \bar{x}_q)$. It follows from Definition 9 that $(x'_p, x'_q) \in \delta_R(\bar{x}_r, \sigma)$, while by induction hypothesis, we may assume that $\bar{x}_r \in \delta^{\pi_t}_{\mathcal{R}}(x^0_{\mathcal{R}}, \bar{e})$. Then

$$x_r \in \epsilon^*_{\mathcal{R}}((x'_p, x'_q)) \subseteq \epsilon^*_{\mathcal{R}}(\delta_{\mathcal{R}}(\bar{x}_r, \sigma)) \subseteq \epsilon^*_{\mathcal{R}}(\delta_{\mathcal{R}}(\delta^{\pi_t}_{\mathcal{R}}(x^0_{\mathcal{R}}, \bar{e}), \sigma)). \tag{13}$$

We conclude from (12) and (13) that $x_r \in \delta^{\pi_t}_{\mathcal{R}}(x^0_{\mathcal{R}}, e)$ as required.

For case (e), let $x_r = (x_p, x_q) \in \delta^{\pi_t}_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p) \times \delta^{\pi_t}_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q)$. The inclusions given by (12) hold as in the previous case. Since $x_p \in \delta^{\pi_t}_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p)$, there exist $\bar{x}_p \in \delta^{\pi_t}_{\mathcal{P}}(x^0_{\mathcal{P}}, \bar{e}_p)$ and $x'_p \in \delta_{\mathcal{P}}(\bar{x}_{\mathcal{P}}, \sigma)$ such that $x_p \in \epsilon^*_{\mathcal{P}}(x'_p)$. Let $\bar{x}_r = (\bar{x}_p, x_q)$. We have

$$\delta_{\mathcal{Q}}(\epsilon^*_{\mathcal{Q}}(x_q), \sigma) \subseteq \delta_{\mathcal{Q}}(\delta^{\pi_t}_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q), \sigma) = \emptyset,$$

where the final equality follows from the assumption that $e_q(\sigma, \emptyset) \notin \pi_t(\mathcal{Q})$. This implies that $\sigma \in \Re_{\mathcal{Q}}(x_q)$. It then follows from Definition 9 that $(x'_p, x_q) \in \delta_{\mathcal{R}}(\bar{x}_r, \sigma)$, while by induction hypothesis, we may assume that $\bar{x}_r \in \delta^{\pi_t}_{\mathcal{R}}(x^0_{\mathcal{R}}, \bar{e})$. Then

$$x_r \in \epsilon^*_{\mathcal{R}}((x'_p, x_q)) \subseteq \epsilon^*_{\mathcal{R}}(\delta_{\mathcal{R}}(\bar{x}_r, \sigma)) \subseteq \epsilon^*_{\mathcal{R}}(\delta_{\mathcal{R}}(\delta^{\pi_t}_{\mathcal{R}}(x^0_{\mathcal{R}}, \bar{e}), \sigma)). \tag{14}$$

We conclude from (12) and (14) that $x_r \in \delta^{\pi_t}_{\mathcal{R}}(x^0_{\mathcal{R}}, e)$ as required. This completes the induction step and establishes (10). □

Corollary 2 The trajectory model is a language congruence with respect to the operation of PSC.

Proof: Suppose that $\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}_1, \mathcal{Q}_2$ are NSM's with $\pi_t(\mathcal{P}_1) = \pi_t(\mathcal{P}_2)$ and $\pi_t(\mathcal{Q}_1) = \pi_t(\mathcal{Q}_2)$. Then

$$\begin{aligned} \pi_l(\mathcal{P}_1 \,{}_A\|_B\, \mathcal{Q}_1) = tr(\pi_t(\mathcal{P}_1 \,{}_A\|_B\, \mathcal{Q}_1) &= tr(\pi_t(\mathcal{P}_1) \,{}_A\|_B\, \pi_t(\mathcal{Q}_1)) = tr(\pi_t(\mathcal{P}_2) \,{}_A\|_B\, \pi_t(\mathcal{Q}_2)) \\ &= tr(\pi_t(\mathcal{P}_2 \,{}_A\|_B\, \mathcal{Q}_2) = \pi_l(\mathcal{P}_2 \,{}_A\|_B\, \mathcal{Q}_2). \end{aligned}$$

□

Remark 8 Theorem 2 shows that the trajectory model of $\mathcal{P} \,{}_A\|_B\, \mathcal{Q}$ can be described using only $\pi_t(\mathcal{P})$ and $\pi_t(\mathcal{Q})$, and not $\mathcal{P}, \mathcal{Q}$ directly. This is in contrast to the situation with the failures model. Theorem 2 and Corollary 2 both fail if the trajectory model is replaced with the failures model. The equality of failures models does not necessarily imply the equality of failures models–or even language models–under prioritized synchronous composition with a fixed system [5, Example 7]. The result in Corollary 2 was mentioned without proof in [5, 6]. However, its rigorous demonstration depends on the precise definition given above for the PSC of NSM's (with ϵ-moves) as well as of the projection map π_t from NSM's to trajectory models.

4 Properties of Prioritized Synchronous Composition

In this section we study some of the properties of the PSC of two or more trajectory models, which are used in Section 5 for the synthesis of supervisors which control the behavior of nondeterministic plants via PSC.

4.1 Associativity

We begin by providing a proof for the following result which is stated without proof as part of [6, Theorem 13.4]:

Proposition 4 For any trajectory models P, Q, R and priority sets A, B, C

$$(P \ {}_A\|_B \ Q) \ {}_{A\cup B}\|_C \ R = P \ {}_A\|_{B\cup C} \ (Q \ {}_B\|_C \ R)$$

This can be interpreted as an associative property as follows. Let P, Q denote trajectory models with alphabet Σ, and let A, B be subsets of Σ. We refer to the pairs (P, A), (Q, B) as *prioritized processes*, and define their *synchronous composition* to be the prioritized process

$$(P, A) \parallel (Q, B) := (P \ {}_A\|_B \ Q, \ A \cup B).$$

Then Proposition 4 asserts that

$$((P, A) \parallel (Q, B)) \parallel (R, C) = (P, A) \parallel ((Q, B) \parallel (R, C)).$$

Thus, the result is simply the associative property for the synchronous composition of prioritized processes.

In order the prove Proposition 4 we will use the following result which gives a monotonicity property of the PSC of trajectories with respect to the dominance partial order.

Lemma 4 *Let P, Q be trajectory models, $A, B \subseteq \Sigma$, $f_p, e_p \in P$, $f_q, e_q \in Q$, with $f_p \sqsubseteq e_p$, $f_q \sqsubseteq e_q$. Then*

$$f_p \ {}_A\|_B \ f_q \subseteq e_p \ {}_A\|_B \ e_q.$$

Proof: The proof is by induction on the sum of trajectory lengths $|e_p|+|e_q|$. If $|e_p|+|e_q| = 0$, then $f_p = \hat{\Sigma}_p$, $e_p = \Sigma_p$, $f_q = \hat{\Sigma}_q$, $e_q = \Sigma_q$ with $\hat{\Sigma}_p \subseteq \Sigma_p$, $\hat{\Sigma}_q \subseteq \Sigma_q$. Let $f \in f_p \ {}_A\|_B \ f_q$. Then

$$f = \Sigma' \subseteq S_{A,B}(\hat{\Sigma}_p, \hat{\Sigma}_q) \subseteq S_{A,B}(\Sigma_p, \Sigma_q).$$

Thus, $f \in e_p \ {}_A\|_B \ e_q$.

For the induction step, write $f_p = \bar{f}_p(\sigma_p, \hat{\Sigma}_p)$, $e_p = \bar{e}_p(\sigma_p, \Sigma_p)$, $f_q = \bar{f}_q(\sigma_q, \hat{\Sigma}_q)$, $e_q = \bar{e}_q(\sigma_q, \Sigma_q)$ with $\bar{f}_p \sqsubseteq \bar{e}_p$, $\hat{\Sigma}_p \subseteq \Sigma_p$, $\bar{f}_q \sqsubseteq \bar{e}_q$, $\hat{\Sigma}_q \subseteq \Sigma_q$. Let $f = \bar{f}(\sigma, \Sigma') \in f_p \ {}_A\|_B \ f_q$. There are three cases to consider: (1) Suppose $\bar{f} \in \bar{f}_p \ {}_A\|_B \ \bar{f}_q$, $\sigma = \sigma_p = \sigma_q$, $\Sigma' \subseteq S_{A,B}(\hat{\Sigma}_p, \hat{\Sigma}_q)$. By the induction hypothesis, $\bar{f} \in \bar{e}_p \ {}_A\|_B \ \bar{e}_q$. Since $\Sigma' \subseteq S_{A,B}(\hat{\Sigma}_p, \hat{\Sigma}_q) \subseteq S_{A,B}(\Sigma_p, \Sigma_q)$, this implies that $f \in e_p \ {}_A\|_B \ e_q$. (2) Suppose $\bar{f} \in \bar{f}_p \ {}_A\|_B \ f_q$, $\sigma = \sigma_p \notin B$, $f_q(\sigma, \emptyset) \notin Q$, $\Sigma' \subseteq$

$S_{A,B}(\hat{\Sigma}_p, \hat{\Sigma}_q)$. By induction hypothesis, $\bar{f} \in \bar{e}_p \;{}_A\|_B\; e_q$. Also, $e_q(\sigma, \emptyset) \notin Q$, since otherwise T3 would imply that $f_q(\sigma, \emptyset) \in Q$, contradiction. Since $\Sigma' \subseteq S_{A,B}(\hat{\Sigma}_p, \hat{\Sigma}_q) \subseteq S_{A,B}(\Sigma_p, \Sigma_q)$, this implies that $f \in e_p \;{}_A\|_B\; e_q$. (3) Suppose $\bar{f} \in f_p \;{}_A\|_B\; \bar{f}_q$, $\sigma = \sigma_q \notin A$, $f_p(\sigma, \emptyset) \notin P$, $\Sigma' \subseteq S_{A,B}(\hat{\Sigma}_p, \hat{\Sigma}_q)$. This case is analogous to case (2). Hence, the induction step is complete. □

Proof of Proposition 4: By symmetry, it suffices to prove the inclusion

$$(P \;{}_A\|_B\; Q) \;{}_{A\cup B}\|_C\; R \subseteq P \;{}_A\|_{B\cup C}\; (Q \;{}_B\|_C\; R).$$

By Lemma 4, it suffices to show that if e_p, e_q, e_r are *saturated* trajectories of P, Q, R respectively, then

$$(e_p \;{}_A\|_B\; e_q) \;{}_{A\cup B}\|_C\; e_r \subseteq P \;{}_A\|_{B\cup C}\; (Q \;{}_B\|_C\; R).$$

To prove this, we show by induction on the sum of the lengths $|e_p| + |e_q| + |e_r|$ that

$$(e_p \;{}_A\|_B\; e_q) \;{}_{A\cup B}\|_C\; e_r \subseteq e_p \;{}_A\|_{B\cup C}\; (e_q \;{}_B\|_C\; e_r). \tag{15}$$

For future reference, we note that the following identity holds:

$$\begin{aligned} S_{A\cup B,C}(S_{A,B}(\Sigma_p, \Sigma_q), \Sigma_r) &= (\Sigma_p \cap \Sigma_q \cap \Sigma_r) \cup (\Sigma_p \cap A) \cup (\Sigma_q \cap B) \cup (\Sigma_r \cap C) \\ &= S_{A,B\cup C}(\Sigma_p, S_{B,C}(\Sigma_q, \Sigma_r)) := \hat{\Sigma} \end{aligned} \tag{16}$$

If $e_p = \Sigma_p$, $e_q = \Sigma_q$, $e_r = \Sigma_r$ each have length 0, then $(e_p \;{}_A\|_B\; e_q) \;{}_{A\cup B}\|_C\; e_r$ consists of all subsets of $S_{A\cup B,C}(S_{A,B}(\Sigma_p, \Sigma_q), \Sigma_r)$ while $e_p \;{}_A\|_{B\cup C}\; (e_q \;{}_B\|_C\; e_r)$ consists of all subsets of $S_{A,B\cup C}(\Sigma_p, S_{B,C}(\Sigma_q, \Sigma_r))$. Thus, (15) follows from (16).

For the induction step, let $e_p = \bar{e}_p(\sigma_p, \Sigma_p)$, $e_q = \bar{e}_q(\sigma_q, \Sigma_q)$, $e_r = \bar{e}_r(\sigma_r, \Sigma_r)$. $\bar{e}_p$, $\bar{e}_q$, $\bar{e}_r$ are saturated since they are prefixes of saturated trajectories. Let $f \in e_p \;{}_A\|_B\; e_q$ and let $h \in f \;{}_{A\cup B}\|_C\; e_r \subseteq (e_p \;{}_A\|_B\; e_q) \;{}_{A\cup B}\|_C\; e_r$. Let $h = \bar{h}(\sigma, \hat{\Sigma})$. (There is no loss of generality in taking the final refusal set of h to be the maximal set $\hat{\Sigma}$.) To establish the induction step, we consider several cases. (Some cases will not apply if at least one of the trajectories e_p, e_q, e_r has length 0.)

(1) $\bar{h} \in f \;{}_{A\cup B}\|_C\; \bar{e}_r, \sigma \notin A \cup B$, $f(\sigma, \emptyset) \notin P \;{}_A\|_B\; Q$, $\sigma = \sigma_r$. (This is when the final event in h occurs in R but not in $P \;{}_A\|_B\; Q$.)

(2a) $\bar{h} \in \bar{f} \;{}_{A\cup B}\|_C\; e_r$, $\sigma \notin C$, $e_r(\sigma, \emptyset) \notin R$, $\sigma = \sigma_p$, $\bar{f} \in \bar{e}_p \;{}_A\|_B\; e_q$, $\sigma \notin B$, $e_q(\sigma, \emptyset) \notin Q$. (This is when the final event in h occurs in $P \;{}_A\|_B\; Q$ but not in R, and within $P \;{}_A\|_B\; Q$, it occurs in P but not in Q.)

(2b) $\bar{h} \in \bar{f} \;{}_{A\cup B}\|_C\; e_r$, $\sigma \notin C$, $e_r(\sigma, \emptyset) \notin R$, $\sigma = \sigma_q$, $\bar{f} \in e_p \;{}_A\|_B\; \bar{e}_q$, $\sigma \notin A$, $e_p(\sigma, \emptyset) \notin P$. (This is when the final event in h occurs in $P \;{}_A\|_B\; Q$ but not in R, and within $P \;{}_A\|_B\; Q$, it occurs in Q but not in P.)

(2c) $\bar{h} \in \bar{f} \;{}_{A\cup B}\|_C\; e_r$, $\sigma \notin C$, $e_r(\sigma, \emptyset) \notin R$, $\sigma = \sigma_p = \sigma_q$, $\bar{f} \in \bar{e}_p \;{}_A\|_B\; \bar{e}_q$. (This is when the final event in h occurs in $P \;{}_A\|_B\; Q$ but not in R, and within $P \;{}_A\|_B\; Q$, it occurs in both P and Q.)

(3a) $\bar{h} \in \bar{f}\ {}_{A\cup B}\|_C\ \bar{e}_r$, $\sigma = \sigma_p = \sigma_r$, $\bar{f} \in \bar{e}_p\ {}_A\|_B\ e_q$, $\sigma \notin B$, $e_q(\sigma, \emptyset) \notin Q$. (This is when the final event in h occurs in $P\ {}_A\|_B\ Q$ and in R, and within $P\ {}_A\|_B\ Q$, it occurs in P but not in Q.)

(3b) $\bar{h} \in \bar{f}\ {}_{A\cup B}\|_C\ \bar{e}_r$, $\sigma = \sigma_q = \sigma_r$, $\bar{f} \in e_p\ {}_A\|_B\ \bar{e}_q$, $\sigma \notin A$, $e_p(\sigma, \emptyset) \notin P$. (This is when the final event in h occurs in $P\ {}_A\|_B\ Q$ and in R, and within $P\ {}_A\|_B\ Q$, it occurs in Q but not in P.)

(3c) $\bar{h} \in \bar{f}\ {}_{A\cup B}\|_C\ \bar{e}_r$, $\sigma = \sigma_p = \sigma_q = \sigma_r$, $\bar{f} \in \bar{e}_p\ {}_A\|_B\ \bar{e}_q$. (This is when the final event in h occurs in $P\ {}_A\|_B\ Q$ and in R, and within $P\ {}_A\|_B\ Q$, it occurs in both P and Q.)

We include a detailed proof for case (2a). The other cases are proven in a similar manner and are left to the reader. Under the assumptions of (2a),

$$\bar{h} \in \bar{f}\ {}_{A\cup B}\|_C\ e_r \subseteq (\bar{e}_p\ {}_A\|_B\ e_q)\ {}_{A\cup B}\|_C\ e_r \subseteq \bar{e}_p\ {}_A\|_{B\cup C}\ (e_q\ {}_B\|_C\ e_r),$$

where the last inclusion is the induction hypothesis. Thus, there exists

$$g = \bar{g}(\sigma', S_{B,C}(\Sigma_q, \Sigma_r)) \in e_q\ {}_B\|_C\ e_r$$

with $\sigma' \in \{\sigma_q, \sigma_r\}$ such that $\bar{h} \in \bar{e}_p\ {}_A\|_{B\cup C}\ g$. (By Lemma 4, there is no loss of generality in taking the final refusal set of g to be the maximal set $S_{B,C}(\Sigma_q, \Sigma_r)$.) Since $e_q(\sigma, \emptyset) \notin Q$ and $e_r(\sigma, \emptyset) \notin R$, it follows from the assumption that e_q and e_r are saturated that $\sigma \in \Sigma_q$ and $\sigma \in \Sigma_r$. Thus, $\sigma \in \Sigma_q \cap \Sigma_r \subseteq S_{B,C}(\Sigma_q, \Sigma_r)$. By T5, $g(\sigma, \emptyset) \notin Q\ {}_B\|_C\ R$. This together with $\sigma \notin B \cup C$, implies that

$$h = \bar{h}(\sigma, \hat{\Sigma}) = \bar{h}(\sigma, S_{A,B\cup C}(\Sigma_p, S_{B,C}(\Sigma_q, \Sigma_r))) \in e_p\ {}_A\|_{B\cup C}\ g \subseteq e_p\ {}_A\|_{B\cup C}\ (e_q\ {}_B\|_C\ e_r),$$

completing the induction step. □

4.2 Monotonicity

Next, we show that prioritized synchronous composition is a monotone operator in each of its arguments.

Proposition 5 Let P, Q_1, Q_2 be trajectory models with $Q_1 \subseteq Q_2$. Then for any priority sets A, B

$$P\ {}_A\|_B\ Q_1 \subseteq P\ {}_A\|_B\ Q_2.$$

Proof: By Lemma 4, it suffices to show that for any $e_p \in P$ and *saturated* $e_q \in Q_1$, then

$$e_p\ {}_A\|^1_B\ e_q = e_p\ {}_A\|^2_B\ e_q,$$

where the left (respectively, right) side denotes the PSC of e_p, e_q in $P\ {}_A\|_B\ Q_1$ (respectively, $P\ {}_A\|_B\ Q_2$). We prove this by induction on the sum of the lengths $|e_p| + |e_q|$.

If $|e_p| + |e_q| = 0$, then $e_p = \Sigma_p$, $e_q = \Sigma_q$, in which case

$$e_p \;{}_A\|^1_B\; e_q = \{\Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)\} = e_p \;{}_A\|^2_B\; e_q.$$

Now suppose the assertion is true whenever the sum of the lengths is less than n, and suppose that $|e_p| + |e_q| = n$. Write $e_p = \bar{e}_p(\sigma_p, \Sigma_p)$, $e_q = \bar{e}_q(\sigma_q, \Sigma_q)$. For $i = 1, 2$ define the following subsets of $P \;{}_A\|_B\; Q_i$:

$$T_1^i := \begin{cases} \{\bar{e}(\sigma_p, \Sigma') |\ \bar{e} \in \bar{e}_p \;{}_A\|^i_B\; e_q,\ \Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)\} & \text{if } \sigma_p \notin B \text{ and } e_q(\sigma_p, \emptyset) \notin Q_i \\ \emptyset & \text{otherwise} \end{cases}$$

$$T_2^i := \begin{cases} \{\bar{e}(\sigma_q, \Sigma') |\ \bar{e} \in e_p \;{}_A\|^i_B\; \bar{e}_q,\ \Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)\} & \text{if } \sigma_q \notin A \text{ and } e_p(\sigma_q, \emptyset) \notin P \\ \emptyset & \text{otherwise} \end{cases}$$

$$T_3^i := \begin{cases} \{\bar{e}(\sigma, \Sigma') |\ \bar{e} \in \bar{e}_p \;{}_A\|^i_B\; \bar{e}_q,\ \Sigma' \subseteq S_{A,B}(\Sigma_p, \Sigma_q)\} & \text{if } \sigma_p = \sigma_q := \sigma \\ \emptyset & \text{otherwise} \end{cases}$$

Since the prefix of a saturated trajectory is saturated, $\bar{e}_q$ is Q_1-saturated. Thus, the induction hypothesis implies that

$$\bar{e}_p \;{}_A\|^1_B\; e_q = \bar{e}_p \;{}_A\|^2_B\; e_q, \quad e_p \;{}_A\|^1_B\; \bar{e}_q = e_p \;{}_A\|^2_B\; \bar{e}_q, \quad \bar{e}_p \;{}_A\|^1_B\; \bar{e}_q = \bar{e}_p \;{}_A\|^2_B\; \bar{e}_q.$$

It follows immediately from this that $T_2^1 = T_2^2$ and $T_3^1 = T_3^2$.

Since e_q is Q_1-saturated, it follows that if $e_q(\sigma_p, \emptyset) \notin Q_1$, then $\sigma_p \in \Sigma_q$. Since $e_q = \bar{e}_q(\sigma_q, \Sigma_q) \in Q_2$, and $\sigma_p \in \Sigma_q$, we obtain from the property T5 of trajectory models that $e_q(\sigma_p, \emptyset) \notin Q_2$. Also since $Q_1 \subseteq Q_2$, if $e_q(\sigma_p, \emptyset) \in Q_1$, then $e_q(\sigma_p, \emptyset) \in Q_2$. Thus, $e_q(\sigma_p, \emptyset) \in Q_1$ if and only if $e_q(\sigma_p, \emptyset) \in Q_2$. It follows that $T_1^1 = T_1^2$. Thus,

$$e_p \;{}_A\|^1_B\; e_q = \cup_{j=1}^3 T_j^1 = \cup_{j=1}^3 T_j^2 = e_p \;{}_A\|^2_B\; e_q,$$

completing the induction step. □

Remark 9 Let P, Q_1, Q_2 be trajectory models and let A, B be arbitrary priority sets. It follows trivially from Proposition 5 that if $Q_1 \subseteq Q_2$, then

$$L(P \;{}_A\|_B\; Q_1) \subseteq L(P \;{}_A\|_B\; Q_2). \tag{17}$$

However, it is important to note that (17) is not true under the weaker assumption that $L(Q_1) \subseteq L(Q_2)$. Were this to be true, it would imply that if $L(Q_1) = L(Q_2)$, then $L(P \;{}_A\|_B\; Q_1) = L(P \;{}_A\|_B\; Q_2)$. However, it is shown in [5, Example 7] that two systems with the same failures model (and hence the same language model) may yield different languages when composed in prioritized synchrony with a fixed system.

Corollary 3 Let P, Q_1, Q_2 be trajectory models with Q_1 deterministic and $L(Q_1) = L(Q_2)$. Then

$$P \;{}_A\|_B\; Q_1 \subseteq P \;{}_A\|_B\; Q_2.$$

Proof: It follows from Proposition 3 that $Q_1 \subseteq Q_2$, so the result is an immediate consequence of Proposition 5. □

4.3 Augmentation and Prioritized Synchronous Composition

We define augmentation of both NSM's and trajectory models, and show that the prioritized synchronous composition of two trajectory models is identical to *strict* synchronous composition of their augmentations, provided the two priority sets exhaust the set of events.

Let $\mathcal{P}$ be an NSM with alphabet Σ, and let $D \subseteq \Sigma$. We denote by $\mathcal{S}(D)$ the deterministic state machine with one state and self-loops labeled by every event in D. The *augmentation of $\mathcal{P}$ by D*, denoted $\mathcal{P}^D$, is defined to be the NSM

$$\mathcal{P}^D := \mathcal{P} \,{}_{\emptyset}\|_{\emptyset}\, \mathcal{S}(D)$$

The state space of $\mathcal{P}^D$ can be identified with the state space of $\mathcal{P}$, and $\mathcal{P}^D$ is then obtained from $\mathcal{P}$ by adding self-loops at each $x \in X_{\mathcal{P}}$ labeled by every event in $D \cap \Re_{\mathcal{P}}(x)$. It is clear that $\mathcal{P}^D$ is deterministic whenever $\mathcal{P}$ is deterministic.

If P is a trajectory model, the *augmentation of P by D*, denoted P^D, is defined to be the trajectory model

$$P^D := P \,{}_{\emptyset}\|_{\emptyset}\, det(D^*).$$

Note that since both priority sets are empty, P^D represents pure interleaving of P and $det(D^*)$ except that the broadcast synchronization requirement means that events in D which can also occur in P occur synchronously in both P and $det(D^*)$.

We will need the following result which shows that PSC of trajectory models preserves determinism.

Proposition 6 If P and Q are deterministic trajectory models, then so is $P \,{}_{A}\|_{B}\, Q$.

Proof: By definition, there exist deterministic state machines $\mathcal{P}, \mathcal{Q}$ such that $\pi_t(\mathcal{P}) = P$, $\pi_t(\mathcal{Q}) = Q$. From Definition 9, it is clear that $\mathcal{P} \,{}_{A}\|_{B}\, \mathcal{Q}$ is deterministic. Since Theorem 2 implies that $P \,{}_{A}\|_{B}\, Q = \pi_t(\mathcal{P} \,{}_{A}\|_{B}\, \mathcal{Q})$, we conclude that $P \,{}_{A}\|_{B}\, Q$ is deterministic. □

Corollary 4 If P is a deterministic trajectory model, then so is its augmentation P^D.

The next result describes the relationship between the augmentation of an NSM and the augmentation of its trajectory model.

Proposition 7 Given a NSM $\mathcal{P}$ and an event set $D \subseteq \Sigma$, $\pi_t(\mathcal{P}^D) = [\pi_t(\mathcal{P})]^D$.

Proof: Immediate consequence of Theorem 2. □

Remark 10 Since $det(D^*)$ can always execute every event in D and can never execute any event in $\Sigma - D$, it follows that

$$P^D = P \,{}_{\emptyset}\|_{\emptyset}\, det(D^*) = P \,{}_{\emptyset}\|_{D}\, det(D^*) = P \,{}_{\Sigma-D}\|_{D}\, det(D^*).$$

The next result shows that augmentation can be used to reduce prioritized synchronous composition to strict synchronization.

Proposition 8 If $A \cup B = \Sigma$, then

$$P \;{}_A\|_B\; Q = P^{B-A} \;{}_\Sigma\|_B\; Q = P^{B-A} \;{}_\Sigma\|_\Sigma\; Q^{A-B}.$$

Proof: It suffices to prove the first equality since the second equality follows from symmetry and a second application of the first equality. Using Remark 10 and Proposition 4 gives

$$\begin{aligned} P^{B-A} \;{}_\Sigma\|_B\; Q &= (P \;{}_A\|_{B-A}\; det((B-A)^*)) \;{}_\Sigma\|_B\; Q \\ &= det((B-A)^*) \;{}_{B-A}\|_\Sigma\; (P \;{}_A\|_B\; Q) \\ &= P \;{}_A\|_B\; Q. \end{aligned}$$

The final equality is an easy consequence of two facts: The priority set of $P \;{}_A\|_B\; Q$ is Σ, so the process $det((B-A)^*)$ cannot execute any events which do not occur in $P \;{}_A\|_B\; Q$. The process $det((B-A)^*)$ can alway execute each event in its priority set, so it cannot block any events in $P \;{}_A\|_B\; Q$. □

4.4 Idempotency

If $\mathcal{P}, \mathcal{Q}$ are NSM's, the language model of the strict synchronous composition (SSC) is the intersection of the language models of $\mathcal{P}$ and $\mathcal{Q}$ (refer to Remark 7 for explanation), i.e.

$$\pi_l(\mathcal{P} \;{}_\Sigma\|_\Sigma\; \mathcal{Q}) = \pi_l(\mathcal{P}) \cap \pi_l(\mathcal{Q}).$$

It follows from this that the SSC of language models is idempotent:

$$\pi_l(\mathcal{P} \;{}_\Sigma\|_\Sigma\; \mathcal{P}) = \pi_l(\mathcal{P}).$$

In contrast to the situation for language models, the SSC of trajectory models is not idempotent. This is illustrated by the following example.

Example 3 Let $\Sigma = \{a, b\}$, $A = B = \Sigma$, $P = (a \to \Delta_\Sigma) \oplus (b \to \Delta_\Sigma)$. P is generated by the maximal trajectories $\{b\}(a, \{a, b\})$ and $\{a\}(b, \{a, b\})$. However, $P \;{}_\Sigma\|_\Sigma\; P$ is generated by the trajectories

$$\{b\}(a, \{a, b\}),\ \{a\}(b, \{a, b\}),\ \{a, b\}.$$

Although $L(P \;{}_\Sigma\|_\Sigma\; P) = L(P)$, the trajectory models $P \;{}_\Sigma\|_\Sigma\; P$ and P are not the same.

The next example shows that when the priority sets A, B are not both equal to Σ, even the language model of $P \;{}_A\|_B\; P$ can differ from that of P.

Example 4 Let Σ and P be as in Example 3, but let $A = B = \emptyset$. $P \;{}_\emptyset\|_\emptyset\; P$ is generated by the trajectories

$$\{b\}(a, \{a, b\}),\ \{a\}(b, \{a, b\}),\ \emptyset(a, \{a\})(b, \{a, b\}),\ \emptyset(b, \{b\})(a, \{a, b\}).$$

Thus, $P \neq P \;{}_\emptyset\|_\emptyset\; P$. In fact, $L(P) \neq L(P \;{}_\emptyset\|_\emptyset\; P)$.

It has been suggested elsewhere that if $A \subseteq B$, then

$$L(P \,{}_A\|_B\, Q) = L((P \,{}_A\|_B\, Q) \,{}_A\|_B\, Q). \tag{18}$$

However, this is not the case as demonstrated by the next example.

Example 5 Let $\Sigma = \{a, b\}$, $A = B = \emptyset$, $P = \Delta_\Sigma$, $Q = (a \rightarrow \Delta_\Sigma) \oplus (b \rightarrow \Delta_\Sigma)$. Then $P \,{}_A\|_B\, Q = Q$, so

$$L((P \,{}_A\|_B\, Q) \,{}_A\|_B\, Q) = L(Q \,{}_\emptyset\|_\emptyset\, Q) = \overline{\{ab, ba\}}.$$

But

$$L(P \,{}_A\|_B\, Q) = L(Q) = \overline{\{a, b\}}.$$

(Overbar denotes prefix-closure.)

Based on the connection between PSC and SSC of augmented systems when $A \cup B = \Sigma$, (18) might be expected to hold under the alternative assumption that $A \cup B = \Sigma$. However, this also turns out to be false as demonstrated by the following example.

Example 6 Let $\Sigma = \{a, b\}$, $A = \{a\}$, $B = \{b\}$, $P = (a \rightarrow \Delta_\Sigma) + (b \rightarrow \Delta_\Sigma)$, $Q = \Delta_\Sigma \oplus (b \rightarrow \Delta_\Sigma)$. Straightforward calculation shows that $P \,{}_A\|_B\, Q = (a \rightarrow \Delta_\Sigma) \oplus ((b \rightarrow \Delta_\Sigma) + (a \rightarrow (b \rightarrow \Delta_\Sigma)))$, while $(P \,{}_A\|_B\, Q) \,{}_A\|_B\, Q = [a \rightarrow \Delta_\Sigma] \oplus [(a \rightarrow (b \rightarrow \Delta_\Sigma)) + (b \rightarrow (a \rightarrow \Delta_\Sigma))] \oplus [(a \rightarrow (b \rightarrow \Delta_\Sigma)) + (b \rightarrow \Delta_\Sigma)]$. Thus, $L(P \,{}_A\|_B\, Q) = \overline{\{b, ab\}}$, while $L((P \,{}_A\|_B\, Q) \,{}_A\|_B\, Q) = \overline{\{ab, ba\}}$.

5 Supervisory Control with Driven Events

In this section, we derive results concerning supervisory control by prioritized synchronous composition in the presence of driven events.

5.1 Supervisory Control Problem

We begin with a result which shows that in a prioritized synchronous composition, a deterministic process participates in every event of any trajectory whose trace belongs to its language.

Lemma 5 Let P, Q be trajectory models with Q deterministic. If $e \in e_p \,{}_A\|_B\, e_q \subseteq P \,{}_A\|_B\, Q$ with $tr(e) \in L(Q)$, then $tr(e) = tr(e_q)$.

Proof: The result follows as a special case of Lemma 6 below. □

The following result gives necessary and sufficient conditions for a given (prefix-closed) language to be realizable as the closed-loop language for a plant supervised by prioritized synchronous composition. The basic assumption is that every event in the alphabet Σ belongs to the priority set A of the plant P or the priority set B of the supervisor. The interpretation is that Σ is partitioned into disjoint subsets Σ_c, Σ_u and Σ_d consisting of the controllable, uncontrollable and driven events, and $A = \Sigma_c \cup \Sigma_u$ while $B = \Sigma_c \cup \Sigma_d$.

Theorem 3 Let P be a trajectory model, $A \cup B = \Sigma$, and let K be a nonempty prefix-closed sublanguage of $L(P^{B-A})$. Then there exists a trajectory model S such that $L(P\ {}_A\|_B\ S) = K$ if and only if

$$K(A-B) \cap L(P^{B-A}) \subseteq K, \tag{19}$$

in which case S can be chosen to be the deterministic trajectory model $det(K)$.

Proof: We begin with sufficiency. Suppose that equation (19) holds. Since K is a nonempty prefix-closed sublanguage of $L(P^{B-A})$, there exists a trajectory model S such that

$$L(P^{B-A}) \cap L(S) = K.$$

Without loss of generality, we may assume that S is deterministic. (In particular, we can choose $S = det(K)$.)

We claim that

$$L(P^{B-A}) \cap L(S^{A-B}) = K. \tag{20}$$

Obviously,

$$K = L(P^{B-A}) \cap L(S) \subseteq L(P^{B-A}) \cap L(S^{A-B}).$$

We establish the reverse inclusion by contradiction. Suppose $L(P^{B-A}) \cap L(S^{A-B})$ strictly contains K. Let $t = s\sigma$ be a minimal length trace in $L(P^{B-A}) \cap L(S^{A-B}) - K$. Then $s \in K = L(P^{B-A}) \cap L(S)$. Since $s\sigma \in L(S^{A-B})$, there exists $g = \bar{g}(\sigma, \emptyset) \in S^{A-B}$ such that $tr(\bar{g}) = s$. Hence, there exist $e = \bar{e}(\sigma', \Sigma') \in S$, $f = \bar{f}(\sigma'', \Sigma'') \in det((A-B)^*)$ such that $g \in e\ {}_\emptyset\|_\emptyset\ f$. First suppose $\sigma \notin A - B$. Then $\sigma \neq \sigma''$, so $\sigma = \sigma'$ and $\bar{g} \in \bar{e}\ {}_\emptyset\|_\emptyset\ f$. Since S is deterministic and $tr(\bar{g}) = s \in L(S)$, it follows from Lemma 5 that $tr(\bar{g}) = tr(\bar{e})$. Thus, $s\sigma = tr(g) = tr(e) \in L(S)$, which implies that $t \in K$, contradiction. On the other hand, if $\sigma \in A - B$, then it follows from (19) that $t \in K$, again a contradiction. Thus, (20) holds.

Using Proposition 8, it follows that

$$K = L(P^{B-A}) \cap L(S^{A-B}) = L(P^{B-A}\ {}_\Sigma\|_\Sigma\ S^{A-B}) = L(P\ {}_A\|_B\ S),$$

showing that S solves the supervisory control problem.

Conversely, suppose there exists a trajectory model S such that

$$L(P\ {}_A\|_B\ S) = K.$$

Then (20) holds. Let $t = s\sigma \in K(A-B) \cap L(P^{B-A})$. Since $s \in K \subseteq L(S^{A-B})$ and $\sigma \in A-B$, it follows that $s\sigma \in L(S^{A-B})$. Thus, $t \in L(P^{B-A}) \cap L(S^{A-B}) = K$, so (19) holds. □

Remark 11 Theorem 3 states that K is realizable as the closed-loop language if and only if it is controllable (in the sense of Ramadge-Wonham [18]) *with respect to the language of the augmented plant.* $L(P^{B-A})$ depends on the trajectory model P–not simply on $L(P)$. Knowledge of $L(P)$ is not sufficient to determine if the supervisory control problem is solvable for a given target language K. This is illustrated by the following example.

Example 7 We consider a very simple air traffic control problem. The plant represents the aircraft and pilot while the supervisor represents the air traffic controller. Let $\Sigma = \{a, b\}$ where $a \in \Sigma_u$ represents a flight maneuver, while $b \in \Sigma_d$ represents a command from the tower not to execute the flight maneuver. The execution of b by the supervisor indicates that the command has been broadcast, whereas the execution of b by the plant indicates that the command has been received.

We consider two alternative trajectory models for the plant:

$$\begin{aligned} P_1 &= (a \rightarrow \Delta_\Sigma) + (b \rightarrow \Delta_\Sigma) \\ P_2 &= [(a \rightarrow \Delta_\Sigma) + (b \rightarrow \Delta_\Sigma)] \oplus (a \rightarrow \Delta_\Sigma) \end{aligned}$$

In P_1, the pilot can initially execute the maneuver or receive the command not to do so. However, in P_2, there is an initial nondeterministic choice between P_1 and the trajectory model $(a \rightarrow \Delta_\Sigma)$ in which the maneuver is possible but the command cannot be received. Thus, P_2 models the possibility of aircraft radio receiver failure. Note that $L(P_1) = L(P_2)$. However, it can be verified that $L(P_1^{B-A}) = (a + \epsilon)b^*$ while $L(P_2^{B-A}) = b^*(a + \epsilon)b^*$.

Suppose that the target language K is not completely specified but is required to contain the trace b and not contain any trace in which the event a occurs after the event b has occurred. In other words, the tower should be initially able to broadcast the command b, and if the command has been broadcast, the pilot must not be able to execute the maneuver a.

The supervisory control problem is clearly solvable for the plant model P_1. For example, if we choose $S = P_1$, then $P_1 \,{}_A\|_B\, S = P_1$, so the closed-loop language is $L(P_1) = \{\epsilon, a, b\}$, which meets the specifications for K. On the other hand, the supervisory control problem is not solvable for the plant model P_2. For any target language K which satisfies the specifications, we have $ba \in K(A - B) \cap L(P_2^{B-A}) - K$. It follows from Theorem 3 that there is no supervisor S such that $L(P_2 \,{}_A\|_B\, S) = K$.

It is worth noting that if P_2 is the correct plant model–i.e., receiver failure can occur–then the supervisory control problem can be made solvable by changing the protocol between the pilot and tower. If the pilot is required to obtain clearance from the tower in order to execute the maneuver a, then a becomes a controllable event and it is then trivial to construct a supervisor which meets the specifications.

Remark 12 If K does not satisfy the condition (19)–i.e., is not controllable relative to the language of the augmented plant–then a supervisor S can be constructed which imposes $K^\uparrow$ as the closed-loop language. Here $K^\uparrow$ is the supremal sublanguage of K which is controllable with respect to $L(P^{B-A})$. $K^\uparrow$ can be computed by applying any of the standard algorithms [19, 2, 11], but using $L(P^{B-A})$ in place of $L(P)$.

Remark 13 The proof of Theorem 3 shows that if K satisfies the condition (19) and if N is any prefix-closed sublanguage of Σ^* with

$$L(P^{B-A}) \cap N = K,$$

then the deterministic supervisor $S := det(N)$ results in K as the closed-loop language $L(P\ {}_A\|_B\ S)$. Since $K \subseteq N$, it follows from Lemma 5 that every event executed by the closed-loop system occurs in S. In particular, every uncontrollable event is executed by the supervisor even though such events do not belong to its priority set. This behavior is induced by the broadcast synchronization requirement in prioritized synchronous composition.

When there are no driven events, then $A = \Sigma_c \cup \Sigma_u = \Sigma$ and $B = \Sigma_c$. In this case Theorem 3 specializes to give the following two corollaries.

Corollary 5 Let K be a nonempty prefix-closed sublanguage of $L(P)$. Then there exists a trajectory model S such that $L(P\ {}_\Sigma\|_{\Sigma_c}\ S) = K$ if and only if

$$L(P\ {}_\Sigma\|_{\Sigma_c}\ det(K)) = K.$$

Corollary 6 Let K be a nonempty prefix-closed sublanguage of $L(P)$. Then there exists a trajectory model S such that $L(P\ {}_\Sigma\|_{\Sigma_c}\ S) = K$ if and only if

$$K\Sigma_u \cap L(P) \subseteq K. \tag{21}$$

Remark 14 Corollary 6 shows that when there are no driven events, the necessary and sufficient conditions for supervisory control by prioritized synchronous composition are the same as those in the Ramadge-Wonham framework [17]. Corollary 6 was stated without proof in [5, Theorem 1] and [6, Theorem 14.2]. Corollary 5 was stated in [6, Theorem 14.1] accompanied by an incomplete proof. (Compare [6, Propositions 13.12, 13.13 and Corollary 13.3], on which the given proof of [6, Theorem 14.1] is based, with Remark 9 and Example 5 above.)

Remark 15 It is interesting to specialize Remark 13 to the case where there are no driven events. It follows from Remark 13 that if K satisfies the condition (21) and if N is any prefix-closed sublanguage of Σ^* with

$$L(P) \cap N = K,$$

then the supervisor $S := det(N)$ imposes K as the closed-loop language and participates in every event executed by the closed-loop system. Since $A = \Sigma$, the plant also participates in every event. Thus, the plant and supervisor function as though they are connected by *strict synchronization* rather than by prioritized synchronous composition. In particular, this is the case when the supervisor is chosen to be $det(K)$. The determinism of S is essential here. If S is a nondeterministic trajectory model with $L(S) = N$, there is no guarantee that the closed-loop language will be K. This is demonstrated by the next example.

Example 8 Let $\Sigma = \{a, b\}$, $\Sigma_c = \{a\}$, $\Sigma_u = \{b\}$, $P = (a \rightarrow \Delta_\Sigma) + (b \rightarrow (a \rightarrow \Delta_\Sigma))$, $S = (a \rightarrow \Delta_\Sigma) \oplus (b \rightarrow \Delta_\Sigma)$. Then

$$L(P) = \{\epsilon, a, b, ba\}, \quad L(S) = \{\epsilon, a, b\}.$$

Let $K = L(S)$. Then K satisfies the controllability condition (21) as well as $L(P) \cap L(S) = K$. A straightforward calculation shows that

$$P \,{}_{\Sigma}\|_{\Sigma_c}\, S = ((a \to \Delta_\Sigma) + (b \to (a \to \Delta_\Sigma)) \oplus (b \to \Delta_\Sigma).$$

Thus,

$$L(P \,{}_{\Sigma}\|_{\Sigma_c}\, S) = \{\epsilon, a, b, ba\} = L(P) \neq K.$$

What happens is that since S is nondeterministic, the event b can be executed as the initial event solely in P even though $b \in L(S)$. (This cannot happen for deterministic S by Lemma 5.) Thus, strict synchronization is lost. This permits a trace of $P \,{}_{\Sigma}\|_{\Sigma_c}\, S$ which is not a trace of S.

5.2 Restricted Supervisory Control Problem

We continue to assume that $A \cup B = \Sigma$ where $A = \Sigma_c \cup \Sigma_u$ and $B = \Sigma_c \cup \Sigma_d$. In the closed-loop system $P \,{}_{A}\|_{B}\, S$, the events in $A - B$–i.e., the uncontrollable events–are generated by the plant P and are broadcast to the supervisor S where they are synchronously executed whenever enabled. It may happen that information about the occurrence of certain uncontrollable events is unavailable for broadcast due to lack of sensors, or it may be desired to implement a simplified supervisor which ignores such information. This suggests a generalization of prioritized synchronous composition in which the broadcast synchronization requirement is disregarded for a specified subset $\Gamma \subseteq A - B$ of uncontrollable events. Since events in $A - B$ cannot occur spontaneously in S, this effectively prevents S from ever executing the events in Γ. Thus, instead of modifying the definition of prioritized synchronous composition, it is equivalent to restrict the admissible supervisors to those which do not execute events in Γ.

Let Π_Γ denote the natural projection defined by

$$\Pi_\Gamma(\sigma) = \begin{cases} \epsilon & \text{for } \sigma \in \Gamma \\ \sigma & \text{for } \sigma \in \Sigma - \Gamma \end{cases}$$

Π_Γ extends to a map on Σ^* in the obvious way. We define the *Restricted Supervisory Control Problem (RSCP)* to be as follows: Given a prefix-closed sublanguage K of $L(P^{B-A})$ and $\Gamma \subseteq A - B$, determine if there exists a supervisor S such that

- $L(P \,{}_{A}\|_{B}\, S) = K$
- $\Pi_\Gamma(L(S)) = L(S)$

Remark 16 There are two different ways to model an uncontrollable event in the plant which is unobservable to the supervisor. It can be completely suppressed and treated as an ϵ-event in P. Alternatively, it can be treated as a labeled event $\sigma \in \Sigma$ in the plant which does not label any transitions in the supervisor. The advantage of the second approach (which is the one taken in the RSCP) is that such an event can be included in the performance

specifications–i.e., in the target language K. Hence, even though it is unobservable to the supervisor, its occurrence in the closed-loop system can be controlled–albeit subject to the conditions which must be satisfied by K for the solvability of the RSCP.

The next result generalizes Lemma 5 to the case where certain events in $A - B$ are not present in the second process Q.

Lemma 6 Let $\Gamma \subseteq A - B$ and let Π_Γ be the natural projection. Let P, Q be trajectory models with Q deterministic and satisfying $\Pi_\Gamma(L(Q)) = L(Q)$. If $e \in e_p \ {}_A\|_B \ e_q \subseteq P \ {}_A\|_B \ Q$ with $\Pi_\Gamma(tr(e)) \in L(Q)$, then $\Pi_\Gamma(tr(e)) = tr(e_q)$.

Proof: The proof is by induction on $|e|$. The assertion holds trivially when $|e| = 0$. For the induction step, write $e = \bar{e}(\sigma, \Sigma')$ and let $\bar{e}_p$, $\bar{e}_q$ denote the prefixes of e_p, e_q obtained by deleting the final event and refusal set from each trajectory.

If σ occurs synchronously in both P and Q, then $\bar{e} \in \bar{e}_p \ {}_A\|_B \ \bar{e}_q$. Then $\sigma \notin \Gamma$, so

$$\Pi_\Gamma(tr(e)) = \Pi_\Gamma(tr(\bar{e}))\sigma.$$

Since $L(Q)$ is prefix-closed, $\Pi_\Gamma(tr(\bar{e})) \in L(Q)$. Applying the induction hypothesis gives $\Pi_\Gamma(tr(\bar{e})) = tr(\bar{e}_q)$. Thus,

$$\Pi_\Gamma(tr(e)) = \Pi_\Gamma(tr(\bar{e}))\sigma = tr(\bar{e}_q)\sigma = tr(e_q).$$

The same argument applies in the case where $\bar{e} \in e_p \ {}_A\|_B \ \bar{e}_q$–i.e., when σ occurs only in Q.

Suppose $\bar{e} \in \bar{e}_p \ {}_A\|_B \ e_q$. I.e., σ occurs only in P. If $\sigma \in \Gamma$, then

$$\Pi_\Gamma(tr(e)) = \Pi_\Gamma(tr(\bar{e})) = tr(e_q),$$

where the second equality follows from the induction hypothesis. Now suppose that $\sigma \notin \Gamma$. Since σ occurs only in P, it follows that $e_q(\sigma, \emptyset) \notin Q$. Since Q is deterministic, Proposition 3 then implies that $tr(e_q)\sigma \notin L(Q)$. Since $L(Q)$ is prefix-closed, $\Pi_\Gamma(tr(\bar{e})) \in L(Q)$. Using the induction hypothesis, we have

$$tr(e_q)\sigma = \Pi_\Gamma(tr(\bar{e}))\sigma = \Pi_\Gamma(tr(e)) \in L(Q),$$

a contradiction. Thus, this final case cannot occur. □

For the standard supervisory control problem with partial observations (and no driven events), a target language K is obtainable as the language of the closed-loop system if and only if K is controllable and observable relative to the language of the plant [15, 3]. The following result shows that in the presence of driven events, the RSCP is solvable if and only if K is controllable and observable *relative to the language of the augmented plant.*

Theorem 4 Let $A \cup B = \Sigma$, $\Gamma \subseteq A - B$, and let K be a nonempty prefix-closed sublanguage of $L(P^{B-A})$. Then there exists a trajectory model S such that

$$L(P \ {}_A\|_B \ S) = K, \qquad \Pi_\Gamma(L(S)) = L(S) \tag{22}$$

if and only if the following two conditions are satisfied:

$$(a) \quad K(A-B) \cap L(P^{B-A}) \subseteq K \tag{23}$$

$$(b) \quad \text{If } \bar{s},\ \bar{t} \in K \text{ with } \Pi_\Gamma(\bar{s}) = \Pi_\Gamma(\bar{t}), \text{ and if } \bar{s}\sigma \in K,\ \bar{t}\sigma \in L(P^{B-A}), \text{ then } \bar{t}\sigma \in K. \tag{24}$$

In this case, S can be chosen to be the process $det(\Pi_\Gamma(K))$.

Proof: We first show the necessity of the controllability condition (23) and observability condition (24). Suppose there exists a trajectory model S such that (22) is satisfied. Then (23) follows from Theorem 3. Let $\bar{s},\ \bar{t} \in K$ with $\Pi_\Gamma(\bar{s}) = \Pi_\Gamma(\bar{t})$, and suppose that $\bar{s}\sigma \in K,\ \bar{t}\sigma \in L(P^{B-A})$. We need to show that $\bar{t}\sigma \in K$. Since

$$L(P^{B-A}) \cap L(S^{A-B}) = K,$$

it suffices to show that $\bar{t}\sigma \in L(S^{A-B})$. Since K is controllable, it suffices to consider $\sigma \in B$. Note that

$$S^{A-B} = (S^{A-B-\Gamma})^{\Gamma}.$$

Also, since $\Pi_\Gamma(L(S)) = L(S)$, events in Γ are never executed in S. Hence

$$\Pi_\Gamma(L(S^{A-B-\Gamma})) = L(S^{A-B-\Gamma}).$$

In other words, events in Γ are never executed in $S^{A-B-\Gamma}$. Hence the language $L(S^{A-B}) = L((S^{A-B-\Gamma})^{\Gamma})$ is obtained by pure interleaving of the languages $L(S^{A-B-\Gamma})$ and $L(\det(\Gamma^*)) = \Gamma^*$. Since the string $\bar{s}\sigma \in L(S^{A-B})$, we have $\Pi_\Gamma(\bar{s}\sigma) \in L(S^{A-B-\Gamma})$. Also, since $\Pi_\Gamma(\bar{s}\sigma) = \Pi_\Gamma(\bar{s})\sigma = \Pi_\Gamma(\bar{t})\sigma = \Pi_\Gamma(\bar{t}\sigma)$, we have $\Pi_\Gamma(\bar{t}\sigma) \in L(S^{A-B-\Gamma})$. Since $\bar{t}\sigma$ is a pure interleaving of $\Pi_\Gamma(\bar{t}\sigma) \in L(S^{A-B-\Gamma})$ and a string in Γ^*, $\bar{t}\sigma \in L(S^{A-B})$. This establishes (24) and completes the proof of necessity.

To prove sufficiency, suppose that (23) and (24) both hold. Let $S = det(\Pi_\Gamma(K))$. By Proposition 8, it is equivalent to prove

$$L(P^{B-A}\ {}_{\Sigma}\|_{B}\ S) = K.$$

Given any $t \in K$, there exists $e \in P^{B-A}$ with $tr(e) = t$ and there exists $f \in S$ with $tr(f) = \Pi_\Gamma(t)$. Since $\Gamma \cap B = \emptyset$ and S can never execute an event in Γ, it follows that $e\ {}_{\Sigma}\|_{B}\ f$ is nonempty and every trajectory which it contains has trace t. Thus,

$$K \subseteq L(P^{B-A}\ {}_{\Sigma}\|_{B}\ S).$$

It remains to prove

$$L(P^{B-A}\ {}_{\Sigma}\|_{B}\ S) \subseteq K. \tag{25}$$

We establish (25) by contradiction. Let $g = \bar{g}(\sigma, \Sigma') \in P^{B-A}\ {}_{\Sigma}\|_{B}\ S$ and suppose g has minimal length among the trajectories of $P^{B-A}\ {}_{\Sigma}\|_{B}\ S$ whose traces are not in K. Let $\bar{t}$ and $t = \bar{t}\sigma$ denote the traces of $\bar{g}$ and g respectively. Then $t \notin K$ and

$$\bar{t} \in K, \quad t \in L(P^{B-A}), \tag{26}$$

where the final membership follows from the fact that the priority set of P^{B-A} is Σ.

If $\sigma \in A - B$, then it follows from (23) that $t \in K$, contrary to assumption. Thus, without loss of generality, we may assume that $\sigma \in B$. Since

$$\bar{r} := \Pi_\Gamma(\bar{t}) \in \Pi_\Gamma(K) = L(S),$$

it follows from Lemma 6 that S executes every event in $\bar{r}$ while $P^{B-A}\ {}_\Sigma\|_B\ S$ executes the trajectory $\bar{g}$. Since $\sigma \in B$, the final event in g must occur synchronously in P^{B-A} and S. This implies that

$$\bar{r}\sigma \in L(S) = \Pi_\Gamma(K).$$

Thus, there exists $s \in K$ such that $\Pi_\Gamma(s) = \bar{r}\sigma$. Since the last observable event in s is σ, by replacing s with a prefix if necessary, we may assume that $s = \bar{s}\sigma$. Then

$$\bar{s} \in K, \quad \Pi_\Gamma(\bar{s}) = \bar{r} = \Pi_\Gamma(\bar{t}) \tag{27}$$

It follows from (26), (27) and the observability assumption that $t \in K$, contrary to assumption. This establishes (25) and completes the proof of sufficiency. □

Remark 17 It follows from Lemma 6 that if $S := det(\Pi_\Gamma(K))$ is used to solve the Restricted Supervisory Control Problem, then every event in $\Sigma - \Gamma$ which occurs in the closed-loop system is executed by the supervisor. The events in Γ are not observed by the supervisor and are executed only by the plant.

6 Conclusion

In this paper we have studied the supervisory control of nondeterministic plants in the presence of driven events under complete as well as partial observation. We have shown that prioritized synchronous composition is an adequate control mechanism for this purpose. The trajectory model, used for describing the behavior of nondeterministic systems, is shown to be a language congruence with respect to prioritized synchronous composition. Hence it is quite useful for describing the behaviors of nondeterministic systems which may be controlled via PSC. It is shown that the supervisory control problem with driven events is solvable if and only if the target language is controllable and observable with respect to the language of the plant augmented by the set of driven events. Due to the augmentation, the solvability depends on the trajectory model of the plant–not simply on its language. We have also described some of the basic properties–associativity, monotonicity, augmentation, idempotency, etc.–of PSC, which are useful in the analysis of supervisory control.

References

[1] S. Balemi, G. J. Hoffmann, P. Gyugyi, H. W. Toi, and G. F. Franklin. Supervisory control of a rapid thermal multiprocessor. Technical Report ISL/GFF/91-1, Department of Electrical Engineering, Stanford University, Stanford, CA 94305, November 1991.

[2] R. D. Brandt, V. K. Garg, R. Kumar, F. Lin, S. I. Marcus, and W. M. Wonham. Formulas for calculating supremal and normal sublanguages. *Systems and Control Letters*, 15(8):111-117, 1990.

[3] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya. Supervisory control of discrete event processes with partial observations. *IEEE Transactions on Automatic Control*, 33(3):249-260, 1988.

[4] C. H. Golaszewiski and P. J. Ramadge. Control of discrete event processes with forced events. In *Proceedings of 26th IEEE Conference on Decision and Control*, pages 247–251, Los Angeles, CA, 1987.

[5] M. Heymann. Concurrency and discrete event control. *IEEE Control Systems Magazine*, 10(4):103–112, 1990.

[6] M. Heymann and G. Meyer. An algebra of discrete event processes. Technical Report NASA 102848, NASA Ames Research Center, Moffett Field, CA, June 1991.

[7] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice Hall, Inc., Englewood Cliffs, NJ, 1976.

[8] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, Reading, MA, 1979.

[9] K. M. Inan and P. P. Varaiya. Algebras of discrete event models. *Proceedings of the IEEE*, 77(1):24-38, 1989.

[10] K. M. Inan and P. P. Varaiya. Finitely recursive process models for discrete event systems. *IEEE Transactions on Automatic Control*, 33(7):626-639, 1988.

[11] R. Kumar, V. K. Garg, and S. I. Marcus. On controllability and normality of discrete event dynamical systems. *Systems and Control Letters*, 17(3):157–168, 1991.

[12] R. Kumar, V. K. Garg, and S. I. Marcus. On supervisory control of sequential behaviors. *IEEE Transactions on Automatic Control*, 1992. To appear.

[13] R. Kumar, V. K. Garg, and S. I. Marcus. Predicates and predicate transformers for supervisory control of discrete event systems. *IEEE Transactions on Automatic Control*, 1992. To appear.

[14] R. Kumar, V. K. Garg, and S. I. Marcus. Stability and stabilizability of behavior of discrete event systems. *SIAM Journal of Control and Optimization*, 1992. To appear.

[15] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44(3):173-198, 1988.

[16] R. Milner. *A Calculus of Communicating Systems.* Springer Verlag, Berlin, 1980.

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

[18] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.

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