CHALMERS

Supervisory Control of Extended Finite Automata

Using Transition Projection Abstraction

Technical report

Mohammad Reza Shoaei, Lei Feng,
Bengt Lennartson

Department of Signals and systems
CHALMERS UNIVERSITY OF TECHNOLOGY R004/2012
Göteborg, Sweden ISSN 1403-266x

Abstract

An abstraction method for Extended Finite Automata (EFA), i.e., finite automata extended with variables, using Transition Projection is presented in this work. A DES modeled by EFA is abstracted into subsystems that embody internal interacting dependencies. Synthesis and verification of subsystems are achieved through their model abstractions rather than their global model. Sufficient conditions are presented to guarantee that supervisors, result in maximally permissive and nonblocking control. Two examples demonstrates the computational effectiveness of our approach.

1 Introduction

Supervisory control theory (SCT), established by Ramadge and Wonham [1, 2], is a formal framework for the modeling and control of *discrete-event systems* (DES). Application domains include manufacturing systems, vehicular traffic, robotics, computer, and communication networks. Problems that SCT can address include dynamic allocation of resources, the prevention of system blocking, etc. and, within these constraints, maximally permissive system behavior. Although SCT can systematically synthesize supervisory controllers that are able to prevent a DES from executing undesirable behavior, industrial acceptance is scarce. A number of issues that hinder industrial use have been identified by various researchers [3, 4, 5]. Two main issues are the lack of a compact representation of large models and computational complexity.

In the former case, Sköldstam et al. [6] introduce a modeling formalism, called Extended Finite Automata (EFA), which are the ordinary automaton extended with discrete variables, guard expressions and action functions. The guards and action functions are attached to the transitions, which admits local design techniques of systems consisting of different parts. EFAs have been used in several research works and successfully applied to a range of examples such as [7, 8, 9, 10, 11]. Beside a number of methods for synthesizing EFAs [12, 13, 14], the EFA framework has been implemented in Supremica [15, 16], a verification and supervisory control tool, where powerful algorithms exist for analysis of DES [12, 17, 13, 18].

Even though EFAs eased the modeling experience by providing a compact modeling, SCT analysis is performed on their underlying automta models and therefore, the fundamental obstruction to the development of SCT, i.e. the computational complexity of synthesizing maximally permissive and nonblocking supervisors, still remains. Indeed, the nonblocking supervisory control problem for DES is NP-hard [19, 20]. It is well known that the exponential complexity of supervisor design arises from synchronizing subsystems into a global system model. Researchers are, therefore, seeking effective control methods for various subclasses of DES that enjoy special structure. Such structure will admit modularity [21, 22, 23, 24, 25] and model abstraction [26, 27, 28, 29] to circumvent computing global dynamic models.

The most effective model abstraction operator in SCT is the causal reporter map having the observer property [30, 27, 31, 32]. While [30] treated hierarchical control using general causal reporter maps, Feng and Wonham [33, 34, 35, 36], construct model abstractions only with natural observers, i.e., natural projections [37, 38] with the observer property. In this method, if two components share only a small number of common events, their abstractions tend to be small, and either verifying the nonconflicting property (if it holds) or designing a coordinator to achieve it may require only modest effort. Natural projection is a language-theoretic operation which needs the language of a system to be known or can be obtained by its generators, for instance, automaton. But, this is not the case for DES modeled by EFAs since the transitions are conditional, i.e., augmented with guards and actions and therefore, the language of the components can both be larger than or smaller than the language of the synchronized system. Hence, one cannot enjoy the compositional computation of natural projections.

In this paper, we tackle this by applying the projection on the transitions of a system modeled by EFAs rather than their underlying languages. To this end, we substitute the natural projection with transition projection to be able to abstract the system without knowing its language. We presents a sufficient condition for optimal nonblocking controller with partial observation in EFA by preserving the information needed for reliable representation of the nonblocking and controllability property.

This paper is organized as follows: Section 2 briefly describes Extended Finite Automata that is the modeling formalism used to model our problems. In Section 3, we introduce a model abstraction using Transition Projection, that is the projection on transition system, followed by Sections 4 and 5 in which its properties are explained. Two practical examples has been modeled and abstracted in Section 6. Finally, in Section 7 we conclude our work.

2 Preliminaries

2.1 Languages and Automata

The behavior of DES [38, 37] are described in term of event sequences and regular languages [2]. A regular language is a subset of strings that can be recognized by a *finite automaton* (FA) $G = (Q, \Sigma, \mapsto, Q^0, Q^m)$. Q is the finite *state* set. Σ is a non-empty finite event set called *alphabet*. $\mapsto \subseteq Q \times \Sigma \times Q$ is the state *transition function* mapping elements of $Q \times \Sigma$ into singletons of Q. The element $Q^0 \subseteq Q$ is the set of *initial states* and $Q^m \subseteq Q$ is the set of *marker states*.

The transition relation in G is written in infix notation $p \overset{\sigma}{\mapsto} q$. Let Σ^* be the set of all finite strings over Σ, including the empty string ε. Then, these notations can be extended to strings in Σ^* in the natural way by letting $p \overset{\varepsilon}{\mapsto} p$ for all $p \in Q$ and $p \overset{s\sigma}{\mapsto} q$ if $p \overset{s}{\mapsto} r$ and $r \overset{\sigma}{\mapsto} q$ for some $r \in Q$. Let $p \overset{\sigma}{\mapsto}$ denotes that there exists a state q such that $p \overset{\sigma}{\mapsto} q$. and $p \mapsto q$ denotes there exists a string $s \in \Sigma^*$ such that $p \overset{s}{\mapsto} q$ Automaton G is deterministic if Q^0 is a singleton q_0 and $p \overset{\sigma}{\mapsto} q$ and $p \overset{\sigma}{\mapsto} \acute{q}$ always implies $q = \acute{q}$.

In practice, some *uncontrollable* events in alphabet can never, or need not be disabled while some *controllable* can be inhibited by the supervisor. Hence, the event set Σ is partitioned into two disjoint subsets, controllable events Σ_c and uncontrollable events Σ_u, such that $\Sigma = \Sigma_c \dot{\cup} \Sigma_u$. Note that, by definition, the symbol ε does not belong to either of Σ, Σ_c, or Σ_u. If it is to be included, the event sets $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}, \Sigma_{\varepsilon,c} = \Sigma_c \cup \{\varepsilon\}$, and $\Sigma_{\varepsilon,u} = \Sigma_u \cup \{\varepsilon\}$ are used instead. An important property of an automaton is *nonblocking.* The automaton G is nonblocking if any state reachable from the initial state $q_0 \in Q^0$ can also reach a marker state via some string, i.e., $(\forall q \in Q) q_0 \mapsto q \Rightarrow q \mapsto p$ for some $p \in Q^m$.

Given two event sets Σ and $\Sigma_\ell \subseteq \Sigma$, the *natural projection* is the function $P : \Sigma^* \to (\Sigma - \Sigma_\ell)^*$ such that $P(\varepsilon) = \varepsilon$ and

$$P(\sigma) = \begin{cases} \varepsilon, & \sigma \in \Sigma_\ell \\ \sigma, & \sigma \in \Sigma - \Sigma_\ell \end{cases}$$
$$P(s\sigma) = P(s)P(\sigma), s \in \Sigma^*, \sigma \in \Sigma$$

The effect of P on a string $s \in \Sigma^*$ is just to erase the events in s that belongs to Σ_ℓ, but keep the events in $\Sigma - \Sigma_\ell$ unchanged. The *inverse image* of the natural projection P is a function $P^{-1} : Pwr(\Sigma^*) \to Pwr((\Sigma - \Sigma_\ell)^*)$ where Pwr is the power set.

2.2 Extended Finite Automata

Finite automaton can be extended with a set of variables to an *Extended Finite Automata* (EFA) whose transitions are augmented with Boolean conditions and actions on these variables to enjoy a compact and symbolic description of DES.

Let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be the set of n typed variables and D_i be the domain (type) of v_i. Let η denotes a tuple of *variable evaluations* $\eta : (\eta_1, \ldots, \eta_n) \to D$ that assigning to each variable $v_i \in \mathcal{V}$ its current value D_i. $\mathcal{G}$ is the set of Boolean conditions over $\mathcal{V}$ in which each condition g is a propositional logic formulæ whose propositional symbols are of the form $\bar{v} \in \bar{D}$ where $\bar{v} = (v_1, \cdots, v_n)$ is a n-tuple of pairwise distinct variables in $\mathcal{V}$ and $\bar{D}$ is a subset of the domains $D = D_1 \times \cdots \times D_n$. Let an arithmetic expression φ be formed according to the grammar

$$\varphi \ ::= \ \omega \mid w \mid (\varphi) \mid \varphi + \varphi \mid \varphi - \varphi \mid \varphi * \varphi \mid \varphi / \varphi \mid \varphi \% \varphi,$$

where $w \in \mathcal{V}$, and $\omega \in \bigcup_{i=1}^{n} D_i$. Then g is formed according to the grammar

$$\begin{aligned} g \ ::= \ & \varphi < \varphi \mid \varphi \leq \varphi \mid \varphi > \varphi \mid \varphi \geq \varphi \mid \varphi = \varphi \mid \\ & (g) \mid g \wedge g \mid g \vee g \mid \mathbf{T} \mid \mathbf{F}, \end{aligned}$$

where $\mathbf{T}$ and $\mathbf{F}$ represent boolean logic `true` and `false`, respectively, and all nonzero values are considered as $\mathbf{T}$. Given two guards g and h, we say that g is a subguard of h, denoted $g \preceq h$, if $g \wedge h = g$, and we say both g and h have the same evaluation for η, denoted $g = h$, if $\eta \models g \Leftrightarrow \eta \models h$ where $\models$ is the satisfaction relation [39]. Let $\mathcal{A}$ be the set of actions where each action

$a \in \mathcal{A}$ is an n-tuple of functions $(a_1, \ldots, a_n)$, updating the current variables evaluation η to the new evaluation $a(\eta)$. Every action function $a_i : D_i \to D_i$ is formed as $v_i := \varphi$. The symbol ξ is used to indicating that no variable is updated; and in vector form $\Xi = \{\xi, \ldots, \xi\}$. If $a_i = \xi$, we say that a_i is a don't care updating of the variable v_i, namely, $a(\eta(v_i)) = \eta(v_i)$.

Definition 1 (Extended Finite Automaton)**.**
An extended finite automaton over a set of variables $\mathcal{V}$ is a 8-tuple

$$E = (L, D, \Sigma, T, L^0, D^0, L^m, D^m),$$

where

- L is a finite set of discrete locations,
- $D = D_1 \times \cdots \times D_n$ is the domain of variables,
- Σ is a nonempty finite set of events (alphabets),
- $T \subseteq L \times \Sigma \times \mathcal{G} \times \mathcal{A} \times L$ is the conditional transition relation,
- $L^0 \subseteq L$ is the set of initial locations,
- $D^0 = D_1^0 \times \cdots \times D_n^0$ is the set of variables initial value,
- $L^m \subseteq L$ is the set of marked (desire) locations,
- $D^m \subseteq D$ is the set of marked value of the variables.

The initial variable evaluation is denoted by a tuple $\eta^0 = (\eta_1^0, \ldots, \eta_n^0)$ assigning each variable to its initial value $\eta_i^0 : v_i \to D_i^0$. The notation $\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}$ is used as shorthand for $(\ell, \sigma, g, a, \acute{\ell}) \in T$. If the condition, also called guard, of the conditional transition $\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}$, is a tautology, e.g. $g = \mathbf{T}$ or $g = (v < 1) \vee (v \geqslant 1)$, then we simply write $\ell \xrightarrow{\sigma}_{a} \acute{\ell}$.

It is assumed that all actions are written as constant functions $v_i := a(v_i)$ where the new value of v_i only depends on its previous value. Any transition can be decomposed into multiple transitions of this form. For instance, the transition $\ell \xrightarrow{\sigma}_{x:=y+1} \acute{\ell}$ where $D(y) = \{0, 1\}$ can be decomposed into multiple transitions $\ell \xrightarrow{\sigma}_{y=0/x:=1} \acute{\ell}$ and $\ell \xrightarrow{\sigma}_{y=1/x:=2} \acute{\ell}$.

Each EFA can be unfolded to its underlying FA whose states and transitions are defined as follows:

Definition 2 (FA Semantics of an EFA)**.**
Let $E = (L, D, \Sigma, T, L^0, D^0, L^m, D^m)$ be an EFA over the set of variables $\mathcal{V}$. The FA $G(E)$ is the tuple $(Q_E, \Sigma_E, \mapsto_E, Q_E^0, Q_E^m)$ where

- $Q_E = L \times D$,

- $\mapsto_E \subseteq Q \times \Sigma \times Q$ is defined by the following rule:

$$\frac{\ell \xrightarrow{\sigma}_{g/a} \acute{\ell} \wedge \eta \models g}{\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \acute{\ell}, a(\eta) \rangle},$$

- $Q_E^0 = L^0 \times D^0$,
- $Q_E^m = L^m \times D^m$.

States of $G(E)$ are the set of reachable states of E and each state consists of a location ℓ together with a tuple η of variable evaluations. Note that in the definition of transition relation $\mapsto$, if the proposition above the horizontal line holds, then the proposition under the line holds as well (also known as Structured Operational Semantics), namely, whenever the guard g of the conditional transition $\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}$ holds for the variables evaluation η, i.e. $\eta \models g$, then there is a transition in $G(E)$ from state $\langle \ell, \eta \rangle$ to state $\langle \acute{\ell}, a(\eta) \rangle$. Note that, the DFA generated directly from EFA by constructing the state set as $L \times D$ is not guaranteed to be the canonical recognizer and therefore further reduction needs to be done by using the standard algorithm of minimization [40]. In the sequel, we assume that the DFA obtained by the above transformation is a canonical recognizer of the language represented by the input EFA model.

Since we are interested in the deterministic systems, we only focus on deterministic EFAs and, for the sake of brevity, we simply write EFAs for deterministic EFAs.

Definition 3 (Deterministic EFA)**.**
An EFA E is deterministic if $G(E)$ is deterministic, namely, the set of initial states of $G(E)$ is a singleton $\langle \ell^0, \eta^0 \rangle$, where $\ell^0 \in L^0$ and η^0 is initial variable evaluation, and for all transitions $\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \acute{\ell}, \acute{\eta} \rangle$ and $\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \grave{\ell}, \grave{\eta} \rangle$ always implies $\langle \acute{\ell}, \acute{\eta} \rangle = \langle \grave{\ell}, \grave{\eta} \rangle$.

The isomorphism for two deterministic EFAs is defined as follows.

Definition 4 (Isomorphic EFAs)**.**
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, \eta^0, L_k^m, D_k^m)$ be two EFAs over the set of variables $\mathcal{V}_k$ $(k = 1, 2)$. E_1 and E_2 are isomorphic up to renaming of the locations and variables if the following holds:

(i) $\Sigma_1 = \Sigma_2$,

(ii) There exists two bijective functions
$F : L_1 \to L_2$ and $V : \mathcal{V}_1 \to \mathcal{V}_2$ such that
$(\forall v_i \in V_1) \eta_2^0(V(v_i)) = \eta_1^0(v_i), F(\ell_1^0) = \ell_2^0$, and
$\ell_1 \xrightarrow{s}_{1,g/a} \acute{\ell}_1 \Leftrightarrow F(\ell_1) \xrightarrow{s}_{2,V(g)/V(a)} F(\acute{\ell}_1)$.

Proposition 1. *Let E_k be two EFAs over the set of variables $\mathcal{V}_k (k = 1, 2)$. If E_1 and E_2 are isomorphic then $G(E_1)$ and $G(E_2)$ are isomorphic.*

Proof. Let $G(E_k)$ be the tuples $(Q_k, \Sigma_k, \mapsto_k, q_k^0, Q_k^m)$ for $k = 1, 2$. Assume a function $U : L_1 \times D \to L_2 \times D$ such that $U(\langle \ell_1^0, \eta_1^0 \rangle) = \langle \ell_2^0, \eta_2^0 \rangle$ and $\langle \ell_1, \eta_1 \rangle \overset{s}{\mapsto}_1 \langle \acute{\ell}_1, \acute{\eta}_1 \rangle \Leftrightarrow U(\langle \ell_1, \eta_1 \rangle) \overset{s}{\mapsto}_2 U(\langle \acute{\ell}_1, \acute{\eta}_1 \rangle)$. To justify the statement, we must show that $\Sigma_1 = \Sigma_2$ and U is a bijective function that is every state in $G(E_1)$ consists of a location in L_1 together with a variables evaluation in D are mapped to exactly one state in $G(E_2)$ consists of a location in L_2 and evaluation in D. By the hypothesis assumption since E_1 and E_2 are isomorphic, we can immediately see that $\Sigma_1 = \Sigma_2$. Also, by the assumption there exists two bijective functions $F : L_1 \to L_2$ and $V : \mathcal{V}_1 \to \mathcal{V}_2$ mapping the locations and variables. The function V is by assumption bijective which implies the evaluation of variables on both sides are the same up to renaming the variables, that is $\eta_1^0 = \eta_2^0, \eta_1 = \eta_2$ and $\acute{\eta}_1 = \acute{\eta}_2$. F is also bijective and therefore, every location in L_1 is paired with exactly one location in L_2. We can conclude that every state in $G(E_1)$ which consists of a location in L_1 and a tuple of variable valuation η in D are mapped by U to exactly one state in $G(E_2)$. Hence, U is a bijective function and $G(E_1)$ and $G(E_2)$ are isomorphic up to renaming the locations and variables. □

Definition 5 (Sub-EFA)**.**
Let $E_k = (L_k, D, \Sigma, T_k, L^0, D^0, D_k^m, L_k^m),\ k = 1, 2$, be two EFAs over the set of shared variables $\mathcal{V}$ with the same set of events,variables domain, initial locations, and initial variables value. Then E_1 is a *sub-EFA* of E_2 written $E_1 \subseteq E_2$, if $L_1 \subseteq L_2, T_1 \subseteq T_2, D_1^m \subseteq D_2^m$, and $L_1^m \subseteq L_2^m$.

Since the transitions in EFAs are conditional, it is not possible to describe the static behavior of the system by following its transitions before evaluating its guards and actions. Therefore, we introduce a notion of dynamic execution fragment that is a series of conditional transitions ending with a location.

Definition 6 (Finite Dynamic Execution Fragment)**.**
Let $E = (L, D, \Sigma, T, L^0, D^0, L^m, D^m)$ be an EFA over set of variables $\mathcal{V}$. A *finite dynamic execution fragment* ϱ in E is a series of transitions

$$\varrho = \ell_0 \overset{\sigma_1}{\rightarrow}_{g_1/a_1} \ell_1 \overset{\sigma_2}{\rightarrow}_{g_2/a_2} \cdots \overset{\sigma_{i+1}}{\rightarrow}_{g_{i+1}/a_{i+1}} \ell_{i+1},\ (0 \leqslant i < n),$$

in T where $n \geqslant 0$ and the variables evaluation $\eta_{i+1} = a(\eta_i)$.

The integer n is the length of the ϱ and $\varrho = \ell_0$ for some $\ell_0 \in L$ is a legal finite dynamic execution fragment of length $n = 0$. Note that, in finite dynamic execution fragments, we do not explicitly list the selfloops of the empty string ε as they are trivially contained in any EFA. From now on, for the sake of simplicity, the term *dynamic execution fragment* will be used to denote a finite dynamic execution fragment. The first and last location of ϱ is denoted by *first*(ϱ) and *last*(ϱ), respectively, *str*(ϱ) denotes $\sigma_1\sigma_2\ldots\sigma_{i+1}$, and *Loc*$(\varrho)$ denotes the set of locations ℓ_j $(\forall j \in \mathbf{n})$, that can be reached by following the transitions in ϱ. We call ϱ an initial dynamic execution fragment if *first*$(\varrho) \in L^0$ and $\eta_0 = \eta^0$, and a marked dynamic execution fragment if *last*$(\varrho) \in L^m$ and $\eta_n \in D^m$. Finally, ϱ is accepted by E if for every transition $\ell_i \overset{\sigma_{i+1}}{\rightarrow}_{g_{i+1}/a_{i+1}} \ell_{i+1} \in \varrho$ we have $\eta_i \models g_{i+1}$.

Additionally, for two dynamic execution fragments $\varrho, \acute{\varrho}$ in E, ϱ is a *precedence* of $\acute{\varrho}$, written $\varrho \sqsubseteq \acute{\varrho}$, if *last*$(\varrho) =$ *first*$(\acute{\varrho})$, and $\varrho = \acute{\varrho}$ if *str*$(\varrho) =$ *str*$(\acute{\varrho})$ and for all $\ell_i \overset{\sigma_{i+1}}{\rightarrow}_{g_{i+1}/a_{i+1}} \ell_{i+1} \in \varrho$ there exist $\acute{\ell}_i \overset{\sigma_{i+1}}{\rightarrow}_{\acute{g}_{i+1}/\acute{a}_{i+1}} \acute{\ell}_{i+1} \in \acute{\varrho}$ such that $\eta_i \models g_{i+1} \Leftrightarrow \acute{\eta}_i \models \acute{g}_{i+1}$, and $a_{i+1}(\eta_i) = \acute{a}_{i+1}(\acute{\eta}_i)$.

EFAs similar to ordinary automata are composed by extended full synchronous composition (EFSC). By the definition of EFSC, it is assumed that the variables are shared by all EFAs with the same initial values. In the composition of two EFAs, a shared event is enabled if and only if it is enabled by each of the composed EFAs.

Definition 7 (EFSC)**.**
Let $E = (L_k, D, \Sigma_k, T_k, \ell_k^0, \eta^0, L_k^m, D_k^m),\ k = 1, 2$, be two EFAs over the set of shared variables $\mathcal{V}$. The *Full Synchronous Composition* of E_1 and E_2 is

$$E_1 \| E_2 = (L, D, \Sigma, T, \ell^0, \eta^0, L^m, D^m)$$

where

- $L = L_1 \times L_2$,
- $\Sigma = \Sigma_1 \cup \Sigma_2$,
- T is defined by the rules:
  - $\dfrac{\ell_1 \overset{\sigma}{\rightarrow}_{1,g_1/a_1} \acute{\ell}_1 \wedge \ell_2 \overset{\sigma}{\rightarrow}_{2,g_2/a_2} \acute{\ell}_2 \wedge \sigma \in (\Sigma_1 \cap \Sigma_2)}{\langle \ell_1, \ell_2 \rangle \overset{\sigma}{\rightarrow}_{g/a} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle}$ such that

    (i) $g = g_1 \wedge g_2$,

    (ii) For $i = 1, \ldots, n$:

$$a_i = \begin{cases} a_{1i} & \text{if } a_{1i} = a_{2i} \\ a_{1i} & \text{if } a_{2i} = \xi \\ a_{2i} & \text{if } a_{1i} = \xi \\ \eta_i & \text{otherwise;} \end{cases}$$

  - $\dfrac{\ell_1 \overset{\sigma}{\rightarrow}_{1,g_1/a_1} \acute{\ell}_1 \wedge \ell_2 = \acute{\ell}_2 \wedge \sigma \in (\Sigma_1 - \Sigma_2)}{\langle \ell_1, \ell_2 \rangle \overset{\sigma}{\rightarrow}_{g_1/a_1} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle}$;
  - $\dfrac{\ell_2 \overset{\sigma}{\rightarrow}_{2,g_2/a_2} \acute{\ell}_2 \wedge \ell_1 = \acute{\ell}_1 \wedge \sigma \in (\Sigma_2 - \Sigma_1)}{\langle \ell_1, \ell_2 \rangle \overset{\sigma}{\rightarrow}_{g_2/a_2} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle}$.

- $L^0 = L_1^0 \times L_2^0$,
- $L^m = L_1^m \times L_2^m$.

Note that if the action functions of E_1 and E_2 tries to update a shared variable to different values, the variable is, by default, not updated. In situation where two values are conflicting, is usually a consequence of bad modeling. In this work, in order to avoid conflicting variables, we assume that for any two conditional transitions in the system with the same label, say $\ell_1 \overset{\sigma}{\rightarrow}_{g/a} \ell_2$ and $\acute{\ell}_1 \overset{\sigma}{\rightarrow}_{\acute{g}/\acute{a}} \acute{\ell}_2$, if $a, \acute{a} \neq \Xi$ always implies $a(\eta) = \acute{a}(\acute{\eta})$. In general, this assumption may restrict the modeling using EFA. But, in practice the common events in ordinary finite automata are used for communication (synchronous composition) in contrast to EFAs were communication is normally performed by variables. Therefore, without loss of generality, a DES modeled by finite automata, in which shared events are used for instance to specify mutual exclusion, can now be modeled by EFAs using guards and actions on transitions labeled by distinct events.

2.3 Supervisory Control of EFA

SCT is a formal framework for the modeling and control of DES consist of plant and specification. In the context of SCT, the behavior of a system is usually represented by its language, i.e., the sets of strings that the system may generate. Conventionally, automata has been used as the modeling formalism to generate the language. In this, the control problems are modeled by EFAs, while the SCT analysis is performed on their corresponding DFA models. There are different methods of computing a supervisor as mentioned earlier and in this work, we use the symbolic algorithm presented in [14] to efficiently synthesize a supervisor. The algorithm iteratively strengthens the guards on conditional transitions to avoid forbidden or blocking states.

Given a DES control problem, we consider that plant is modeled by an EFA P and specification by EFA Sp. The specification can be represented, without loss of generality, with a set of forbidden locations which can be obtained by a refined plant R model with the same behaviors as P such that the executions not allowed in Sp end up in certain forbidden locations in R. See [14] for more elaboration on refinement.

From now on, we assume that the plant model is given as the refined EFA R and the specification is given as the set of forbidden locations $L_f \subset L_R$. Let denote the set of safe locations by $L_s = L - L_f$ and recall the set of reachable states Q_R in $G(R)$. A state $q = \langle \ell, \eta \rangle \in Q_R$ is a forbidden state iff $\ell \in L_f$, otherwise, q is a safe state. In the sequel, R_s denotes the EFA obtained from R by assigning $\mathbf{F}$ to the guard g of every transition $\ell \overset{\sigma}{\rightarrow}_{g/a} \acute{\ell}$ for which $\acute{\ell} \in L_f$, i.e., $\acute{\ell}$ is a forbidden location. R_s is constructed such that $R_s \subseteq R$ and is called the safe sub-EFA of R.

Definition 8 (Nonblocking, Safety, Controllability)**.**
[14] Let R be an EFA, L_f its set of forbidden locations, and R_s its safe subautomaton. A reachable state $q \in Q_R$ is: (a) *nonblocking* if there exists a state $p \in Q_R^m$ such that $q \overset{s}{\mapsto} p$ for some string $s \in \Sigma^*$; (b) safe if $q \in Q_{R_s}$ and (c) (R, L_f, Σ_u)-controllable (or simply *controllable* when clear from context) if q is safe and $\forall \sigma \in \Sigma_R(q) \cap \Sigma_u$ where $Sg_R(q)$ denote the set of active events, we have $Q_R(q, \sigma) \subseteq Q_{R_s}$. The EFA R is, respectively, nonblocking, safe, and controllable if every reachable state of is, respectively, nonblocking, safe, and controllable.

A supervisor $\mathcal{S}$ for R can be seen as a function $\mathcal{S} : T \to \mathcal{G}$ which maps each transition to a supervision guard such that $\mathcal{S}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}) \preceq g$ if $\sigma \in \Sigma_c$, and $\mathcal{S}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}) = g$ if $\sigma \in \Sigma_u$. Let $R^{\mathcal{S}}$ denote the sub-EFA obtained from R by replacing its guards by those provided by $\mathcal{S}$. Then, $\mathcal{S}$ is said to be nonblocking if $R^{\mathcal{S}}$ is nonblocking and safe if $R^{\mathcal{S}}$ is safe. In case $R^{\mathcal{S}}$ is blocking or uncontrollable, a search will be performed to find a safe and nonblocking supervisor $\mathcal{S}$ such that $R^{\mathcal{S}} \subseteq R_s$. Let $\mathcal{S}(R, L_f)$ denotes the set of nonblocking and safe supervisor candidates of R, then $\mathcal{S}^{\uparrow} := sup\mathcal{S}(R, L_f)$, is the *most permissive nonblocking and safe supervisor* than any other supervisor in $\mathcal{S}(R, L_f)$ when the latter is nonempty. The $R^{\mathcal{S}^{\uparrow}}$ is called the supremal controllable and nonblocking sub-EFA of R_s.

$R^{\mathcal{S}^{\uparrow}}$ is calculated by the Supervisory Synthesis for EFA (SSEFA)[14] using fixed-point iteration method. Given a refined EFA R and a set $L_f \subset L$ of forbidden location, SSEFA(R, L_f) computes stronger, maximally permissive, guards for the transitions of R in N steps such that the obtained EFA is nonblocking, safe and controllable. The iteration is terminated in N steps when no guards is modified, namely, for all locations $\ell \in L$ and current variable evaluation η we have $g^{N+1} = g^N$. To compute the stronger guards for the controllable transitions, the algorithm uses two guards associated to every location ℓ: a *nonblocking guard*, denoted N_ℓ, and a *bad location guard*, denoted B_ℓ. In the jth iteration, a state $\langle \ell, \eta \rangle$ is flagged nonblocking if $\eta \models N_\ell^j$ and undesirable (blocking, forbidden or uncontrollable) if $\eta \models B_\ell^j$.

Theorem 1 (Supremal Controllable and Nonblocking EFA)**.**
Given an EFA $R = (L, D, \Sigma, T, L^0, D^0, L^m, D^m)$ and a set $L_f \subset L$ of forbidden location, if SSEFA(R) is nonblocking and controllable, then it is the supremal controllable and nonblocking sub-EFA of R.

Proof. See [14]. □

If R is deterministic, then SSEFA(R, L_f) is also deterministic. In this paper, any nondeterministic EFA is the result of an abstraction of an deterministic model and we will use transformations ensuring that a meaningful supervisor can also be constructed.

3 EFA Projection

Traditionally, brute-force computation is used for the verification and coordination [37, 27]. This we wish to avoid since the nonblocking supervisory control problem in the SCT [27] is NP-hard [19, 20]. We may find efficient solutions only for various subclasses of discrete-event systems (DES) that enjoy special structures. Such structures will admit modularity [21, 41, 42, 43] and model abstraction [26, 28, 29, 27] to obviate computing global dynamic models. Abstraction introduces hierarchy into system structure, as it reports only the events shared with other subsystems and conceals the rest. The fewer the reported events, the greater state reduction will be achieved. Natural projection [44] with observer property is a language-theoretic operation which cannot be used for EFAs where the language of the components cannot be used before evaluating the guards and the actions. In order to use the model abstraction using projection, we substitute the natural projection with transition projection to be able to abstract the system without knowing its language. In this section a DES is assumed to consist of a group of simple plant EFA components subject to a conjunction of modular control specifications. Before introducing the transition projection we need the following notions.

For an event σ, let $\mathit{Act}(\sigma) \subseteq \mathcal{A}$ and $\mathit{Con}(\sigma) \subseteq \mathcal{G}$ be the sets of actions and conditions, respectively, retrieved from all transitions labeled with σ. Note that, by the assumption, the set $\mathit{Act}(\sigma)$ is a singleton a_σ.

Definition 9 (Local Event)**.**
For an EFA $E_i, i \in \mathbf{n}$, over the set of shared variables $\mathcal{V}$, an event $\sigma \in \Sigma_i$ is *local* to E_i if for all $j \in \mathbf{n}$ we have

(i) $\sigma \in \Sigma_i - \bigcup \Sigma_j (j \neq i)$,

(ii) $(\forall g \in \mathit{Con}(\sigma))\ g = \mathbf{T}$,

(iii) $(\forall g \in \mathcal{G}_j)\eta \models g \Leftrightarrow a_\sigma(\eta) \models g$.

The set of local events is denoted Σ_ℓ.

Here, condition (i) guarantees that the event σ only appears in E_i and not in any other EFAs $E_j (j \neq i)$, (ii) ensures that guards on any transition labeled by σ is always evaluates to true; hence σ can cause the transition to occur at any time, and (iii) guarantees that the execution of action a_σ has no effect on any guards evaluation. Any transition labeled with a local events is called a *local transition* and similarly any dynamic execution fragment is local if its transitions are all local.

For a system described by a language $L \subseteq \Sigma^*$ and a natural projection P with the observer property, the model abstraction of the system is the induced system representing language $P(L)$. The observer concept is equivalent [27] to observation equivalence introduced by Milner [45]. The equivalence kernel of a natural observer is a bisimulation relation and the abstraction is the bisimulation quotient of the original system modeled by automata [27, 46, 39]. The natural projection, as a special form of reporter map, allows compositional definition and computation.

For the same system modeled by a group of EFA components, the language of individual component is not defined without considering the global behavior of the system, i.e. the synchronous composition of all components. Therefore, we substitute the natural projection with a function, called transition projection, that directly projects the transitions in EFAs.

For an EFA E over the set of variables $\mathcal{V}$ and the set of events Σ, the transition projection $\bar{P}$ for the conditional transition relation T and the set $\Sigma_\ell \subseteq \Sigma$ is defined as follows:

$$\bar{P} : T \times \Sigma_\ell \to T$$

where for every transition

$$\bar{P}[\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \varepsilon] = \ell \xrightarrow{\sigma}_{g/a} \acute{\ell}$$

$$\bar{P}[\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \gamma] = \begin{cases} \ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, & \sigma \neq \gamma \\ \ell \xrightarrow{\varepsilon}_{g/a} \acute{\ell}, & \sigma = \gamma \end{cases}$$

The transition projection $\bar{P}$ replace the label of transitions labeled by events in Σ_ℓ with ε symbol. In effect, an EFA is allowed to make a transition spontaneously, without receiving an input event and the actions of such transitions has no effect on global behavior. Extending T to its power set $Pwr(T)$, we get $\bar{P} : Pwr(T) \times \Sigma_\ell \to Pwr(T)$ such that for any $\tau \in \Sigma_\ell$, $N \subseteq T$: $\bar{P}(N, \tau) = \{\bar{P}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \tau) | \ell \xrightarrow{\sigma}_{g/a} \acute{\ell} \in N\}$. If we further extend Σ_ℓ to its power set $Pwr(\Sigma_\ell)$, $\bar{P}$ becomes $\bar{P} : Pwr(T) \times Pwr(\Sigma_\ell) \to Pwr(T)$ such that for $A \in \Sigma_\ell$, $N \subseteq T$: $\bar{P}(N, A) = \bigcup\{\bar{P}(N, \tau) | \tau \in A\}$. If the action of $\bar{P}$ on T is understood then $\bar{P}[T, \Sigma_\ell]$ may be written $\bar{P}_{\Sigma_\ell} T$ and if $\bar{P}$ is defined $\bar{P}T$.

Given any EFA, Algorithm 1, denoted by $\hat{P}$, computes the projection for the conditional transitions of E, eliminates the resulting ε-transitions, and returns the Projected EFA $\tilde{E}$. The elimination of ε-transitions in Algorithm 1 is very close to the algorithms of eliminating ε-transitions and subset construction for DFA in [40]. The algorithm uses the notation $\mathcal{S}_\varepsilon(\ell)$ to find the ε-closure of ℓ, i.e., the set of ε-locations by finding every location that can be reached from ℓ along any path whose transitions are all labeled ε. Formally, $\mathcal{S}_\varepsilon(\ell)$ is defined recursively as follows:

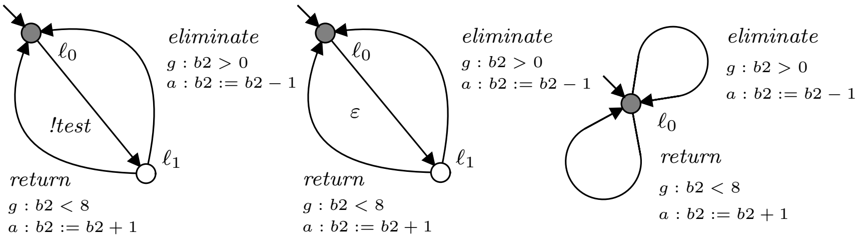
1. $\ell \in \mathcal{S}_\varepsilon(\ell)$
2. $(\forall \acute{\ell} \in \mathcal{S}_\varepsilon(\ell))\ \acute{\ell} \xrightarrow{\varepsilon}_{g/a} \grave{\ell} \Rightarrow \grave{\ell} \in \mathcal{S}_\varepsilon(\ell)$.

```
**Algorithm 1** EFA Projection ($\hat{P}$)
**Input:** An EFA $E = (L, D, \Sigma, T, \ell^0, \eta^0, L^m, D^m)$ and subset of events $\Sigma_\ell \subseteq \Sigma$
$\tilde{T} := \emptyset$
$\tilde{L} := \emptyset$
$\tilde{\Sigma} := \Sigma - \{\varepsilon\}$
$\bar{P} : T \times \Sigma_\ell \to T$
$T = \bar{P}[T, \Sigma_\ell]$
$S := \mathcal{S}_\varepsilon(\ell^0)$
$\tilde{L} = \tilde{L} \cup \{S\}$
**do**
  $X = \emptyset$
  **foreach** $\acute{S} \in S$ **do**
    **foreach** $\sigma \in \tilde{\Sigma}$ **do**
      $\grave{S} := \{\mathcal{S}_\varepsilon(\acute{\ell}) | (\forall \ell \in \acute{S})(\ell, \sigma, g, a_\sigma, \acute{\ell}) \in T\}$
      **if** $\grave{S} \neq \emptyset$ **then**
        $\tilde{L} = \tilde{L} \cup \{\grave{S}\}$
        $X = X \cup \{\grave{S}\}$
        $g_\sigma :=$ **F**
        $(\forall \acute{\ell} \in \acute{S})(\forall \grave{\ell} \in \grave{S})(\acute{\ell}, \sigma, g, a_\sigma, \grave{\ell}) \in T \Rightarrow$
        $g_\sigma = g_\sigma \vee g$
        **if** $g_\sigma =$ **F** **then** $g_\sigma =$ **T** **end if**
        $\tilde{T} = \tilde{T} \cup \{(\acute{S}, \sigma, g_\sigma, a_\sigma, \grave{S})\}$
      **end if**
    **end for**
  **end for**
  $S = X$
**until** $S = \emptyset$
$\tilde{L}^m := \{S \in \tilde{L} | S \cap L^m \neq \emptyset\}$
$\tilde{L}^0 := \{S \in \tilde{L} | S \cap \ell^0 \neq \emptyset\}$
**Output:** An EFA $\tilde{E} = (\tilde{L}, D, \tilde{\Sigma}, \tilde{T}, \tilde{L}^0, D^0, \tilde{L}^m, D^m)$
```

$\mathcal{S}_\varepsilon$ can be extended to set of locations by letting $\mathcal{S}_\varepsilon(S) = \bigcup_{\ell \in S} \mathcal{S}_\varepsilon(\ell)$ for some set of locations S.

Example 1. *Consider EFA TU, TU_ε, and $\tilde{TU}$ in Fig. 1. Assume the set of local events Σ_ℓ with $\{!test\} \in \Sigma_\ell$. Then TU_ε is the result of transition projection $\bar{P} : T \times \Sigma_\ell \to T$ by projecting the local transition labeled with $\{!test\}$ and $\tilde{TU}$ is the projected EFA $\tilde{TU}$ by eliminating ε-transition in TU_ε.*

Turning to supervisory control problem, consider a system consisting of two EFA components, E_1 and E_2, over event sets $\Sigma_i (i = 1, 2)$. To obtain a reduction of the system, we could first compute the systems global behavior



(a) TU (b) TU_ε (c) $\tilde{TU}$

Figure 1: TU_ε is the result of projecting transition labeled *!test* and $\tilde{TU}$ is the projected EFA by eliminating ε-transition.

$E_1\|E_2$ and then its transition projection. When, however, the local events of the two components are all defined the result is obtained more economically from reductions of the components, according to the following proposition. This result is central to our method.

Proposition 2.
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, \eta^0, L_k^m, D_k^m), k = 1, 2,$ be two EFAs over the set of shared variables $\mathcal{V}$. Consider T as the set of transition relation for $E_1\|E_2$ and $\Sigma_\ell \subseteq \Sigma := \Sigma_1 \cup \Sigma_2$. Define $\bar{P} : T \times \Sigma_\ell \to T$ and $Q_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i (i = 1, 2)$. If Σ_ℓ is the set of local events then

$$\hat{P}[E_1\|E_2, \Sigma_\ell] = \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]\|\hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell].$$

The proof is lengthy but straightforward and needs the following lemmas.

Lemma 1. *Let $E = (L, D, \Sigma, T, L^0, D^0, L^m, D^m)$ be an EFA with the set of variables $\mathcal{V}$, $\Sigma_\ell \subseteq \Sigma$ be the subset of local events, and let $\Pi(\ell) := \{\varrho \in E \mid \varrho$ is local and first$(\varrho) = \ell\}$ be a set of local dynamic execution fragments starting from $\ell \in L$. Define the transition projection $\bar{P} : T \times \Sigma_\ell \to T$ and consider a set $\mathcal{S}_\varepsilon(\ell)$ of ε-closure that can be reached from ℓ. Then $\bigcup_{\varrho\in\Pi(\ell)} Loc(\bar{P}\varrho) = \mathcal{S}_\varepsilon(\ell)$.*

Proof. For any local transition $\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}$ where $\sigma \in \Sigma_\ell$ we have $\bar{P}[\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}] = \ell \xrightarrow{\varepsilon}_a \acute{\ell}$. Thus, for all $\varrho = \ell_0 \xrightarrow{\sigma_1}_{a_1} \cdots \xrightarrow{\sigma_{i+1}}_{a_{i+1}} \ell_{i+1} \in \Pi(\ell) (0 \leqslant i < n)$ we have $\bar{P}\varrho = \bar{P}[\ell_0 \xrightarrow{\sigma_1}_{a_1} \cdots \xrightarrow{\sigma_{i+1}}_{a_{i+1}} \ell_{i+1}, \Sigma_\ell] = \ell_0 \xrightarrow{\varepsilon}_{a_1} \cdots \xrightarrow{\varepsilon}_{a_{i+1}} \ell_{i+1}$, and $str(\bar{P}\varrho) = \varepsilon$. Therefore, $\mathcal{S}_\varepsilon(\ell) = \bigcup_{\varrho\in\Pi(\ell)} Loc(\bar{P}\varrho)$. □

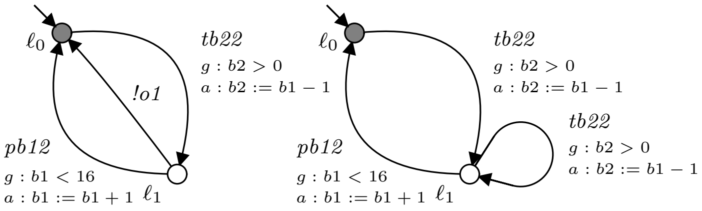
Returning to the proof of Proposition 2,

Proof of Proposition 2. It needs to be shown that both conditional transition relations of $\hat{P}[E_1\|E_2, \Sigma_\ell]$ and $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]\|\hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ are the same (up to isomorphism). It can be proved by induction on size of a dynamic execution fragment ρ. Let the intermediate ε-EFAs result of

$\hat{P}[E_1 \| E_2, \Sigma_\ell]$, $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]$, and $\hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ in Algorithm 1 be $\bar{E}, \bar{E}_1$, and $\bar{E}_2$, respectively, and let $\langle \ell_0^1, \ell_0^2 \rangle, \ell_0^1$, and ℓ_0^2 be the initial locations of $\hat{P}[E_1 \| E_2, \Sigma_\ell]$, $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]$, and $\hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$, respectively.

> **BASIS:** Let $|\rho| = 0$, i.e. ρ is the initial location. Based on EFSC, for the initial locations we have $\langle \ell_0^1, \ell_0^2 \rangle = \ell_0^1 \times \ell_0^2$. We need to show that both sets of ε-locations that can be reached from the initial location of $\bar{E}$ and the initial location of $\bar{E}_1$ and $\bar{E}_2$ are the same, namely $\mathcal{S}_\varepsilon(\langle \ell_0^1, \ell_0^2 \rangle) = \mathcal{S}_\varepsilon^1(\ell_0^1) \times \mathcal{S}_\varepsilon^2(\ell_0^2)$, so they construct the same set of initial locations after eliminating the ε-transitions by Algorithm 1 in $\hat{P}[E_1 \| E_2, \Sigma_\ell]$, $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]$, and $\hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$, respectively. Let $\Pi(\langle \ell_0^1, \ell_0^2 \rangle), \Pi^1(\ell_0^1)$, and $\Pi^2(\ell_0^2)$ be the sets of local dynamic execution fragments from the initial locations of $E_1 \| E_2, E_1$, and E_2, respectively. Then we have, $\Pi(\langle \ell_0^1, \ell_0^2 \rangle) = \Pi^1(\ell_0^1) \| \Pi^2(\ell_0^2)$ and by Lemma 1, we get $\bigcup_{\varrho \in \Pi(\langle \ell_0^1, \ell_0^2 \rangle)} Loc(\bar{P}\varrho) = \bigcup_{\varrho^1 \in \Pi^1(\ell_0^1)} Loc(\bar{P}\varrho^1) \times \bigcup_{\varrho^2 \in \Pi^2(\ell_0^2)} Loc(\bar{P}\varrho^2)$. Therefore, we can conclude that $\mathcal{S}_\varepsilon(\langle \ell_0^1, \ell_0^2 \rangle) = \mathcal{S}_\varepsilon^1(\ell_0^1) \times \mathcal{S}_\varepsilon^2(\ell_0^2)$.
>
> **INDUCTION:** Let ρ constructed by $\acute{\rho}$ and $\grave{\rho}$ such that $\acute{\rho} \sqsubseteq \grave{\rho}$, be of length $n+1$, and assume the statement for length n, i.e. $\acute{\rho}$. By inductive hypothesis, both sets of locations that can be reached by following $\acute{\rho}$ from the initial location of $\hat{P}[E_1 \| E_2, \Sigma_\ell]$ and the initial location of $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ are the same. Let this set be $\{\ell_1, \ldots, \ell_k\}$. Then, we compute all the locations that can be reached by $\grave{\rho}$ starting from any of ℓ_k. Let the set of these locations be $\mathcal{S} := \{\ell_1, \ldots, \ell_m\} \subseteq L_1 \times L_2$. We know that any location $\ell_m \in \mathcal{S}$ is constructed by the locations $\ell_m^1 \in L_1$ and $\ell_m^2 \in L_2$ such that $\ell_m = \ell_m^1 \times \ell_m^2$. Let these two sets be denoted $\mathcal{S}^1 := \{\ell_1^1, \ldots, \ell_m^1\} \subseteq L_1$ and $\mathcal{S}^2 := \{\ell_1^2, \ldots, \ell_m^2\} \subseteq L_2$, respectively. Now, it is enough to show that the set of ε-locations for all ℓ_m in $\bar{E}$ is the same as the set of ε-locations for $\ell_m^1 \times \ell_m^2$ in $\bar{E}_1$ and $\bar{E}$, respectively, namely $\bigcup_{\ell_i \in \mathcal{S}} \mathcal{S}_\varepsilon(\ell_i) = \bigcup_{\ell_i^1 \in \mathcal{S}^2} \mathcal{S}_\varepsilon^1(\ell_i^1) \times \bigcup_{\ell_i^2 \in \mathcal{S}^2} \mathcal{S}_\varepsilon^2(\ell_i^2)$. Let $\bigcup_{\ell_i \in \mathcal{S}} \Pi(\ell_i), \bigcup_{\ell_i^1 \in \mathcal{S}^1} \Pi^1(\ell_i^1)$, and $\bigcup_{\ell_i^2 \in \mathcal{S}^2} \Pi^2(\ell_i^2)$ be the sets of local dynamic execution fragments in $E_1 \| E_2, E_1$, and E_2 for all $\ell_i \in \mathcal{S}, \ell_i^1 \in \mathcal{S}^1$, and $\ell_i^2 \in \mathcal{S}^2$, respectively. For every ℓ_i, ℓ_i^1, and ℓ_i^2 we have $\Pi(\ell_i) = \Pi^1(\ell_i^1) \parallel \Pi^2(\ell_i^2)$ and by Lemma 1, $\bigcup_{\varrho \in \Pi(\ell_i)} Loc(\bar{P}\varrho) = \bigcup_{\varrho^1 \in \Pi^1(\ell_i^1)} Loc(\bar{P}\varrho^1) \times \bigcup_{\varrho^2 \in \Pi^2(\ell_i^2)} Loc(\bar{P}\varrho^2)$. Therefore, we can conclude that $\mathcal{S}_\varepsilon(\ell_i) = \mathcal{S}_\varepsilon^1(\ell_i^1) \times \mathcal{S}_\varepsilon^2(\ell_i^2)$.

We have now proved that both $\hat{P}[E_1 \| E_2, \Sigma_\ell]$ and $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ transition relations are the same up to renaming the locations and variables. □



(a) $M2$ (b) $\tilde{M}2$

Figure 2: $\tilde{M}2$ is the projection of $M2$ but not an E-observer.

4 Nonblocking Observer

Consider a DES described by EFA E. Given a set of local events, we can define the transition projection $\bar{P} : T \times \Sigma_\ell \to T$ and then the EFA projection $\hat{P}(E, \Sigma_\ell)$. Crucial to successful model abstraction using transition projection is that the projected system contains necessary and sufficient information needed for reliable representation of the nonblocking property. In other word, the EFA projection $\hat{P}$ may remove critical information and be inconsistence with the original DES with respect to controllability and nonblocking. For instance, the projection of a blocking DES could be nonblocking, so a nonblocking supervisor designed from the EFA projection could result in a blocking supervisor for the original DES. To avoid this pitfall, one must carefully select the local events of a DES.

A "good" selection of local events for the transition projection $\bar{P}$ is whenever a projected EFA reaches a location by $\bar{P}\varrho_s$ and then to a marker location by $\bar{P}\varrho_t$, the original system, must be able to reach a marker location from $\varrho_{\acute{s}}$, via some $\varrho_{\acute{t}}$ such that $\bar{P}\varrho_s = \bar{P}\varrho_{\acute{s}}$ and $\bar{P}\varrho_{\acute{t}} = \bar{P}\varrho_t$.

Definition 10 (E-observer)**.**
Assume a nonblocking EFA E and let $\Sigma_\ell \subseteq \Sigma$ be the subset of events. The transition projection $\bar{P} : T \times \Sigma_\ell \to T$ is an *E-observer*, if for all initial dynamic execution fragments ϱ_s and $\varrho_{\acute{s}}$ and for all marked dynamic execution fragment ϱ_t in E such that $\varrho_s \sqsubseteq \varrho_t$ and $\bar{P}\varrho_s = \bar{P}\varrho_{\acute{s}}$, there exist a marked dynamic execution fragment $\varrho_{\acute{t}}$ in E such that $\varrho_{\acute{s}} \sqsubseteq \varrho_{\acute{t}}$ and $\bar{P}\varrho_{\acute{t}} = \bar{P}\varrho_t$.

Note that if Σ_ℓ is equal to Σ or $\emptyset$, $\bar{P}$ is automatically an L-observer.

Example 2. *Consider EFAs $M2$ and $\tilde{M}2$ in Fig. 2. Assume the set of local events Σ_ℓ with $\{!o1\} \in \Sigma_\ell$. The shaded circle is the marked location. Define the transition projection $\bar{P} : T \times \Sigma_\ell \to T$ and let $\varrho_s = \ell_0 \overset{tb22}{\rightarrow}_{b2>0/b2:=b1-1} \ell_1$, $\varrho_t = \ell_1 \overset{pb12}{\rightarrow}_{b1<16/b1:=b1+1} \ell_0$, and $\varrho_{\acute{s}} = \ell_0 \overset{tb22}{\rightarrow}_{b2>0/b2:=b1-1} \ell_1 \overset{pb12}{\rightarrow}_{b1<16/b1:=b1+1} \ell_0$ with $\varrho_s \sqsubseteq \varrho_t$ and $\bar{P}\varrho_s = \bar{P}\varrho_{\acute{s}}$. We cannot find any dynamic execution fragment, say $\varrho_{\acute{t}}$ in $M1$ such that $\varrho_{\acute{s}} \sqsubseteq \varrho_{\acute{t}}$ and $\bar{P}\varrho_t = \bar{P}\varrho_{\acute{t}}$. Thus $\bar{P}$ is not an E-observer.*

$$\begin{tikzcd}
\bar{P}_1\varrho \in E_1 \arrow[d, "{\bar{Q}_1[., \Sigma_1 \cap \Sigma_\ell]}"] & \varrho \in E_1 \| E_2 \arrow[l, "{\bar{P}_1[., \Sigma_2 - \Sigma_1]}"'] \arrow[r, "{\bar{P}_2[., \Sigma_1 - \Sigma_2]}"] \arrow[d, "{\bar{P}[., \Sigma_\ell]}"'] & \bar{P}_2\varrho \in E_2 \arrow[d, "{\bar{Q}_2[., \Sigma_2 \cap \Sigma_\ell]}"'] \\
\bar{Q}_1\bar{P}_1\varrho \in \tilde{E}_1 \arrow[dr, "\bar{P}_2"'] & \bar{P}\varrho \in E_1 \tilde{\|} E_2 \arrow[l, "\bar{R}_1 := \bar{Q}_1 \circ \bar{P}_1"'] \arrow[r, "\bar{R}_2 := \bar{Q}_2 \circ \bar{P}_2"] \arrow[d, "\bar{Z}"'] & \bar{Q}_2\bar{P}_2\varrho \in \tilde{E}_2 \arrow[dl, "\bar{P}_1"] \\
 & \bar{Z}\varrho \in E_1 \times E_2 &
\end{tikzcd}$$

Figure 3: Commutative diagram of transition projections for a dynamic execution fragment $\varrho \in (E_1 \| E_2)$ with respect to the set of local events $\Sigma_\ell \subseteq \Sigma_1 \cup \Sigma_2$. Here, $\bar{Z}$ is defined $\bar{Z}[., (\Sigma_1 \cup \Sigma_2) - (\Sigma_1 \cap \Sigma_2)]$.

Proposition 3.
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, \eta^0, L_k^m, D_k^m), k = 1, 2$ be two nonblocking EFAs over the set of shared variables $\mathcal{V}$. Consider T as the set of transition relation for $E_1 \| E_2$. Define the transition projections $\bar{P} : T \times \Sigma_\ell \to T$ and $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ $(i = 1, 2)$ where $\Sigma_\ell \subseteq \Sigma_1 \cup \Sigma_2$. If Σ_ℓ is the set of local events and for both $i = 1, 2, \bar{Q}_i$ is an E_i-observer for E_i, then $\bar{P}$ is an E-observer for $E_1 \| E_2$.

The proof needs the following Lemma.

Lemma 2. *Define the transition projection $\bar{P}_i$ and $\bar{Q}_i$ as in the notation of Proposition 3 for $i = 1, 2$. For any two EFAs E_1 and E_2 with $\Sigma_1 \cap \Sigma_2 \neq \emptyset$ we have $E_1 \| E_2 \neq \emptyset \Leftrightarrow \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \neq \emptyset$.*

> *Proof.*
> $(\Rightarrow)$ Since $E_1 \| E_2 \neq \emptyset$, there exists some dynamic execution fragment $\varrho \in E_1 \| E_2$ and $\bar{P}_i\varrho \in E_i (i = 1, 2)$. Applying the transition projection $\bar{Q}_i$ on both side, we get $\bar{Q}_i P_i \varrho \in \hat{Q}_i[E_i, \Sigma_i \cap \Sigma_\ell]$. Based on Proposition 2, $\bar{Q}_i P_i \varrho = \bar{P}\varrho (i = 1, 2)$. Hence, it implies $\bar{P}\varrho \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]$ and $\bar{P}\varrho \in \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. Therefore, $\bar{P}\varrho \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \neq \emptyset$.
>
> $(\Leftarrow)$ Since $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \neq \emptyset$, we can select a dynamic execution fragment $\acute{\varrho} \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. Then, there must be two dynamic execution fragments $\varrho_1 \in E_1$ and $\varrho_2 \in E_2$ such that $\bar{Q}_1\varrho_1 = \acute{\varrho} = \bar{Q}_2\varrho_2$. From ϱ_1 and ϱ_2 we can construct a set of dynamic execution fragments $\Pi := \{\grave{\varrho} \in E_1 \| E_2 \mid \bar{P}_1\grave{\varrho} = \varrho_1 \wedge \bar{P}_2\grave{\varrho} = \varrho_2\}$. Evidently, taking any dynamic execution fragment $\grave{\varrho} \in \Pi$ we can see that $\hat{P}_1\grave{\varrho} = \varrho_1, \hat{P}_2\grave{\varrho} = \varrho_2$, and $\grave{\varrho} \in E_1 \| E_2$ so $E_1 \| E_2 \neq \emptyset$.
>
> □

Returning to the proof of Proposition 3, we bring in the corresponding transition projections defined in Fig. 3.

Proof of Proposition 3. Let $\bar{P}_i : T \times (\Sigma_j - \Sigma_i) \to T$ $(j \neq i)$ and $\bar{R}_i := \bar{Q}_i \circ \bar{P}_i$ for $i, j = 1, 2$. Let $\varrho_s, \varrho_{\acute{s}}, \varrho_t \in E_1 \| E_2$ be the dynamic execution fragments in notation of Definition 10. To justify the statement, we must find a marked dynamic execution fragment $\varrho_{\acute{t}} \in E_1 \| E_2$ such that $\bar{P}\varrho_t = \bar{P}\varrho_{\acute{t}}$ and $\varrho_{\acute{s}} \sqsubseteq \varrho_{\acute{t}}$. We have $\bar{P}\varrho_s, \bar{P}\varrho_{\acute{s}}, \bar{P}\varrho_t \in \hat{P}[E_1 \| E_2, \Sigma_\ell]$. Consequently, for $i = 1, 2$, $\bar{R}_i\bar{P}\varrho_t \in \hat{R}_i\hat{P}[E_1 \| E_2, \Sigma_\ell] = \hat{Q}_i[\hat{P}_i[E_1 \| E_2, \Sigma_j - \Sigma_i], \Sigma_i \cap \Sigma_\ell] \subseteq \hat{Q}_i[E_i, \Sigma_i \cap \Sigma_\ell]$ and $\bar{P}_i(\varrho_s), \bar{P}_i(\varrho_{\acute{s}}) \in E_i$. Because $\varrho_s \sqsubseteq \varrho_t$, We can apply $\bar{R}_i\bar{P}$ on both sides, to get $\bar{R}_i\bar{P}\varrho_s \sqsubseteq \bar{R}_i\bar{P}\varrho_t$. Now that $\bar{R}_i\bar{P}\varrho_t \in \hat{Q}_i[E_i, \Sigma_i \cap \Sigma_\ell], \bar{Q}_i\bar{P}_i\varrho_s = \bar{Q}_i\bar{P}_i\varrho_{\acute{s}}$, and $\bar{P}_i\varrho_s, \bar{P}_i\varrho_{\acute{s}} \in E_i$, we can conclude by the hypothesis of the proposition that $\bar{Q}_i$ is an E_i-observer, so there exist a marked $\varrho_{\acute{t}_i}$ in E_i such that $\bar{P}_i\varrho_{\acute{s}} \sqsubseteq \varrho_{\acute{t}_i}$ and $\bar{Q}_i\varrho_{\acute{t}_i} = \bar{R}_i\bar{P}\varrho_t$. Apply $\bar{P}_j(j = 1, 2; j \neq i)$ on both sides of equation to get $\bar{P}_j[\bar{Q}_i\varrho_{\acute{t}_i}] = \bar{P}_j[\bar{R}_i\bar{P}\varrho_t]$. Since $\Sigma_\ell \subseteq (\Sigma_1 - \Sigma_2) \cup (\Sigma_2 - \Sigma_1)$ therefore, $\bar{P}_j\bar{Q}_i\varrho_{\acute{t}_i} = \bar{P}_j\varrho_{\acute{t}_i}$ and $\bar{P}_j\bar{R}_i\bar{P}\varrho_t = \bar{Z}\varrho_t$. Hence, $\bar{P}_2\bar{Q}_1\varrho_{\acute{t}_1} = \bar{Z}\varrho_t = \bar{P}_1\bar{Q}_2\varrho_{\acute{t}_2} \Rightarrow \bar{P}_2\varrho_{\acute{t}_1} = \bar{P}_1\varrho_{\acute{t}_2}$. Suppose a set of marked dynamic execution fragments $\Pi := \{\varrho_w \in E_1 \| E_2 \mid \bar{P}_1\varrho_w = \varrho_{\acute{t}_1} \wedge \bar{P}_2\varrho_w = \varrho_{\acute{t}_2}\}$. Recall that $\varrho_{\acute{t}_i} \in E_i$ then by Lemma 2, $\Pi \neq \emptyset$. Apply $\bar{P}$ to ϱ_w we can see $\bar{P}\varrho_w \in \hat{P}[E_1 \| E_2, \Sigma_\ell] = \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. We have $\bar{P}_i\varrho_{\acute{s}} \sqsubseteq \varrho_{\acute{t}_i}$ so taking any dynamic execution fragment from Π, say $\varrho_w \in \Pi$, we see immediately $\varrho_{\acute{s}} \sqsubseteq \varrho_w$ and $\bar{P}_i\varrho_w = \varrho_{\acute{t}_i}$. We know $\bar{Q}_i\varrho_{\acute{t}_i} = \bar{R}_i\bar{P}\varrho_t$. Consequently, $\bar{P}\varrho_w = \bar{P}\varrho_t$ which we can conclude that $\bar{P}$ is an E-observer for $E_1 \| E_2$. □

As we establish a "reliable interface" for EFAs by introducing E-observer, the interaction between two complex system may be examined through their projections rather than their global behavior. If $\bar{P}$ has the observer property, we can check if two EFAs E_1 and E_2 are synchronously nonconflicting by checking whether their projections $\hat{P}[E_1, \Sigma_1 \cap \Sigma_\ell]$ and $\hat{P}[E_2, \Sigma_2 \cap \Sigma_\ell]$ are synchronously nonconflicting. Since the EFA models of $\hat{P}[E_i, \Sigma_i \cap \Sigma_\ell]$ are smaller than those of E_i, we may save significant computational effort, in accordance with the following.

Theorem 2 (Synchronously Nonconflicting Criterion)**.** *Let $E_k(k = 1, 2)$, be two EFAs with the set of shared variables $\mathcal{V}$ and let Σ_ℓ be the set of local events. If $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ are E_i-observer $(i = 1, 2)$, then $E_1 \| E_2$ is nonblocking if and only if $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ is nonblocking.*

Proof. Define the transition projections $\bar{P}_i : T \times (\Sigma_j - \Sigma_i) \to T$ $(j \neq i), \bar{Z} : T \times (\Sigma_1 \cap \Sigma_2) \to T$, and $\bar{R}_i := \bar{Q}_i \circ \bar{P}_i(i, j = 1, 2)$ for the EFA projections in the commutative diagram illustrated by Fig. 3.

> (**If**) Let ϱ_s be a initial dynamic execution path in $E_1 \| E_2$. We must show that there exists a marked dynamic execution path ϱ_t such that $\varrho_s \sqsubseteq \varrho_t$. Apply $\bar{P}_i$ to ϱ_s, we get $\bar{P}_i\varrho_s \in E_i(i = 1, 2)$. Moreover,

$\bar{P}(\varrho_s) \in \hat{P}[E_1 \| E_2, \Sigma_\ell]$. Because of assumption that Σ_ℓ is the set of local events and Proposition 1, $\bar{P}\varrho_s \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. Then there must exists a marked execution fragment $\acute{\varrho}_t$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$, and therefore $\bar{R}_i\bar{P}\varrho_s$ and $\bar{R}_i\acute{\varrho}_t$ are in $\hat{Q}_i[E_i, \Sigma_i \cap \Sigma_\ell](i = 1, 2)$. From Fig. 3, we see that $\bar{R}_i \circ \bar{P} = \bar{Q}_i \circ \bar{P}_i(i = 1, 2)$. Consequently, both $\bar{Q}_i\bar{P}_i\varrho_s$ and $\bar{R}_i\acute{\varrho}_t$ are in $\hat{Q}_i[E_i, \Sigma_i \cap \Sigma_\ell](i = 1, 2)$. Since $\bar{P}_i\varrho_s \in E_i$ and $\bar{Q}_i$ is an E_i-observer, there exists a marked dynamic execution fragment $\varrho_{w_i} \in E_i$ such that $\bar{P}_i\varrho_s \sqsubseteq \varrho_{w_i}$ and $\bar{Q}_i\varrho_{w_i} = \bar{R}_i\acute{\varrho}_t$. Applying $\bar{P}_j(j = 1, 2; j \neq i)$ to both sides of this equation, we get $\bar{P}_j\bar{Q}_i\varrho_{w_i} = \bar{P}_j\bar{R}_i\acute{\varrho}_t = \bar{Z}\acute{\varrho}_t$. This implies that $\bar{P}_2\varrho_{w_1} = \bar{Z}\acute{\varrho}_t = \bar{P}_1\varrho_{w_2}$. Therefore, $\Pi := \{\varrho_w \in E_1 \| E_2 \mid \bar{P}_1\varrho_w = \varrho_{\acute{w}_1} \wedge \bar{P}_2\varrho_w = \varrho_{\acute{w}_2}\}$ is nonempty by Lemma 2. Taking any marked dynamic execution fragment form the set Π, say $\varrho_w \in \Pi$, we have $\bar{P}_i\varrho_w = \varrho_{w_i}(i = 1, 2)$. Since $\varrho_{w_i} \in E_i$, we have $\bar{P}_i\varrho_w \in E_i(i = 1, 2)$. Consequently, $\varrho_w \in E_1 \| E_2$, and as required ϱ_w is marked and $\varrho_s \sqsubseteq \varrho_w$.

(**Only if**) According to assumption $E_1 \| E_2$ is nonblocking, therefore for any initial dynamic execution fragment ϱ_s there exists a marked dynamic execution fragment ϱ_t such that $\varrho_s \sqsubseteq \varrho_t$. Apply $\bar{P}$ to both ϱ_s and ϱ_t,we get $\bar{P}\varrho_s$ and $\bar{P}\varrho_t$ in $\hat{P}[E_1 \| E_2, \Sigma_\ell]$ and by Proposition 2 in $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. Since $\bar{P}$ is E-observer there must exist a marked execution fragment $\acute{\varrho}_t \in \hat{P}[E_1 \| E_2, \Sigma_\ell]$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$ and $\bar{P}\acute{\varrho}_t = \bar{P}\varrho_t$. By Proposition 2, $\acute{\varrho}_t \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. Therefore, for any dynamic execution fragment $\bar{P}\varrho_s \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ there exists a marked dynamic execution fragment $\acute{\varrho}_t$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$ hence $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ is also nonblocking.

□

In case the two EFAs E_1 and E_2 are synchronously conflicting, a third EFA E, called a coordinator, must be introduced to resolve the conflict. We can now, instead of computing the coordinator directly from the two EFAs themselves, we perform this computation through their abstractions.

Proposition 4. *Let $E_k(k = 1, 2)$, be two EFAs with the set of shared variables $\mathcal{V}$ and let Σ_ℓ be the set of local events. In notation of Theorem 2, if for $i = 1, 2$, $\bar{Q}_i$ is an E_i-observer and there is an EFA E such that $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \| E$ is nonblocking then $E_1 \| E_2 \| E$ is also nonblocking.*

The coordinator E alphabet depends only upon the event set $(\Sigma_1 \cup \Sigma_2) - \Sigma_\ell$ which contains the shared events of Σ_1 and Σ_2 and defines the required E-observer. As long as E can resolved the conflict between $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]$ and $\hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$, it can resolve the conflict between E_1 and E_2.

Proof. Let $\acute{E} := \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \| E$ which is by assumption nonblocking. Consequently, $E_1 \| E_2 \| \acute{E} = (E_1 \| \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]) \| (E_2 \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]) \| E = E_1 \| E_2 \| \acute{E}$. The proposition is now reduced to showing that $E_1 \| E_2 \| \acute{E}$ is nonblocking. We first show that E_1 and E_2 are each synchronously nonconflicting with $\acute{E}$. For $i = 1, 2$, since $\hat{Q}_i(E_i, \Sigma_i \cap \Sigma_\ell) \| \acute{E} = \acute{E}$ is nonblocking, we can claim that $Q_i(E_i, \Sigma_i \cap \Sigma_\ell)$ and $\acute{E}$ are synchronously nonconflicting, i.e., $\hat{Q}_i(E_i, \Sigma_i \cap \Sigma_\ell) \| \acute{E}$ is nonblocking. According to the assumption that $\bar{Q}_i$ is an E_i-observer, Theorem 2 ensures that E_i and $\acute{E}$ are synchronously nonconflicting, i.e., $E_i \| \acute{E}$ is nonblocking. Let $J_i := E_i \| \acute{E} \subseteq E_1 \| E_2 (i = 1, 2)$. By Proposition 3, $\bar{P}$ is also a J_i-observer. Because synchronous product is associative and commutative, $E_1 \| E_2 \acute{E} = (E_1 \| \acute{E}) \| (E_2 \| \acute{E}) = J_1 \| J_2$. Next we show that J_1 and J_2 are also synchronously nonconflicting. By Proposition 2, $\hat{P}[J_i, \Sigma_\ell] = \hat{P}[E_i \| \acute{E}, \Sigma_\ell] = \hat{Q}_i[E_i, \Sigma_i \cap \Sigma_\ell] \| \acute{E} (i = 1, 2)$. Hence, $\hat{P}[J_1, \Sigma_\ell] \| \hat{P}[J_2, \Sigma_\ell] = \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_i, \Sigma_2 \cap \Sigma_\ell] \| \acute{E} = \acute{E}$ and is nonblocking. Therefore, $\hat{P}[J_1, \Sigma_\ell]$ and $\hat{P}[J_2, \Sigma_\ell]$ are nonconflicting. Since $\bar{P}$ is J_1 and J_2-observer, we can conclude that J_1 and J_2 are synchronously nonconflicting, namely, $J_1 \| J_2$ is nonblocking. Using the definition of $J_i (i = 1, 2)$ in the above equation, we get $E_1 \| E_2 \| \acute{E}$ is nonblocking. □

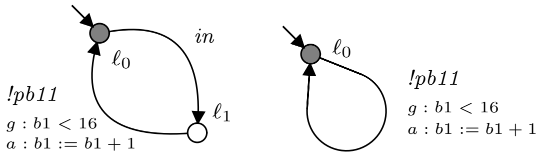
5 Optimal Nonblocking and Controllable Supervisor

An optimal supervisor with full observation usually disables the nearest controllable events preceding or upstream to a prohibited uncontrollable event (say, σ). If, however, some of these controllable events are unobservable, a decentralized supervisor must disable controllable events further back, and so is more restrictive. For this restriction to be relaxed, the local event set must be selected properly enough to contain all the upstream controllable events nearest to σ. Such a decentralized supervisor will prevent the occurrence of an uncontrollable event while allowing maximal freedom of system behavior. A projection with such a local event set is called output control consistent (OCC).

Definition 11 (OCC).
Let $E = (L, D, \Sigma, T, L^0, D^0, L^m, D^m)$ be an EFA over the set of variables $\mathcal{V}$ and let $\Sigma_\ell, \Sigma_u \subseteq \Sigma$ be the local and uncontrollable event sets. The transition projection $\bar{P} : T \times \Sigma_\ell \to T$ is *output control consistent (OCC)* for the EFA E, if for every finite dynamic execution fragment ϱ of the form

$$\varrho = \ell_0 \overset{\sigma_1}{\rightarrow}_{g_1/a_1} \cdots \overset{\sigma_{i+1}}{\rightarrow}_{g_{i+1}/a_{i+1}} \ell_{i+1} \text{ or}$$
$$\varrho = \ell \overset{\sigma}{\rightarrow}_{g/a} \ell_0 \overset{\sigma_1}{\rightarrow}_{g_1/a_1} \cdots \overset{\sigma_{i+1}}{\rightarrow}_{g_{i+1}/a_{i+1}} \ell_{i+1}, 0 \leqslant i < n$$

which satisfies the conditions that $n \geqslant 1, \sigma \in \Sigma - \Sigma_\ell, \sigma_j \in \Sigma_\ell (j \in \mathbf{n\text{-}1})$ and $\sigma_n \in \Sigma - \Sigma_\ell$, we have the property that $\sigma_n \in \Sigma_u \Rightarrow (\forall j \in \mathbf{n}) \sigma_j \in \Sigma_u$.



(a) $M1$ (b) $\tilde{M}1$

Figure 4: $\tilde{M}1$ is the projection of $M1$ but the transition projection is not OCC for $M1$.

In the definition, when σ_n is not local and uncontrollable, its immediately preceding local events must all be uncontrollable, namely, its nearest controllable event must not be unobservable.

Example 3. *Consider EFAs $M1$ and $\tilde{M}1$ in Fig. 4. Assume the set of local and uncontrollable events $\Sigma_\ell, \Sigma_u \subseteq \Sigma$ and let $\{in\} \in \Sigma_\ell$ and $\{!pb11\} \in \Sigma_u$. Define the transition projection $\bar{P} : T \times \Sigma_\ell \to T$. Then, by definition of OCC, event $\{in\}$ must be uncontrollable in $\varrho = \ell_0 \xrightarrow{in} \ell_1 \xrightarrow{pb11}_{b1<16/b1:=b1+1} \ell_0$ because $\{!pb11\}$ is uncontrollable which is not in this case. Thus $\bar{P}$ is not OCC for $M1$.*

We can now state a practical and concise sufficient condition for Optimal Nonblocking and Controllable Supervisor (ONCS).

Theorem 3 (ONCS).
Let R be a nonblocking EFA plant over the set of variables $\mathcal{V}$, along with local and uncontrollable event sets $\Sigma_\ell, \Sigma_u \subseteq \Sigma$, respectively. Define the transition projection $\bar{P} : T \times \Sigma_\ell \to T$ and let EFA $\tilde{R} = \hat{P}[R, \Sigma_\ell]$. Suppose the set of forbidden locations is $L_f \subset \tilde{L}$. If the transition projection $\bar{P}$ is an R-observer and OCC for R, then

$$supS(R, Lf) = supS(\tilde{R}, Lf) \| R$$

Proof. It needs to be shown that the fixed points of the algorithm executions of $supS(R, Lf)$ and $supS(\tilde{R}, Lf)$, $\text{SSEFA}(R, L_f)^N$ and $\text{SSEFA}(\tilde{R}, L_f)^N$, are the same. This can be proved by induction on the step iterator j.

($\subseteq$) **BASE:** $j = 0$. By definition $\text{SSEFA}(\tilde{R}, L_f)^N \subseteq L \times D = \text{SSEFA}(R, L_f)^0$. **INDUCTION:** Assuming that the property holds for j it needs to be shown that it also holds for $j+1$. Let $p = \langle \ell, \eta \rangle \in \text{SSEFA}(\tilde{R}, L_f)^N$. By the inductive assumption it holds that $p \in \text{SSEFA}(R, L_f)^j$. Assume that $p \notin \text{SSEFA}(R, L_f)^{j+1}$. This implies that either p is (α) uncontrollable or (β) blocking and removed by the algorithm. (α) Then there exists $v \in \Sigma_u$ such that $p \overset{v}{\mapsto}_R q \notin \text{SSEFA}(R, L_f)^j$ for some $q := \langle \acute{\ell}, \acute{\eta} \rangle \in L \times D$. Assume $v \in \Sigma_u \cap (\Sigma - \Sigma_\ell)$. Then v is not local so the same transition exists in $\tilde{R}$. Therefore, $p \overset{v}{\mapsto}_{\tilde{R}} q \notin$

SSEFA$(R, L_f)^j \supseteq$ SSEFA$(\tilde{R}, L_f)^N$. But then $p \notin$ SSEFA$(\tilde{R}, L_f)^N$ which is in contradiction. Now, assume $v \in \Sigma_u \cap \Sigma_\ell$. Then for all sequences of consecutive transitions of the form $p \overset{\sigma_1}{\mapsto}_R \cdots \overset{\sigma_k}{\mapsto}_R q$ $(k \geq 1)$ in SSEFA$(\tilde{R}, L_f)^j$ which satisfies the conditions $v = \sigma_1, \ldots, \sigma_i \in \Sigma_\ell(i \in$ **k-1**) and $\sigma_k \in \Sigma - \Sigma_\ell$, if we have $\sigma_k \in \Sigma_u$ then by the assumption that $\bar{P}$ is OCC for R we can immediately see $(\forall i \in \mathbf{k})\sigma_i \in \Sigma_u$. Consequently, $p \overset{\sigma_k}{\mapsto}_{\tilde{R}} r \notin$ SSEFA$(R, L_f)^j \supseteq$ SSEFA$(\tilde{R}, L_f)^N$ and $p \notin$ SSEFA$(\tilde{R}, L_f)^N$ which is in contradiction. Otherwise, if we have $\sigma_k \in \Sigma_c$ then the same transition $p \overset{\sigma_k}{\mapsto}_R q \notin$ SSEFA$(R, L_f)^j$ exists in $\tilde{R}$ and similarly $p \overset{\sigma_k}{\mapsto}_{\tilde{R}} q \notin$ SSEFA$(R, L_f)^j \supseteq$ SSEFA$(\tilde{R}, L_f)^N$ which implies $p \notin$ SSEFA$(\tilde{R}, L_f)^N$ which is in contradiction. (β) Then $p \overset{s}{\mapsto}_{\text{SSEFA}(R,L_f)^j} q$ implies $q \notin L^m \times D^m$ for all states $q \in L \times D$ and $s \in \Sigma^*$. If $s \in (\Sigma - \Sigma_\ell)^*$ then also $p \notin$ SSEFA$(\tilde{R}, L_f)^N$ which is a contradiction. It may be the case that $s \in \Sigma_\ell^*$, Because $\bar{P}$ is a R-observer and we know $p \overset{\acute{t}}{\mapsto}_{\text{SSEFA}(\tilde{R},L_f)^N} r$ for some $r \in L^m \times D^m$ and $\acute{t} \in \Sigma_\ell^*$, then there is a string $t \in \Sigma^*$ such that $q \overset{t}{\mapsto}_{\text{SSEFA}(R,L_f)^j} l$ for some $l \in L^m \times D^m$. Hence, $p \overset{st}{\mapsto}_{\text{SSEFA}(R,L_f)^j} l$ and therefore, $p \in$ SSEFA$(R, L_f)^j$. This contradicts the initial assumption.

$(\supseteq)$ **BASE:** $j = 0$. By definition SSEFA$(R, L_f)^N \subseteq L \times D =$ SSEFA$(\tilde{R}, L_f)^0$. **INDUCTION:** Assuming that the property holds for j it needs to be shown that it also holds for $j+1$. Let $p = \langle \ell, \eta \rangle \in$ SSEFA$(R, L_f)^N$. By the inductive assumption it holds that $p \in$ SSEFA$(\tilde{R}, L_f)^j$. Assume that $p \notin$ SSEFA$(\tilde{R}, L_f)^{j+1}$. This implies that either p is (α) uncontrollable or (β) blocking and removed by the algorithm.

(α) Then there exists $v \in \Sigma_u \cap (\Sigma - \Sigma_\ell)$ such that $p \overset{v}{\mapsto}_{\tilde{R}} q \notin$ SSEFA$(\tilde{R}, L_f)^j$ for some $q := \langle \acute{\ell}, \acute{\eta} \rangle \in L \times D$. Let a sequence of consecutive transitions in SSEFA$(R, L_f)^N$ be the form $p \overset{\sigma_1}{\mapsto}_R \cdots \overset{\sigma_k}{\mapsto}_R q$ $(k \geq 1)$ such that $\sigma_i \in \Sigma_\ell(i \in$ **k-1**) and $v = \sigma_k \in \Sigma_u \cap (\Sigma - \Sigma_\ell)$. Then immediately by definition 11 we have, $\sigma_i \in \Sigma_u$. Hence, there is a transition $p \overset{\sigma_1}{\mapsto}_R r \in$ SSEFA$(R, L_f)^N$ that implies also $p \notin$ SSEFA$(R, L_f)^N$ which is a contradiction.

(β) Then $p \overset{s}{\mapsto}_{\text{SSEFA}(\tilde{R},L_f)^j} q$ implies $q \notin L^m \times D^m$ for all states $q \in L \times D$ and $s \in (\Sigma - \Sigma_\ell)^*$. Let a sequence of consecutive transitions in SSEFA$(R, L_f)^N$ be the form $p \overset{\sigma_1}{\mapsto}_R \cdots \sigma_{k-1} \mapsto_R l \overset{s}{\mapsto}_R q$ $(k \geq 1)$ where $\sigma_i \in \Sigma_\ell(i \in$ **k-1**) and $s \in (\Sigma - \Sigma_\ell)^*$. Then there is a sequence of transitions in SSEFA$(R, L_f)^N$, say $p \overset{\acute{s}}{\mapsto}_{\text{SSEFA}(R,L_f)^N} r$ with $\acute{s} = \sigma_1 \ldots \sigma_i$, such that $r \in L^m \times D^m$ for some $r \in L \times D$. Because $\bar{P}$ is a R-observer, then there is a $p \overset{u}{\mapsto}_{\text{SSEFA}(\tilde{R},L_f)^j} l$ such that $l \in L^m \times D^m$ and $u \in (\Sigma - \Sigma_\ell)^*$. Hence, $p \overset{su}{\mapsto}_{\text{SSEFA}(\tilde{R},L_f)^j} l$ and therefore,

$p \in \text{SSEFA}(\tilde{R}, L_f)^j$. This contradicts the initial assumption that $p \notin \text{SSEFA}(\tilde{R}, L_f)^j$.

□

We can extend Theorem 3 to Proposition 5 to accommodate systems composed from two components.

Proposition 5. *Let E_1 and E_2 be two nonblocking EFAs over the set of shared variables $\mathcal{V}$, along with local and uncontrollable event sets $\Sigma_\ell, \Sigma_u \subseteq \Sigma := \Sigma_1 \cup \Sigma_2$, respectively, and let $R := E_1 \| E_2$. Define the transition projections $\bar{P} : T_R \times \Sigma_\ell \to T_R$,$\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ $(i = 1, 2)$ and let EFA $\tilde{R} = \hat{P}[E_1 \| E_2, \Sigma_\ell]$. Suppose the set of forbidden locations is $L_f \subset \tilde{L}_R$. If for $i = 1, 2$, $\bar{Q}_i$ is an E_i-observer and OCC for E_i then*

$$supS(E_1 \| E_2, Lf) = supS(\hat{P}[E_1 \| E_2, \Sigma_\ell], Lf) \| E_1 \| E_2$$

Proof.
($\supseteq$) Let $E := \text{SSEFA}(E_1 \| E_2, L_f)$ and $\tilde{E} := \text{SSEFA}(\hat{P}[E_1 \| E_2, \Sigma_\ell], L_f)$. Then, $\tilde{E} \subseteq \hat{P}[E_1 \| E_2, \Sigma_\ell] = \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. Because Q_i is an E_i-observer $(i = 1, 2)$, by Proposition 3 $\tilde{E}$ is nonblocking. Moreover, $\tilde{E}$ is controllable with respect to $\hat{P}[E_1 \| E_2, \Sigma_\ell]$ and therefore, $\tilde{E} \| E_1 \| E_2$ is controllable with respect to $\hat{P}[E_1 \| E_2, \Sigma_\ell] \| E_1 \| E_2 = E_1 \| E_2$. Because $\tilde{E} \subseteq \hat{P}[E_1 \| E_2, \Sigma_\ell]$, we know $\tilde{E} \| E_1 \| E_2 \subseteq \hat{P}[E_1 \| E_2, \Sigma_\ell] \| E_1 \| E_2$. Since we already know that $\tilde{E} \| E_1 \| E_2$ is controllable with respect to $E_1 \| E_2$, $\tilde{E} \subseteq E$.

($\subseteq$) While the condition that the $\bar{Q}_i$ are E_i-observers $(i = 1, 2)$ implies that $\bar{P}$ is an E-observer, the condition that $\bar{Q}_i$ are OCC for $E_i (i = 1, 2)$ does not imply that $\bar{P}$ is OCC for $E_1 \| E_2$ so the result of Theorem 3 is not applicable. Turning to induction on the fixed points of $\text{SSEFA}(E_1 \| E_2, L_f)^N$ and $\text{SSEFA}(\hat{P}[E_1 \| E_2, \Sigma_\ell], L_f)^N$, for the uncontrollability part, we see it follows the same arguments as in Theorem 3:($\subseteq$):(α) for the transitions $p \overset{v_i}{\hookrightarrow}_{\tilde{R}} q_i \notin \text{SSEFA}(\hat{P}[E_1 \| E_2, \Sigma_\ell], L_f)^j$ where $v_i \in \Sigma_c (i = 1, 2)$, v_1,v_2 are local to E_1 and E_2, respectively, and for some state $q_1, q_2 \in L_1 \times L_2$ therefore, is left out. For the case $v \in \Sigma_u$ and v is shared in E_1 and E_2, we have $p \overset{v}{\hookrightarrow}_{\tilde{R}} q \notin \text{SSEFA}(\hat{P}[E_1 \| E_2, \Sigma_\ell], L_f)^j$ for some state $q \in L_1 \times L_2$. Since v is not local so the same transition exists in R thus $p \overset{v}{\hookrightarrow}_{\tilde{R}} q \notin \text{SSEFA}(E_1 \| E_2, L_f)^j \supseteq \text{SSEFA}(\hat{P}[E_1 \| E_2, \Sigma_\ell], L_f)^N$ then $p \notin \text{SSEFA}(\hat{P}[E_1 \| E_2, \Sigma_\ell], L_f)^N$ which is in contradiction

□

By an argument similar to that for Theorem 3 we can further extend the Proposition 5 for n number of EFAs as follows.

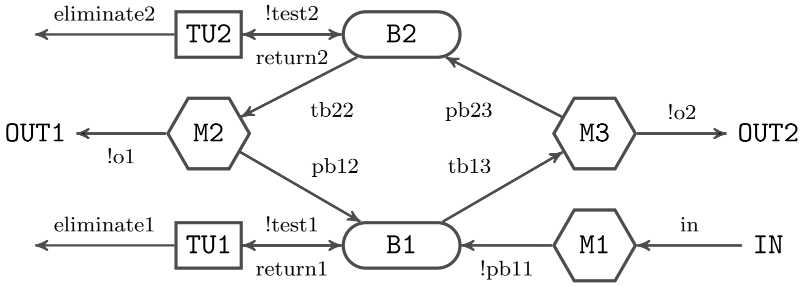


Figure 5: The manufacturing workcell control flow.

Corollary 1. *Let $R := E_1 \| E_2 \| \cdots \| E_n$ be the plant consist of $n \geq 2$ non-blocking components. Assume the set of local and uncontrollable events $\Sigma_\ell, \Sigma_u \subseteq \Sigma := \bigcup_{i=1}^{n} \Sigma_i$, respectively. Define the $\bar{P} : T \times \Sigma_\ell \to T, \bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$. Let $\tilde{R} := \hat{P}[R, \Sigma_\ell]$ and the set of forbidden location be $L_f \subset \tilde{L}$. If for $i \in \boldsymbol{n}, Q_i$ is an E_i-observer and OCC for E_i, then*

$$supS(R, Lf) = supS(\tilde{R}, Lf) \| R$$

Proof. The proof is similar to that of Proposition 5 by considering E_2 as $E_2 \| \cdots \| E_n$. □

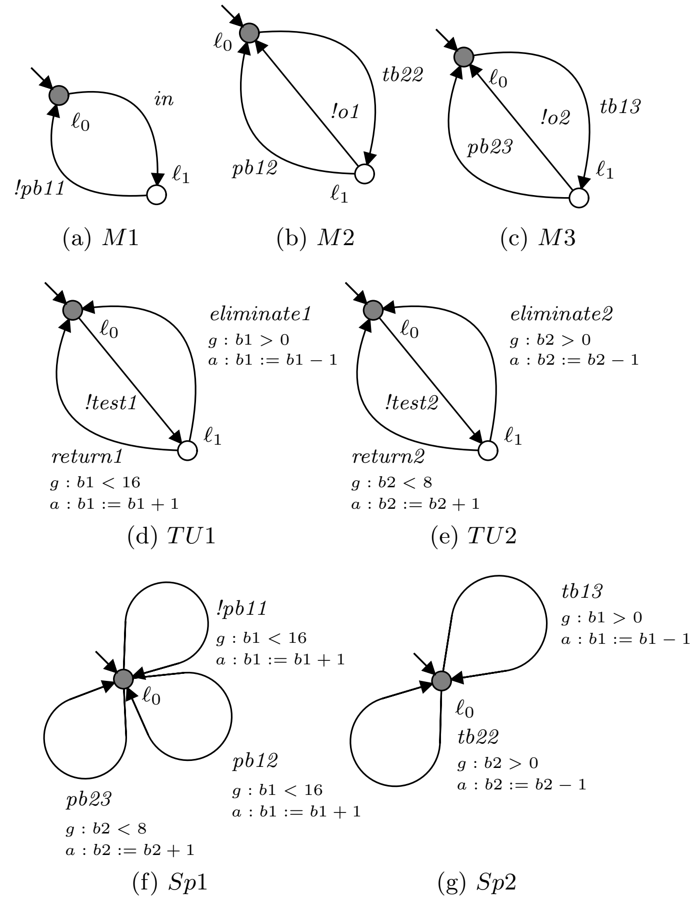
This property was pointed out by [21, 38] and later in more general form by [35], and Corollary 1 extends it to systems modeled by EFAs.

6 Example

6.1 Manufacturing Wokrcell

Consider a manufacturing workcell, borrowed from [14], consisting of three machines M1, M2, and M3, working on parts stored in two buffers B1 and B2 of size 16 and 8, respectively. Parts are supplied through an input buffer IN (of infinite size) and stored after being processed in two output buffers OUT1 and OUT2 (of infinite size). M1 supplies B1 with parts taken from the input buffer IN; M2 takes a part from B2 and after processing puts it either in OUT1 or in B1; and M3 takes a part from B1 and after processing puts it either in OUT2 or in B2. To increase the practical usage and complexity of the cell, two inspection unites TU1 and TU2 is added to randomly inspecting parts from B1 and B2. Parts which are qualified will be returned to the buffer otherwise will be eliminated. Fig. 5 shows the workcell control flow and Fig. 6 illustrates the EFA models of the system.

In Fig. 6, the events with exclamation mark are the uncontrollable events and shaded circles are the marked locations. The domain of the variable $b1$ and $b2$ is $D_1 = \{1, 2, \ldots, 16\}$ and $D_2 = \{1, 2, \ldots, 8\}$, respectively, which

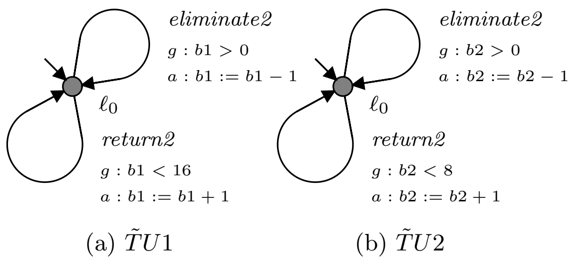


(a) $M1$ (b) $M2$ (c) $M3$

(d) $TU1$ (e) $TU2$

(f) $Sp1$ (g) $Sp2$

Figure 6: EFAs of the manufacturing workcell



(a) $\tilde{T}U1$ (b) $\tilde{T}U2$

Figure 7: Abstracted EFA models of TU1 and TU2 by using algorithm 1 and the set of local events $\Sigma_\ell = \{$*!test1, !test2*$\}$.

indicates the number of parts in the two buffers and their maximum capacity. B1 and B2 initially contain no part, i.e. $D^0 = \{(0,0)\}$, and all values are marked $D^m = D_1 \times D_2$. The workcell specifications are as follows. SPEC1:

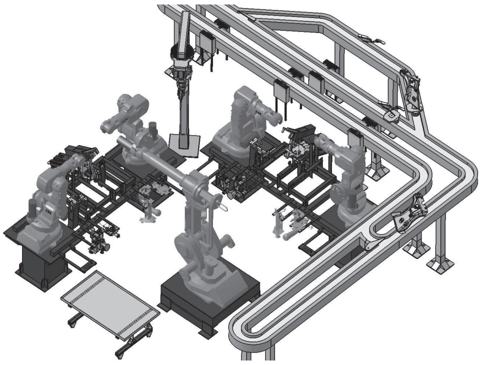


Figure 8: The robot workcell.

Table 1: Optimal nonblocking supervisory synthesis results of the manufacturing workcell example

	Reachable States	Supervisor States
Original Models	4896	4752
Abstracted Models	1224	1188

buffers B1 and B2 must not overflow, i.e., a machine must not try to put a part in a buffer when it is full, i.e., when $b1 = 16$ or $b2 = 8$. SPEC2: buffers B1 and B2 must not underflow, i.e., a machine must not try to take a part from a buffer when it is empty, i.e., when $b1 = 0$ or $b2 = 0$. The EFAs Sp1 and Sp2 of SPEC1 and SPEC2 are depicted in Fig. 6(f) and (g), respectively.

To apply the model abstraction using transition projection as mentioned earlier, first we find the local events in the system by checking the conditions in Definition 9 for all the events. The first candidates for the set of local events are {*in, !test1, !test2, !o1, !o2*}. Then, by Definition 10, the events {*!o1*} and {*!o2*} cannot fulfill the E-observer conditions and therefore, are eliminated from the list. Next, by checking the OCC conditions as in Definition 11, the event {*in*} is found to be inconsistent thus is removed. Finally, the list Σ_ℓ = {*!test1, !test2*} together with EFAs TU1 and TU2 are used as the input parameters of Algorithm 1 in order to compute the projected EFAs. The prjected EFAs of TU1 and TU2 by Algorithm 1 are represented in Fig. 7(a) and (b), respectively. The optimal nonblocking and controllable guards added by the algorithm SSEFA is the same as the example in [14]. Table 1 shows the result of optimal nonblocking supervisory synthesis for both original and abstracted models.

Table 2: Optimal nonblocking supervisory synthesis results of the robotic workcell example

	Reachable States	Supervisor States
Original Models	4.023620354×10^9	1.758696194×10^9
Abstracted Models	5.235889×10^6	2.961409×10^6

6.2 Robotic Workcell

The abstraction method described above has been implemented as a toolbox in Sequence Planner [47] software and applied efficiently on sequences of operations [8] for a robot cell at Chalmers Robot and Automation Lab modeled by EFAs. The cell consists of five ABB robots, two fixtures, an AGV, and a conveyor. The desired behavior of the cell is the following: Two parts are loaded by the operator and transported to the robot station by the conveyor. Two robots pick and place the parts on fixture and assemble the parts. After that, the assembled parts are unloaded by the third robot and delivered to a second station for further manipulation. In that station, two other robots pop-rivet the remaining points. Then, the finished product is unloaded by the third robot and transported from the workstation by an AGV, see Fig. 8. A set of local events for all operations are created and checked to be observer and OCC. Then, operation models are projected by Algorithm 1 and a non-blocking supervisor is synthesized by Supremica tool. Table 2 shows the result supervisors for both original and abstracted models. Note that, the significant abstraction that is achieved in tis example is because of the special structure of operation models. For more details regarding operation model and the example refer to [8, 48].

7 Conclusion

In this paper we have extended previous work on model abstraction by natural projection with observer property to include EFA modeling formalism. Transition projection is introduced to substitute natural projection for EFAs by projecting the dynamic transitions without knowing its underlying language. We independently compute the projection of the low-level components without regard to their mutual conflict. Subsequently, to reduce computational complexity, we compute the high-level coordinators based only on abstracted models of the low-level components. Effective and consistent model abstraction is accomplished through transition projections with the observer and OCC properties.

References

[1] P. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal of Control and Optimization*, vol. 25, no. 1, pp. 635–650, 1987.

[2] ——, "The control of discrete event systems," *Proceedings of IEEE, Special Issue on Discrete Event Dynamic Systems*, vol. 77, no. 1, pp. 81–98, 1989.

[3] M. Fabian and A. Hellgren, "PLC-based Implementation of Supervisory Control for Discrete Event Systems," in *37th Decision and Control*, Tampa, FL, USA, 1998.

[4] S. Balemi, G. J. Hoffmann, P. Gyugyi, H. Wong-Toi, and G. F. Franklin, "Supervisory Control of a Rapid Thermal Multiprocessor," *IEEE Transactions on Automatic Control*, vol. 38, no. 7, pp. 1040–1059, 1993.

[5] X.-R. Cao, G. Cohen, A. Giua, W. M. Wonham, and J. H. van Schuppen, "Unity in Diversity, Diversity in Unity: Retrospective and Prospective Views on Control of Discrete Event Systems," *Discrete Event Dynamic Systems*, vol. 12, pp. 253–264, 2002.

[6] M. Skoldstam, K. Åkesson, and M. Fabian, "Modeling of discrete event systems using finite automata with variables," *2007 46th IEEE Conference on Decision and Control*, pp. 3387–3392, 2007.

[7] S. Miremadi, K. Åkesson, and B. Lennartson, "Supervisor Computation and Representation: A Case Study," 2010.

[8] B. Lennartson, K. Bengtsson, C. Yuan, K. Andersson, M. Fabian, P. Falkman, and K. Åkesson, "Sequence planning for integrated product, process and automation design," *Automation Science and Engineering, IEEE Transactions on*, vol. 7, no. 4, pp. 791–802, Oct. 2010.

[9] K. Bengtsson, C. Thorstensson, B. Lennartson, K. ÅKesson, S. Miremadi, and P. Falkman, "Relations identification and visualization for sequence planning and automation design," in *2010 IEEE International Conference on Automation Science and Engineering*, Aug. 2010, pp. 841–848.

[10] M. R. Shoaei, B. Lennartson, and S. Miremadi, "Automatic generation of controllers for collision-free flexible manufacturing systems," in *6th IEEE International Conference on Automation Science and Engineering.* IEEE, Aug. 2010, pp. 368–373.

[11] P. Magnusson, N. Sundström, K. Bengtsson, B. Lennartson, P. Falkman, and M. Fabian, “Planning transport sequences for flexible manufacturing systems,” in *Preprints of the 18th IFAC World Congress*, Milano, Italy, 2011, pp. 9494–9499.

[12] A. Vahidi, M. Fabian, and B. Lennartson, “Efficient supervisory synthesis of large systems,” *Control Engineering Practice*, vol. 14, no. 10, pp. 1157–1167, Oct. 2006.

[13] S. Miremadi, B. Lennartson, and K. Åkesson, “BDD-based Supervisory Control on Extended Finite Automata,” in *Proceedings of the 7th Annual IEEE Conference on Automation Science and Engineering, CASE'11*, Trieste, 2011.

[14] L. Ouedraogo, R. Kumar, R. Malik, and K. Åkesson, “Nonblocking and Safe Control of Discrete-Event Systems Modeled as Extended Finite Automata,” *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 3, pp. 560–569, Jul. 2011.

[15] K. Åkesson, M. Fabian, H. Flordal, and R. Malik, “Supremica—an integrated environment for verification, synthesis and simulation of discrete event systems,” in *Proceedings of the 8th international Workshop on Discrete Event Systems, WODES'08*, Ann Arbor, MI, USA, 2006, pp. 384–385.

[16] K. Åkesson, M. Fabian, H. Flordal, and A. Vahidi, “Supremica—a Tool for Verification and Synthesis of Discrete Event Supervisors,” in *11th Mediterranean Conference on Control and Automation*, Rhodos, Greece, 2003.

[17] S. Miremadi, K. Åkesson, M. Fabian, A. Vahidi, and B. Lennartson, “Solving two supervisory control benchmark problems using Supremica,” in *9th International Workshop on Discrete Event Systems, 2008, WODES 08.*, May 2008, pp. 131–136.

[18] Z. Fei, S. Miremadi, and K. Åkesson, “Efficient Symbolic Supervisory Synthesis and Guard Generation,” in *3rd International Conference on Agents and Artificial Intelligence*, Rome, Italy, 2011.

[19] K. Rohloff and S. Lafortune, “On the computational complexity of the verification of modular discrete-event systems,” in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 1. IEEE, pp. 16–21.

[20] P. Gohari and W. M. Wonham, “On the complexity of supervisory control design in the RW framework.” *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE*

Systems, Man, and Cybernetics Society, vol. 30, no. 5, pp. 643–52, Jan. 2000.

[21] M. H. Queiroz, J. E. R. Cury, and M. de Queiroz, "Modular control of composed systems," in *American Control Conference*, vol. 6, no. June. American Autom. Control Council, Jun. 2000, pp. 4051–4055.

[22] L. S-H. and K. C. Wong, "Structural decentralized control of concurrent discrete-event systems," *European Journal of Control*, pp. 1125–1134, 2002.

[23] K. Schmidt, T. Moor, and S. Perk, "A Hierarchical Architecture for Nonblocking Control of Discrete Event Systems," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control.* IEEE, 2005, pp. 902–907.

[24] K. Schmidt, J. Reger, and T. Moor, "Hierarchical control for structural decentralized DES," in *Discrete event systems 2004 (WODES'04)*, Elsevier Science. Elsevier Science, 2004, p. 279.

[25] Y. Heymann and M. Willner, "On supervisory control of concurrent discrete-event systems," *International J. on Control*, vol. 54, pp. 1119–1142, 1991.

[26] R. C. Hill and D. M. Tilbury, "Modular Supervisory Control of Discrete-Event Systems with Abstraction and Incremental Hierarchical Construction," in *Proceedings of the 8th international Workshop on Discrete Event Systems, WODES'06*, Ann Arbor, MI, USA, Jul. 2006, pp. 399–406.

[27] K. C. Wong and W. M. Wonham, "Hierarchical control of discrete-event systems," *Discrete Event Dynamic Systems*, vol. 6, no. 3, pp. 241–273, Jul. 1996.

[28] R. Leduc, B. Brandin, M. Lawford, and W. M. Wonham, "Hierarchical interface-based supervisory Control-part I: serial case," in *IEEE Transactions on Automatic Control*, vol. 50, no. 9, Orlando, FL, USA, Sep. 2005, pp. 1322–1335.

[29] R. Leduc, M. Lawford, and W. M. Wonham, "Hierarchical interface-based supervisory control-part II: parallel case," in *IEEE Transactions on Automatic Control*, vol. 50, no. 9, Sep. 2005, pp. 1336–1348.

[30] K. C. Wong, "Discrete-event control architecture: An algebraic approach," thesis, Toronto, 1994.

[31] K. C. Wong and W. M. Wonham, "Modular control and coordination of discrete-event systems," *Discrete Event Dynamic Systems*, vol. 8, no. 3, pp. 247–297, Oct. 1998.

[32] ——, "On the Computation of Observers in Discrete-Event Systems," *Discrete Event Dynamic Systems*, vol. 14, no. 1, pp. 55–107, Jan. 2004.

[33] L. Feng and W. M. Wonham, "Computationally Efficient Supervisor Design: Abstraction and Modularity," in *Proceedings of the 8th international Workshop on Discrete Event Systems, WODES'06*, Ann Arbor, MI, USA, Jul. 2006, pp. 3–8.

[34] L. Feng, W. M. Wonham, and P. S. Thiagarajan, "Designing communicating transaction processes by supervisory control theory," *Form. Methods Syst. Des.*, vol. 30, no. 2, pp. 117–141, 2007.

[35] L. Feng and W. M. Wonham, "On the Computation of Natural Observers in Discrete-Event Systems," *Discrete Event Dynamic Systems*, vol. 20, no. 1, pp. 63–102, Oct. 2008.

[36] ——, "Supervisory control architecture for discrete-event systems," *Automatic Control, IEEE Transactions on*, vol. 53, no. 6, pp. 1449–1461, 2008.

[37] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Springer, 2008.

[38] W. M. Wonham, *Supervisory Control of Discrete Event Systems*, Toronto, Canada, 2011.

[39] C. Baier and J.-P. Katoen, *Principles of Model Checking.* The MIT Press, 2008.

[40] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., ser. Series in Computer Science. Addison-Wesley, 2001.

[41] M. de Queiroz and J. E. R. Cury, "Modular Supervisory Control of Large Scale Discrete Event Systems," in *Discrete Event Systems, Analysis and Control*, R. Boel and G. Stremersch, Eds. Kluwer, 2000, pp. 103–110.

[42] K. Schmidt and C. Breindl, "Maximally Permissive Hierarchical Control of Decentralized Discrete Event Systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 723–737, Apr. 2011.

[43] Y. WILLNER and M. HEYMANN, "Supervisory control of concurrent discrete-event systems," *International Journal of Control*, vol. 54, no. 5, pp. 1143–1169, Nov. 1991.

[44] L. Feng, “Computationally efficient supervisor design for discrete-event systems,” Doctor of Philosophy, University of Toronto, 2007.

[45] R. Milner, *Communication and concurrency*, ser. Series in Computer Science. Prentice-Hall, 1989.

[46] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking.* MIT Press, 2000.

[47] E. Ohlson and C. Torstensson, “Development, implementation and testing of Sequence Planner - A concept for modeling of automation systems,” Tech. Rep. EX/2009, Chalmers University of Technology, 2009.

[48] M. R. Shoaei, S. Miremadi, K. Bengtsson, and B. Lennartson, “Reduced-order synthesis of operation sequences,” in *ETFA2011.* IEEE, Sep. 2011, pp. 1–8.