Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

On the Computation of Natural Observers for Extended Finite Automata ⋆

Mohammad Reza Shoaei * **Lei Feng** ** **Bengt Lennartson** *

* *Department of Signals and Systems, Chalmers University of Technology, SE-412 96, Gothenburg, Sweden (email: {shoaei, bengt.lennartson}@chalmers.se)*
** *Department of Machine Design, KTH - Royal Institute of Technology, SE-100 44 Stockholm, Sweden (email: leifeng@md.kth.se)*

Abstract: Compared to finite automata, Extended Finite Automata (EFAs) allows us to efficiently represent discrete-event systems that involve non-trivial data manipulation. However, the complexity of designing supervisors for such systems is still a challenge. In our previous works, we have studied model abstraction for EFAs using natural projections with observer property on events as well as data. In this paper, we provide sufficient conditions for verifying the observer properties and further enhance the EFAs when the property does not hold. To this end, we introduce symbolic simplification techniques for data and generalize existing algorithms in the literature for the events to compute natural observers for EFAs. The importance of this combined abstraction and symbolic simplification method is demonstrated by synthesis of a nonblocking controller for an industrial manufacturing system.

Keywords: Discrete-event systems; hierarchical control; supervisory control theory.

1. INTRODUCTION

Automatically computing the controllers that supervise and coordinate manufacturing plants enhance the performance and reduce the cost of production processes in the industries. The main challenge, however, is to guarantee the safety properties and to avoid deadlocks. *Supervisory Control Theory* (SCT), established by Ramadge and Wonham Ramadge and Wonham [1989], is a formal framework for modeling and control of *Discrete-Event Systems* (DES). Problems that SCT can address include dynamic allocation of resources, the prevention of system blocking, etc. and, within these constraints, maximally permissive system behavior.

Nevertheless, the industrial acceptance of SCT is still scarce. Main issues are the discrepancy between the signal based reality and the event-based SCT framework, the lack of a compact representation of large models, and computational complexity. To overcome these problems, Extended Finite Automata (EFAs), Skoldstam et al. [2007], are introduced to provide efficient and compact modeling solutions for complex DES with data manipulation. Although EFAs ease the modeling experience, SCT analysis is performed on their underlying automata models and therefore, the fundamental obstruction to the development of SCT, i.e., the computational complexity of synthesizing nonblocking supervisors, still remains. Indeed, the nonblocking supervisory control problem for DES is NP-hard, Gohari and Wonham [2000].

Researchers are therefore seeking effective model abstractions for DES, see Feng and Wonham [2008], Schmidt and Breindl [2011], Wong and Wonham [2004]. The most effective model abstraction operator in SCT is the causal reporter map having the observer property, Wong and Wonham [1998]. In Feng and Wonham [2008], a model abstraction, only with natural observers, i.e., natural projections, Wonham [2013], with the observer property, is introduced. However, DES modeled by EFAs, beside events, carry data and therefore, the observer concept needs to be defined for the data as well. In Shoaei et al. [2012a] and Shoaei et al. [2012b], a hierarchical control architecture for EFAs is studied where the abstraction is achieved by natural projections with event and data observer properties. Recently, in Mohajerani et al. [2013], a compositional nonblocking verification of EFAs based on generalization of observation equivalence is also proposed.

In this paper, we provide a sufficient condition for the verification of the observer property for natural projections and address the case where the verification fails, i.e., when natural projections violate event- and/or data-observer conditions. For the latter case, we use a symbolic interpretation and execution technique in Shoaei and Lennartson [2014] and further introduce guard propagation rules for EFAs. Recent experience shows that without these generalizations we would often have limited or none abstraction possibilities for EFAs.

In the case that the event-observer conditions are not fulfilled, we generalize the existing algorithms in the literature to compute projections which respect event-observer property. Finally, we formulate the computation of appropriate natural projections with both event- and data observer properties for the nonblocking control of EFAs by an observer extension algorithm. This algorithm calls the simplification techniques for EFAs and the event set extension algorithm in Feng and Wonham [2008], until an appropriate alphabet extension is found.

The remainder of the paper is organized as follows. Section 2 briefly describes EFAs and their semantics. In Section 3, we introduce natural observers for EFAs and their properties. Section 4 explains the observer computation together with simplification techniques for EFAs. A manufacturing example has been modeled and abstracted in Section 5 and we conclude our work in Section 6. The proof details are referred to the appendix.

⋆ This work was carried out at the Wingquist Laboratory VINN Excellence Center within the Area of Advance – Production at Chalmers University of Technology, supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA).

Copyright © 2014 IFAC

2. PRELIMINARIES

In this section, we recall some basic definitions and concepts to be used later.

2.1 Predicate Logic

Syntax The formulas of our logic are *quantifier-free first-order logic with equality* over a countable set V of *individual variables* $x, y, \ldots$, and a *signature* set Θ consisting of n-ary *function symbols* $f \in \Theta$, where *constants* are denoted by nullary functions, *predicate symbols* $p \in \Theta$ including the *binary equality* symbol $=$, 1, 0, and the propositional connectives $\leftrightarrow, \rightarrow, \wedge, \vee, \neg$. A *term* $t \in T_\Theta(V)$ is a (well formed) expression over symbols in Θ and V. A term is called a *ground term* if it contains no variables. *Formulas* $\phi, \psi, \ldots$ are defined inductively as follows. A formula is either an *atomic formula* $p(t_1, \ldots, t_n)$ where p is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms, a spacial formula $\bot$ (resp. $\top$) which is always false (resp. true), or of the form $\neg\varphi$ or $\varphi \triangleright \psi$ where $\triangleright \in \{\leftrightarrow, \rightarrow, \wedge, \vee\}$ and φ, ψ are formulas.

Semantics Terms and formulas constructed over Θ and V take on meaning when interpreted over a structure called *model*. A model is a pair $\mathcal{M} = (D, \mathcal{I})$ consisting of: A finite and nonempty set D called *domain* (or universe), where we distinguish the values of an individual variable x by a nonempty set $D_x \subseteq D$; and an *interpreter* function $\mathcal{I}$ that assigns an n-ary function $f^{\mathcal{I}} : D^n \to D$ to each n-ary function symbol $f \in \Theta$ where we regard constants (nullary functions) as just elements of D, and an n-ary relation $p^{\mathcal{I}} \subseteq D^n$ to each n-ary predicate symbol $p \in \Theta$.

Fix $\mathcal{I}$ and let D be the domain of variables. We define a *valuation* map $\eta : T_\Theta(V) \to D$ on terms $T_\Theta(V)$ over variables V. A valuation is uniquely determined by its values on V, since V generates $T_\Theta(V)$. Moreover, any map $\eta : V \to D$ extends uniquely to a valuation $\eta : T_\Theta(V) \to D$ by induction. A *substitution* is a mapping $\eta : T_\Theta(V) \to T_\Theta(V)$. For a term t, $\eta(t) = t[x/\eta(x) | \forall x \in V]$ is a new term obtained by "substituting" all (free) occurrences of x_i in t with t_i $(1 \leq i \leq n)$ and we denote by ϵ the empty substitution such that $\epsilon(t) = t$. The substitution is done for all variables in t simultaneously. Furthermore, we write $\eta[x/t]$ (or $\eta[x \mapsto t]$) to denote a new substitution μ constructed from η such that $\mu(x) = t$ and $\mu(y) = \eta(y)$ for $y \neq x$. We also write $\eta[x \mapsto \epsilon]$ to denote that we drop the substitution x/t from η. In this paper, without loss of generality, we consider valuations as substitutions where a valuation substitutes all variables to their ground terms.

The *satisfaction relation* $\vDash$ (also called semantic entailment) is defined inductively on the structure of formulas as usual [see Gallier, 2003]. If $\eta \vDash \varphi$ holds, we say that φ is true (in $\mathcal{M}$) *under valuation* η, or that η *satisfies* φ (in $\mathcal{M}$). If Γ is a set of formulas, we write $\eta \vDash \Gamma$ if $\eta \vDash \varphi$ for $\varphi \in \Gamma$. If φ is true in all models, then we write $\vDash \varphi$ and say that φ is *valid*. Two formulas ϕ, ψ are said to be *logically equivalent*, denoted $\phi \equiv \psi$, if $\vDash \phi \leftrightarrow \psi$.

2.2 Extended Finite Automata

An *Extended Finite Automaton (EFA)* is a finite-state automaton whose transitions are augmented with *data* Skoldstam et al. [2007] to symbolically represent DES. In this paper, we formulate the data flow in systems by means of *conditions*, which are predicate formulas over variables, on transitions.

EFA Syntax The behavior of DES, Wonham [2013] and Cassandras and Lafortune [2008], can be recognized by a finite-state automaton (FA) $G = \langle Q, \Sigma, \mapsto, Q^\circ, Q^m \rangle$ with the (finite) set of states Q, the (nonempty) alphabet Σ, the transition function $\delta : Q \times \Sigma \to Pwr(Q)$, where Pwr is the power set, the set of initial state Q° and a set of marked states $Q^m \subseteq Q$. We write $\delta(q, \sigma)!$ if $\delta(q, \sigma) \neq \emptyset$. The set of transitions in G is

$$\mapsto := \{(q, \sigma, q') \in Q \times \Sigma \times Q \mid \delta(q, \sigma)! \text{ and } q' \in \delta(q, \sigma)\}.$$

We sometimes write $q \overset{\sigma}{\mapsto} q'$ instead of $(q, \sigma, q') \in \mapsto$. Let Σ^* be the set of all finite strings over Σ, including the empty string ε. We write $st \in \Sigma^*$ for the concatenation of two strings $s, t \in \Sigma^*$ and $s \leq t$ when s is a *prefix* of t. Further, the notation δ is extended to strings in Σ^* in usual way [see Cassandras and Lafortune, 2008].

For a subset of events $\Sigma_0 \subseteq \Sigma$, a function $P : \Sigma^* \to \Sigma_0^*$ is called *natural projection* according to $P(\varepsilon) = \varepsilon$, and $P(s\sigma)$ is either $P(s)\sigma$ if $\sigma \in \Sigma_0$, or $P(s)$ otherwise. The effect of P on a string $s \in \Sigma^*$ is just to erase the events in s that do not belong to Σ_0, but keep the events in Σ_0 unchanged. The *inverse image* of P is a function $P^{-1} : Pwr(\Sigma_0^*) \to Pwr(\Sigma^*)$, where Pwr is the power set.

Consider a set of variables V. In order to describe the data flow on transition system of EFAs, we add a second set of variables V', where each variable x in V has a corresponding (next-state) variable x' in V' over the same domain. Let $\mathcal{G}_V$ denote the set of formulas over V called *guard formulas* (or just guards), and $\mathcal{A}_V$ denote the set of formulas over V' and/or V called *action formulas* (or just actions). Now, *conditions* $c \in \mathcal{C}_V$ are formulas of the form $c \equiv \phi_g \wedge \phi_a$ for $\phi_g \in \mathcal{G}_V$ and $\phi_a \in \mathcal{A}_V$. Further, for a condition $c \in \mathcal{C}_V$, we denote by $\text{vars}(c)$ (resp. $\text{vars}'(c)$) the set of all variables x (resp. x') appearing in c. Note that, if $V = \emptyset$ then it is assumed that $\mathcal{C}_V = \{\top, \bot\}$.

Definition 1. (Extended Finite Automaton). An extended finite automaton is a tuple $E = \langle V, L, \Sigma, T, \ell^\circ, c^\circ, L^m \rangle$, where V is a finite set of variables, L is a finite set of locations, Σ is a nonempty finite set of events (alphabet), $T \subseteq L \times \Sigma \times \mathcal{C}_V \times L$ is the transition relation, where $\mathcal{C}_V$ is the set of conditions over V and/or V', $\ell^\circ \in L$ is the initial location, $c^\circ \in \mathcal{G}_V$ is the initial guard, and $L^m \subseteq L$ is the set of marked (desired) locations.

We denote by $\ell \xrightarrow{\sigma:c} \ell'$ the presence of a transition in E, from location ℓ to location ℓ' with event $\sigma \in \Sigma$ and condition $c \in \mathcal{C}_V$. Further, we let $\ell \xrightarrow{s:\mathtt{C}} \ell'$ denote the existence of a path in E of the form

$$\ell = \ell_0 \xrightarrow{\sigma_1:c_1} \ell_1 \xrightarrow{\sigma_2:c_2} \cdots \xrightarrow{\sigma_n:c_n} \ell_n = \ell' \ (n > 0)$$

such that $s = \sigma_1\sigma_2 \cdots \sigma_n$ and $\mathtt{C} = c_1; c_2; \ldots; c_n$, where ; is the sequential valuation operator for conditions. We write $\text{vars}(\mathtt{C}) := \bigcup_{i=1}^n \text{vars}(c_i)$ and in a similar way for $\text{vars}'(\mathtt{C})$. In what follows, we ignore the distinction between a sequence $\mathtt{C} = c_1; \ldots; c_m$ and ordered (by index) set $\mathtt{C} := \{c_1, \ldots, c_m\}$ thus interchange them without essential loss.

To be able to relate EFAs to language-based approaches, we define a EFA transition function $\mu : L \times \mathfrak{L} \to Pwr(L)$, where $\mathfrak{L} \subseteq \Sigma \times \mathcal{C}_V$ is a set of *transition labels* where we write $\sigma : c$ instead of $(\sigma, c) \in \mathfrak{L}$. We say that $\mu(\ell, \sigma : c)$ is defined, denoted $\mu(\ell, \sigma : c)!$, if there exists $\ell \xrightarrow{\sigma:c} \ell' \in T$ for some $\ell' \in L$, and write $|\mu(\ell, \sigma : c)|$ for the cardinality of the set of locations given by $\mu(\ell, \sigma : c)$.

The transition function μ is extended to strings recursively as follows: $\mu(\ell, \varepsilon : \top) := \ell$, $\mu(\ell, s\sigma : \mathtt{C}; c) := \mu(\mu(\ell, s : \mathtt{C}), \sigma : c)$ for $s = \sigma_1 \cdots \sigma_m \in \Sigma^*$ and $\mathtt{C} = c_1; \ldots; c_m$; and $\mu(\ell, s\sigma :$

$\mathtt{C}; c)!$ iff $\mu(\ell', \sigma:c)!$ for $\ell' = \mu(\ell, s:\mathtt{C})$. Sometimes, when no confusion is possible, we skip writing the conditions $\mathtt{C}$ and write $\mu(\ell, s)$ instead of $\mu(\ell, s:\mathtt{C})$.

EFA Semantics An instantaneous snapshot of data flow at any moment in *executing* EFAs is determined by the values of variables. Thus our *locations* contains valuation of variables over the domain [1] .

Let η and η' be two valuations of the variables V and V', respectively, over the domain D. Then, we associate the pair (η, η') with a condition c if $(\eta, \eta') \vDash c$. We call η and η' the present-state and the next-state valuation.

For a condition c and subset of variables $W \subseteq V$, let $c_{\wedge,W} \equiv c \wedge \bigwedge_{y \in W - \mathrm{vars}'(c)} y' = y$ denote a new condition in which variables in W whose values are not updated by c keep their current values. The semantics of the sequential valuation operator ; for two conditions c_1, c_2 is defined by

$$\textbf{COMP}\frac{(\eta, \eta'') \vDash c_{1,\wedge,V}, \quad (\eta'', \eta') \vDash c_{2,\wedge,V}}{(\eta, \eta') \vDash c_1; c_2}$$

For example, let x be a variable over domain $\{0, \ldots, 5\}$ and assume $c \equiv x > 2 \wedge x' = x + 1$. Given current-state valuation $\eta[x/3]$, we have that $(\eta[x/3], \eta'[x'/4]) \vDash c$, results in the next-state valuation $\eta'[x'/4]$ whenever the transition is fired. But, if $\eta[x/0]$ or $\eta[x/5]$, then $(\eta, \eta') \nvDash c_1$ for any η' and the transition is disabled.

The semantics of an EFA is given by means of an FA.

Definition 2. (EFA Semantics).
Let $E = \langle V, L, \Sigma, T, \ell^\circ, c^\circ, L^m \rangle$ be an EFA. The finite-state automaton $G(E)$ of E is the tuple $\langle Q_E, \Sigma_E, \mapsto_E, Q_E^\circ, Q_E^m \rangle$ where $Q_E = L \times D$, $\Sigma_E = \Sigma$, $Q_E^\circ = \{\langle \ell^\circ, \eta^\circ \rangle | \eta^\circ \vDash c^\circ \text{ for valuation } \eta^\circ\}$, $Q_E^m = L^m \times D$, and the explicit transition relation $\mapsto_E \subseteq Q_E \times \Sigma_E \times Q_E$ is defined by

$$\textbf{SEM}\frac{\ell \xrightarrow{\sigma:c} \ell', \quad (\eta, \eta') \vDash c_{\wedge,V}}{\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \ell', \eta' \rangle}.$$

Intuitively, the states of $G(E)$ are the reachable states of E, and each state consists of a location ℓ together with a valuation η. The transitions of $G(E)$ are defined by the above inference rule, stating that whenever there exists a transition $\ell \xrightarrow{\sigma:c} \ell'$ in E and two valuations η and η' such that $(\eta, \eta') \vDash c_{\wedge,V}$, there also exists a transition $\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \ell', \eta' \rangle$ in $G(E)$. Note that, in the rule **SEM**, the abbreviation $\hat{c}$ is used to be a new condition obtained from c such that those variables in V whose values are not changed in c keep their current value.

EFA Behavior and Properties The behavior of E is given by the language generated by its underlying explicit transition system $G(E)$. The *language* of E is defined as

$$\mathcal{L}(E) := \{u \in \Sigma^* | (\exists p \in Q_{G(E)})\ q^\circ \overset{u}{\mapsto}_{G(E)} p\}.$$

In addition, for E, we define the *implicit language*

$$\mathfrak{S}(E) := \{s \in \Sigma^* \mid \mu(\ell^\circ, s:\mathtt{C})!\}$$

in accordance with the structure of E without considering the data flow.

[1] Note that, in this paper we fix the interpretation over the theory of equality and integers with standard interpretation of arithmetic symbols. However, any other interpretations and theories can be used as long as a proper semantics is provided.

Definition 3. (EFA properties). An EFA E is called (i) *structurally deterministic* (or just Σ-*deterministic*) if for any $\ell \in L$ and $\sigma \in \Sigma$ the following cases hold: (a) $|\mu(\ell, \sigma)| \leq 1$, (b) for any ℓ_1, ℓ_2 where $\ell \xrightarrow{\sigma:c_i} \ell_i$ $(i = 1, 2)$ always implies that $\mathcal{G}_V(c_1) \wedge \mathcal{G}_V(c_2) \equiv \bot$, and (c) for any two transitions $\ell \xrightarrow{\sigma:c_i} \ell'$ $(i = 1, 2)$, it follows that $\mathcal{A}_V(c_1) \equiv \mathcal{A}_V(c_2)$; (ii) *structurally nonblocking* (or just Σ-*nonblocking*) if $(\forall \ell \in L; \exists s \in \mathfrak{S}(E))\ell = \mu(\ell^\circ, s) \Rightarrow (\exists t \in \Sigma^*)\mu(\ell, t) \in L^m$; and (iii) (explicitly) *deterministic* and *nonblocking* iff $G(E)$ is deterministic and nonblocking.

Note that, in Def. 3-(i), if E is not Σ-deterministic, it does not imply that $G(E)$ is also nondeterministic. For large systems, however, the verification of (explicit) determinism and nonblocking properties for E is computationally expensive, since it demands the actual value of the variables to be known. In the sequel, we model DES by Σ-deterministic EFAs and whenever we say nonblocking we mean Σ-nonblocking, unless otherwise stated. Hence, we consider $\mu : L \times \mathfrak{L} \to L$ as the EFA transition function.

A realistic DES is often composed from a group of EFA components. Consider a discrete-event system

$$\textbf{DES} := \{E_1, \ldots, E_n\} \tag{1}$$

consisting of n deterministic and nonblocking EFA components over the respective alphabet $\Sigma_1, \ldots, \Sigma_n$ and variables $V_1, \ldots, V_n$, for which we want to synthesize a controller. Each component, E_i, can share events as well as variables with other components. These sets of *shared events* and *shared variables* are defined as $\Sigma_{i,\cap} := \bigcup_{j=1, j\neq i}^n (\Sigma_j \cap \Sigma_i)$ and $V_{i,\cap} := \bigcup_{j=1, j\neq i}^n (V_j \cap V_i)$, respectively. Also, some variables values might only be changed by one component in the system. The set of such variables in E_i is defined as $V_i^a := \{x \in V_i | x \in (\mathrm{vars}'(\mathcal{C}_{V_i}) - \bigcup_{j=1, j\neq i}^n \mathrm{vars}'(\mathcal{C}_{V_j}))\}$.

EFAs, similar to ordinary finite automata, are composed by extended full synchronous composition (EFSC).

Definition 4. (EFSC).
Let $E_k = \langle V_k, L_k, \Sigma_k, T_k, \ell_k^\circ, c_k^\circ, L_k^m \rangle$, $k = 1, 2$, be two EFAs. The *Extended Full Synchronous Composition* of E_1 and E_2 is the tuple $E_1 \| E_2 = \langle V, L, \Sigma, T, \ell^\circ, c^\circ, L^m \rangle$, where $V = V_1 \cup V_2$, $L = L_1 \times L_2$, $\Sigma = \Sigma_1 \cup \Sigma_2$, $\ell^\circ = \langle \ell_1^\circ, \ell_2^\circ \rangle$, $c^\circ = c_1^\circ \wedge c_2^\circ$, $L^m = L_1^m \times L_2^m$, and T is defined by the following rules:

$$\textbf{SYN1}\frac{\ell_1 \xrightarrow{\sigma:c_1} \ell_1', \quad \sigma \in (\Sigma_1 - \Sigma_2)}{\langle \ell_1, \ell_2 \rangle \xrightarrow{\sigma:c_1} \langle \ell_1', \ell_2 \rangle}$$

$$\textbf{SYN2}\frac{\ell_2 \xrightarrow{\sigma:c_2} \ell_2', \quad \sigma \in (\Sigma_2 - \Sigma_1)}{\langle \ell_1, \ell_2 \rangle \xrightarrow{\sigma:c_2} \langle \ell_1, \ell_2' \rangle}$$

$$\textbf{SYN3}\frac{\ell_1 \xrightarrow{\sigma:c_1} \ell_1', \quad \ell_2 \xrightarrow{\sigma:c_2} \ell_2', \quad \sigma \in (\Sigma_1 \cap \Sigma_2)}{\langle \ell_1, \ell_2 \rangle \xrightarrow{\sigma:c_1 \wedge c_2} \langle \ell_1', \ell_2' \rangle}$$

Note that, in the rule **SYN3**, if the condition $c_1 \wedge c_2$ tries to simultaneously change the value of a variable to different values (nondeterministic assignment), the underlying transition in $G(E_1 || E_2)$ is not defined, see Def. 2.

Quotient EFA Given an EFA $E = \langle V, L, \Sigma, T, \ell^\circ, c^\circ, L^m \rangle$, a subset of events $\Sigma_0 \subseteq \Sigma$, and an *equivalence relation* $\pi \in \mathcal{E}(L)$ with the associated *canonical projection* $p_\pi : L \to L/\pi$, a (nondeterministic) *quotient* of E mod π is an EFA

$$E_{\pi,\Sigma_0} = \langle V, L/\pi, \Sigma_0 \cup \{\varepsilon\}, \tilde{T}, [\ell^\circ]_\pi, c^\circ, L^m/\pi \rangle, \quad (2)$$

where $[\ell]_\pi := \{\ell' \in L | \ell' \equiv_\pi \ell\}$ for any $\ell \in L$ is the *equivalence class* of ℓ modulo π. The possibly nondeterministic *induced transition relation* $\tilde{T}$ is defined by

$$\textbf{QUO}\ \frac{\ell_1 \xrightarrow{\sigma:c} \ell_2 \in T, \quad \tilde{\ell}_1 := p_\pi(\ell_1), \quad \tilde{\ell}_2 := p_\pi(\ell_2)}{\begin{array}{ll} \tilde{\ell}_1 \xrightarrow{\sigma:c} \tilde{\ell}_2 \in \tilde{T} & \text{if } \sigma \in \Sigma_0 \\ \tilde{\ell}_1 \xrightarrow{\varepsilon:c} \tilde{\ell}_2 \in \tilde{T} & \text{if } \sigma \in \Sigma - \Sigma_0 \text{ and } \tilde{\ell}_1 \neq \tilde{\ell}_2 \end{array}}$$

In the above rule, for any transition in T, if neither of $\sigma \in \Sigma_0$ or $\sigma \in \Sigma - \Sigma_0$ and $\tilde{\ell}_1 \neq \tilde{\ell}_2$ hold then no transition is defined in $\tilde{T}$. Throughout the paper, we use quotient E_{π,Σ_0} of E as the abstraction of E w.r.t. π and Σ_0.

3. NATURAL OBSERVERS

In this section, we introduce the notion of *natural observers* and its conditions w.r.t. event and data for discrete-event systems with data. Later, we provide an algorithmic approach to verify these observer conditions on the abstracted models.

In the sequel, let **DES** be the system in (1). Further, for any E_i $(1 \leq i \leq n)$ in **DES** with Σ_i and V_i, we denote by $\ell \overset{\sigma:c}{\Longrightarrow} \ell'$ the existence of a transition $\ell \xrightarrow{\sigma:c} \ell' \in T_i$ s.t. $\sigma \in \Sigma_i - \Sigma_{i,\cap}$, $\text{vars}(c) \subseteq V_i^a$, and $\text{vars}'(c) = \emptyset$. In a straightforward way, we extend this notation to paths: $\ell_0 \overset{s:\mathtt{C}}{\Longrightarrow} \ell_m$ for $s = \sigma_1 \cdots \sigma_m$ and $\mathtt{C} = c_1; \ldots; c_m$ s.t. $\ell_j \overset{\sigma_{j+1}:c_{j+1}}{\Longrightarrow} \ell_{j+1}$ for all $0 \leq j \leq m-1$. Again, we skip writing the conditions on paths $\Rightarrow$ explicitly, when no confusion is possible.

Observer Conditions Contrary to the ordinary automata, transitions of EFAs are augmented with data. Therefore, the observer condition on natural projections w.r.t. the events of the system needs to be extended to data as well. Before that, we restate the observer condition in Wong and Wonham [1996] for EFAs.

Definition 5. (Event-Observer). Let E be an EFA with alphabet Σ and the corresponding implicit-language $\mathfrak{S}(E)$. Let $P : \Sigma^* \to \Sigma_0^*$ be the natural projection for $\Sigma_0 \subseteq \Sigma$. P is an *event-observer* for E iff for all $s \in \mathfrak{S}(E)$ and $t \in \Sigma_0^*$, it holds that

$$P(s)t \in P(\mathfrak{S}(E)) \Rightarrow$$
$$\exists u \in \Sigma^* \text{ s.t. } su \in \mathfrak{S}(E) \text{ and } P(su) = P(s)t.$$

That is, P is event-observer for E if whenever E can reach a marked location via paths with silent events, the reduced EFA must also be able to follow E and reach to (an equivalence class of) a marked location with some "high-level" paths.

In order to introduce the notion of observability w.r.t. data for EFAs, for any $E_i \in$ **DES**, first we let E_i^ω be a new *marked EFA* constructed from E_i such that

$$E_i^\omega = \langle V_i, L_i, \Sigma_i^\omega, T_i^\omega, \ell_i^\circ, c_i^\circ, L_i^m \rangle, \quad (3)$$

where $\Sigma_i^\omega := \Sigma_i \cup \{m\}$ s.t. $m \notin \Sigma_i$ is a new event label called *marking label*, T_i^ω is defined as $T_i^\omega := T_i \cup \{\ell \xrightarrow{m:\top} \ell : \forall \ell \in L_i^m\}$, i.e. bring in a new self-looped transition $\ell \xrightarrow{\omega:\top} \ell$ to "flag" each marked location $\ell \in L_i^m$, and leave the other elements intact. It is assumed that always $\omega \in \Sigma_{i,\cap}^\omega$. For brevity, in what follows we drop superscript ω from E_i^ω. Note that, this marking label treatment is only used for the definition of data-observer and later for the natural observer verification. We shall use the original EFA E_i anywhere else.

Definition 6. (Data-Observer). Let E_i be an EFA with alphabet Σ_i and set of variables V_i. The natural projection $P_i : \Sigma_i^* \to \Sigma_{i,0}^*$ for $\Sigma_{i,\cap} \subseteq \Sigma_{i,0} \subseteq \Sigma_i$ is *data-observer* for E_i if for all $u \in (\Sigma_i - \Sigma_{i,0})^*$, $\sigma \in \Sigma_{i,0}$, and $\ell, \ell' \in L$ s.t. $\ell' = \mu(\ell, u\sigma : \mathtt{H}; c)$, it holds that (i) $\ell \overset{u:\mathtt{H}}{\Longrightarrow} \ell'$ and (ii) $(\eta, \eta') \vDash c \Rightarrow (\eta, \eta') \vDash \mathtt{H}; c$ for any valuations η, η'.

In words, a natural projection P_i is data-observer for E_i, if for every path in E_i labeled by $u\sigma$ s.t. $P_i(u\sigma) = \sigma$ and for any $h \in \mathtt{H}$, it holds that (i) h has no action and only guards using variables that are changed by E_i and (ii) if the condition c on the immediate observable transition labeled by σ is true under some valuations, so is the sequence $\mathtt{H}; c$. Hence, if an observable event σ is possible after a "silent" string u in E_i, then σ is also possible in the underlying (explicit) transitions of E_i. Note that, if $\sigma = \omega$ then condition (ii) requires that $h \equiv \top$ for all $h \in \mathtt{H}$ since $c_\omega \equiv \top$ for the marked self-looped transition.

Definition 7. (Natural Observer). The natural projection P_i is called a *natural observer* for E_i (or just E_i-observer) if P_i is event- and data-observer for E_i.

Let us denote by $\widetilde{\mathbf{DES}}$ the abstraction of **DES** by P_i for $i \in \{1, \ldots, n\}$. It is stated in Shoaei et al. [2012b]-Theorem 1 that if P_i is natural observer E_i then **DES** is nonblocking iff $\widetilde{\mathbf{DES}}$ is nonblocking. Thus, synthesis of subsystems are achieved through their model abstractions rather than their global model.

Observer Verification Given the EFA $E_i = \langle V_i, L_i, \Sigma_i, T_i, \ell_i^\circ, c_i^\circ, L_i^m \rangle$ in **DES** with the subset of events $\Sigma_{i,\cap} \subseteq \Sigma_{i,0} \subseteq \Sigma_i$, we are interested to algorithmically verify if the natural projection $P_i : \Sigma_i^* \to \Sigma_{i,0}^*$ is an observer for E_i. Let E_i^ω be the marked EFA of E_i given by (3). Again, for brevity, we drop index i and superscript m from E_i^ω.

To address the conditions for the event-observer in Def. 5 we define two functions Δ_σ^Σ and Δ_ω^Σ such that

$$\Delta_\sigma^\Sigma : L \to Pwr(L) : \ell \mapsto \{\mu(\ell, u\sigma u') | u\acute{u} \in (\Sigma - \Sigma_0)^*\},$$
$$\Delta_\omega^\Sigma : L \to Pwr(L^m) : \ell \mapsto \{\mu(\ell, u\omega) | u \in (\Sigma - \Sigma_0)^*\}.$$

Further, we introduce the function $\Delta_\sigma^V : L \to Pwr(L)$ to check the data-observer condition as in Def. 6 such that

$$\Delta_\sigma^V(\ell) = \begin{cases} \Delta_\sigma^\Sigma(\ell) & \text{if } \forall u \in (\Sigma - \Sigma_0)^* \text{ s.t. } \mu(\ell, u\sigma : \mathtt{H}; c)! \\ & \text{it holds that } \ell \overset{u:\mathtt{H}}{\Longrightarrow} \ell'' \text{ for } \ell'' \in L \text{ and} \\ & [\forall \eta, \eta'] \, (\eta, \eta') \vDash c \Rightarrow (\eta, \eta') \vDash \mathtt{H}; c \\ \emptyset & \text{otherwise.} \end{cases}$$

Now, let $\mathfrak{D}_= (L, \{\Delta_\sigma^\Sigma \mid \sigma \in \Sigma_0\} \cup \Delta_\omega^\Sigma \cup \{\Delta_\sigma^V \mid \sigma \in \Sigma_0\})$ be a dynamic system. According to Wong and Wonham [2004], the coarsest *quasi-congruence* $\pi_{\Sigma_0} \in \mathcal{E}(L)$ for $\mathfrak{D}$,

$$\pi_{\Sigma_0}^* = \sup\{\pi_{\Sigma_0} \in \mathcal{E}(L) | \pi \leq \bigwedge_{\sigma \in \Sigma_0} (\pi \circ (\Delta_\sigma^\Sigma \cap \Delta_\sigma^V)) \wedge \pi \circ \Delta_\omega^\Sigma\} \quad (4)$$

exists and can be computed with the algorithm in Fernandez [1990] with a complexity of $\mathrm{O}\big(|L|^3.|T|\big)$.

For E, Σ_0, and the quasi-congruence $\pi_{\Sigma_0}^*$, we refer to a quotient $E_{\pi_{\Sigma_0}^*, \Sigma_0}$ as the *reduction* of E, which we denote by $P(E, \Sigma_0)$. The natural observer condition can now be verified by means of the reduced EFA $P(E, \Sigma_0)$.

Theorem 1. The natural projection P is an observer for E iff $P(E, \Sigma_0)$ is deterministic.

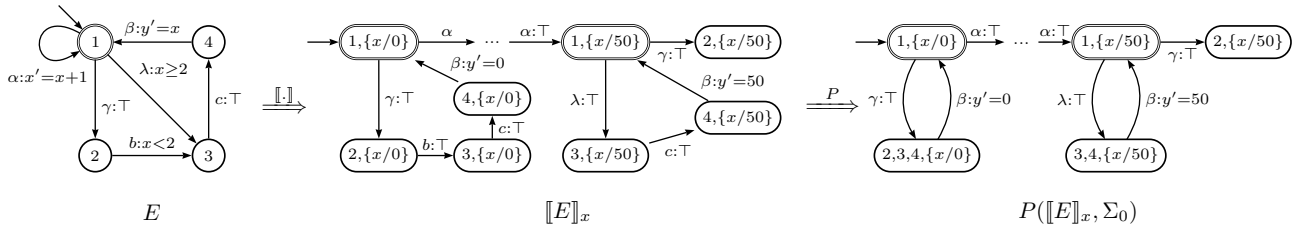

Fig. 1. EFA E, its interpreted EFA $[\![E]\!]_x$ for $\eta^\circ = [x/0]$, and the reduced EFA $P([\![E]\!]_x, \Sigma_0)$ with $\Sigma_0 = \{\alpha, \beta, \gamma, \lambda\}$.

4. COMPUTATION OF NATURAL OBSERVERS

In the previous section, Theorem 1 provides sufficient condition for the verification of observer property for any natural projections. In this section, we address the case where the verification fails due to event- or data-observer conditions as in Def. 5 and Def. 6. For the latter case, we employ a symbolic interpretation technique in Shoaei and Lennartson [2014] and further introduce three guard propagation rules for EFAs, where these techniques can be (repeatedly) applied on each component to logically weaken and/or propagate the conditions on the successor transition so that the projections become data-observer.

For the event-observer condition, we use the existing event set extension algorithm in Feng and Wonham [2008] over the stage-language of EFAs. In that, the algorithm iteratively adds events to the set Σ_0 to remove any ε-transitions and/or resolve the possible nondeterminism in the abstracted model $P(E, \Sigma_0)$. Finally, we formulate the computation of appropriate natural projections for the nonblocking control of EFAs by an observer extension algorithm. This algorithm calls the simplification techniques for EFAs together with the event set extension algorithm in Feng and Wonham [2008] until an appropriate alphabet extension is found.

4.1 Simplification Techniques for EFAs

Symbolic Interpretation In practice, many systems use "internal" variables. Hence, it is of great interest if we could symbolically interpret systems modeled by EFAs w.r.t. their internal variables, in a way that more events become available for the projection. To this end, we use a *symbolic interpretation* technique for EFAs in Shoaei and Lennartson [2014] in order to "interpret and execute" EFAs w.r.t. their internal variables.

The interpretation process can be described as follows: For an EFA E with a set of variables V and a subset of (internal) variables $V_{\text{int}} \subseteq V$, the interpreter $[\![.]\!]$ starts from the initial location of E with initial substitution of variables in V_{int}; iterates over the transitions of E, and by passing each transition, it symbolically interprets and partially evaluates (executes) the condition on that transition w.r.t. the known values of variables V_{int} from the previous step, and leaves the other variables intact. Further, it stores the obtained values (ground terms) as substitutions on the locations; and when it terminates, i.e., reaching a fix point that no more condition is left on the transitions or the evaluation results in the same condition, it returns the interpreted parts of E in form of a *residual* EFA, denote by $[\![E]\!]_{V_{\text{int}}}$, with the set of variables $V - V_{\text{int}}$.

For example, consider EFA E in Fig. 1 over two integer variables x, y with the domain $\{0, \ldots, 50\}$ and $c^\circ \equiv x = 0 \wedge y = 0$. Assume that $V_{\text{int}} := \{x\}$ and let $P : \Sigma^* \to \Sigma_0^*$ be a natural projection with $\Sigma_0 = \{\alpha, \beta, \gamma, \lambda\}$. Clearly, P is not a data-observer for E because the guard on β-transition is true for any value of x but it is not true for the condition $x < 2$ on b-transition when, e.g., x is 2. Thus, the condition Def. 6-(i) is violated by the path $2 \xrightarrow{b:x<2} 3 \xRightarrow{c:\top} 4 \xrightarrow{\beta:y'=x} 1$.

Therefore, we compute $[\![E]\!]_{V_{\text{int}}}$ for $V_{\text{int}} = \{x\}$, see the residual EFA $[\![E]\!]_x$ in Fig. 1. Now, P becomes an observer for E. Thus, we can use P as the model abstraction for $[\![E]\!]_x$, see $P(E^\star, \Sigma_0)$ in Fig. 1. Observe that $P(E^\star, \Sigma_0)$ is blocking for $x \geq 2$.

In this paper, we use the interpreted EFAs in the intermediate step of the natural observer extension algorithm.

Guard Propagation The interpreter $[\![.]\!]$ symbolically interprets and executes EFA w.r.t. their internal variables. This is, however, sometimes require, in the worse case, expands their domains. To avoid this, we introduce three rules on the transitions of EFAs which might help us to verify the data-observer conditions without applying the interpretation.

Consider an EFA E with a location ℓ' s.t. $\ell' \neq \ell^\circ$. Further, for a set X and $\alpha \in X$, let $\alpha \overset{X}{\rightsquigarrow} \beta$ denote $X := (X - \{\alpha\}) \cup \{\beta\}$. We now define three rules based on the incoming and outgoing transitions of ℓ' with the following intuitions:

GP1: ℓ' has only one incoming transition with the property $\ell \xRightarrow{\tau:h} \ell'$ and $\ell \neq \ell'$, and some outgoing transitions. Then the guard on its incoming transition can be propagated to (conjuncted with) all conditions of its outgoing transitions.

GP2: ℓ' has m $(m > 1)$ incoming transitions with the property $\ell_j \xRightarrow{\tau_j:h_j} \ell'$ and $\ell_j \neq \ell'$ for all $j \in \mathbf{m}$, where $\mathbf{m}$ is the index set $\{1, \ldots, m\}$, and all with logically equivalent conditions, i.e., $(\forall j \in \mathbf{m} - \mathbf{1})\ h_j \equiv h_{j+1}$. Then the guard on one of its incoming transition can be propagated to all conditions of its outgoing transitions.

GP3: ℓ' has m $(m > 1)$ incoming transitions where at least one of them has the property $\ell_j \xRightarrow{\tau_j:h_j} \ell'$ for some $j \in \mathbf{m}$, and only one outgoing transition. Furthermore, ℓ' does not belong to any silent cycle in E. Then, the location ℓ' can be split into m equivalent new locations ℓ'_j $(j \in \mathbf{m})$ each of which has one of the incoming transitions of ℓ' and a copy of its outgoing transition. Further, if $\ell' \in L^m$ then $\ell'_j \in L^m$ for all $j \in \mathbf{m}$.

Note that, we use the result of rule **GP3** to construct an intermediate model for which we can apply rule **GP1**. The above rules are illustrated in Fig. 2.

Proposition 1. Let E be an EFA and ℓ' $(\ell' \neq \ell^\circ)$ be a location in E. Consider $\bar{E}$ be the result of applying the guard propagation rules in Fig. 2 to ℓ'. Then E and $\bar{E}$ has the same global behavior.

Corollary 1. In the statement of Prop. 1, consider $\bar{E}^n$ be the result of n times applying the guard propagation rules

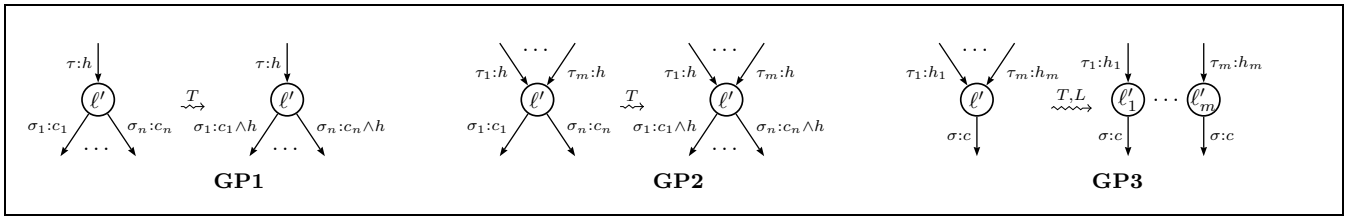

Fig. 2. Guard propagation rules for a location ℓ'. Here, $\overset{X}{\rightsquigarrow}$ denote the modified set after applying the rule.

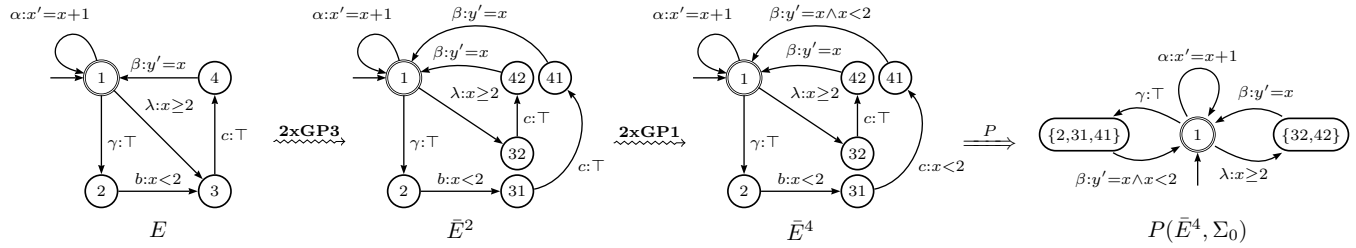

Fig. 3. Applying guard propagation rules on E and then computing the reduced EFA $P(\bar{E}^4, \Sigma_0)$ for $\Sigma_0 = \{\alpha, \beta, \gamma, \lambda\}$ based on the intermediate simplified EFA. Here, the bold formulas denote the propagated guards.

to all locations ℓ' in E s.t. $\ell' \neq \ell^\circ$. Then E and $\bar{E}^n$ has the same global behavior.

For example, consider EFA E and the natural projection P in the previous example. First, we apply rule **GP3** on locations 3 and 4, respectively, resulting in the split locations $31, 32$ and $41, 42$, and rule **GP1** on locations 31 and 41, propagating the guard $x < 2$ on the b-transition to the c- and the following β-transition, see $\bar{E}^2$ and $\bar{E}^4$ in Fig. 3. Now, for the guard propagated EFA $\bar{E}^4$, the natural projection P becomes an observer. Thus, we use P as the model abstraction for $\bar{E}^4$, see $P(\bar{E}^4, \Sigma_0)$ in Fig. 3. Note that, $[\![P(\bar{E}^4, \Sigma_0)]\!]_x$ results in the same EFA as $P(E^\star, \Sigma_0)$ in Fig. 1, though, we obtain the abstraction without expanding the domain of x. However, in general, the interpretation technique often results in better abstraction possibilities than the guard propagation.

4.2 Natural Observer Extension Algorithm

In Feng and Wonham [2008] a polynomial-time observer computation algorithm was presented (w.r.t. event-observer) that returns a reasonable extension of a given projection alphabet such that the natural projection becomes an (event) observer. We generalize that algorithm by Algorithm 1 to consider both data- and event-observer conditions for EFAs, defined in Section 3, and use the simplification techniques, as above.

Algorithm 1 first propagates guards and/or symbolically interprets the given EFA. Then it computes the reduced EFA w.r.t. the quasi-congruence π_{Σ_0}. If the verification of the natural observer condition according to Prop. 1 is successful it returns the current Σ_0 otherwise, an extension of Σ_0 is computed by the event set extension algorithm in Feng and Wonham [2008] where it adds events to Σ_0 in order to eliminate nondeterminism and ε-transitions in the reduced EFA. The observer algorithm iterates until an appropriate alphabet extension is found.

5. APPLICATION

In this section, we apply the proposed approach to a modified version of the cluster tool example in Su et al. [2010]. The cluster tool is an integrated manufacturing system used for wafer processing. It consists of one entering load lock (L_{in}) and one exit load lock (L_{out}), nine chambers (C_{ij}, where, for $i = 1, 2, 3$, we have $j = 1, 2$, and for $i = 4$, we have $j = 1, 2, 3$), three one-slot buffers (B_k for $k = 1, 2, 3$), and four transportation robots (R_i for $i = 1, 2, 3, 4$), see Fig. 5.

Algorithm 1 (Natural Observer Computation for EFAs)

Require: An EFA E and an observable event set Σ_0.

1: Let $\bar{E}$ be the simplified model of E using the rules in Fig. 2 and/or $[\![E]\!]_{V_{\text{int}}}$ w.r.t. internal variables $V_{\text{int}} \subseteq V$
2: **repeat**
3: Compute the quasi-congruence π_{Σ_0} as in Eq. (4)
4: Compute the reduced $\tilde{E} := P(\bar{E}, \Sigma_0)$ as in Eq. (2)
5: **if** $\tilde{E}$ is deterministic **then return** Σ_0
6: Event set extension of Σ_0 as in Feng and Wonham [2008]
7: **until true**

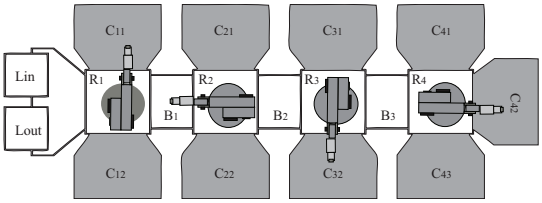

Fig. 5. Structure of Cluster Tool example.

To compute a nonblocking supervisor, we first model the system using EFAs. These models are depicted in Fig. 4. In this, the variables R_i and C_{ij} with domain $\{0, 1\}$ models the robot and chambers status, respectively, where 1 (resp. 0) states that the machine is busy (resp. idle). Further, the variable B_k representing the buffers capacity of one, where 1 (resp. 0) means that corresponding buffer is full (resp. free). In Fig. 4, the desired routing specification are represented by guard formulas on EFAs R_{ij} and specifications B_k.

At first, we let $\Sigma_{i,0} := \Sigma_{i,\cap}$. However, because of the structure of the system, none of the events in $\Sigma_i - \Sigma_{i,0}$ can be abstracted since each of them has action formulas. To end this problem, first we synchronize the robots models $\mathbf{R}_i = \|R_{ij}$ $(i, j = 1, \ldots, 4)$. Then, we apply the partial

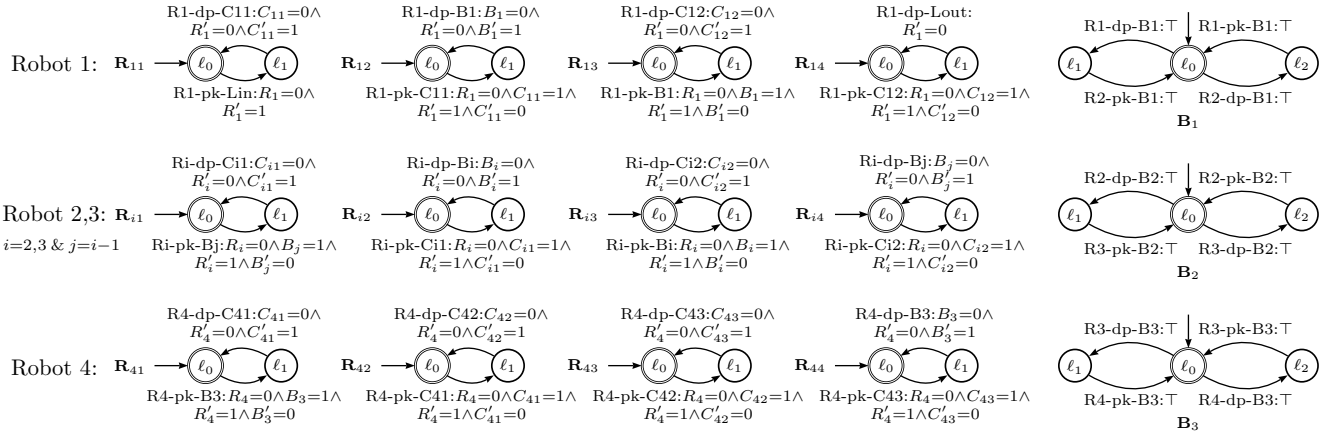

Fig. 4. EFA models of robots ($\mathbf{R}_{ij}$) with the routing specification as guard formulas and buffer specifications $\mathbf{B}_k$.

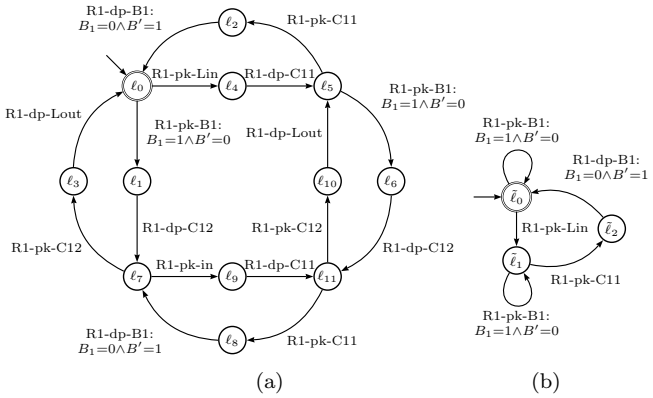

Fig. 6. (a) EFA $\mathbf{R}_1^\star = [\![\mathbf{R}_1]\!]_{\{R_1,C_{1j}\}}$ and (b) the reduced EFA $P_{R_1}(\mathbf{R}_1^\star, \Sigma_0^{R1})$. For brevity, the $\top$ condition are dropped on transitions.

flattening on these synchronized models where we obtain the residual EFAs $\mathbf{R}_i^\star = [\![\mathbf{R}_i]\!]_{\{R_i,C_{ij}\}}$, in each of which the variables R_i and C_{ij} are flattened while the variable B_k remains. The residual EFA $\mathbf{R}_1^\star$ for the synchronized model $\mathbf{R}_1$ is shown in Fig. 6(a), where for the sake of space we drop the variables value in each location. Now, within the residual models, we can see that most of the transitions contains true formula.

In the next step, we use Algorithm 1 to get the extended sets Σ_0^{Ri} such that the natural projection $P_{R_i} : \Sigma_{R_i}^* \to (\Sigma_0^{Ri})^*$ are natural observers for $\mathbf{R}_i^\star$ ($i = 1, \ldots, 4$). For example, using $\mathbf{R}_1^\star$ and $\Sigma_0^{R1} = \{\text{R1-dp-B1}, \text{R1-pk-B1}\}$ as the inputs of the Algorithm 1, we get the extended set $\Sigma_0^{R1} = \{\text{R1-pk-Lin}, \text{R1-pk-C11}, \text{R1-pk-B1}, \text{R1-dp-B1}\}$, where the events R1-pk-Lin and R1-pk-C11 are added to avoid nondeterminism in the reduced model of $P_{R_1}(\mathbf{R}_1^\star, \Sigma_0^{R1})$. With four natural observers P_{R_i}, as above, we compute the reduced models $P_{R_i}(\mathbf{R}_i^\star, \Sigma_0^{Ri})$. For EFA $\mathbf{R}_1^\star$, the reduced model $P_{R_1}(\mathbf{R}_1^\star, \Sigma_0^{R1})$ is depicted in Fig. 6(b), where the locations $\tilde{\ell}_0 = \{\ell_0, \ell_1, \ell_3, \ell_7\}$, $\tilde{\ell}_1 = \{\ell_4, \ell_5, \ell_6, \ell_8, \ell_{10}, \ell_{11}\}$, and $\tilde{\ell}_2 = \{\ell_2, \ell_9\}$ denote the equivalent classes of original locations. Now using these reduced models together with the buffer specifications B_k, we use the DES tool Supremica Akesson et al. [2006] to synthesize a nonblocking supervisor with modest effort. The nonblocking supervisor to achieve a nonblocking control based on the original models has 237 648 states, while the supervisor using the reduced models has 9 682 states.

6. CONCLUSION

We have extended our previous work on model abstraction by natural observers for DES modeled by EFAs and provide sufficient condition for the verification of observer properties over events and data in the system. We further address the case where the verification fails due to observer conditions. To this end, we introduce simplification techniques for EFAs and generalize existing algorithms to compute natural projections for EFAs with observer property. The symbolic simplification techniques, involving both interpretation and propagation, are here turned out useful in order to obtain a significant state reduction in the abstracted models. A manufacturing system demonstrates the practical usage of the proposed approach.

APPENDIX

Proof of Theorem 1

The proof needs the following lemmas. In what follows, for any E, let $\pi \in \mathcal{E}(L)$ be a quasi-congruence on $\mathfrak{D}$ and $p_\pi : L \to L/\pi$ the associated canonical projection.

Lemma 1. Let E be an EFA with Σ and V and domain D. Let $\Sigma_0 \subseteq \Sigma$ and define $P : \Sigma^* \to \Sigma_0^*$. Consider a path $\ell \xrightarrow{s:\mathtt{C}} \ell'$ in E. If P is observer for E then for any η, η', $\langle [\ell]_\pi, \eta\rangle \xmapsto{P(s)} \langle [\ell']_\pi, \eta'\rangle$ exists in $G(P(E, \Sigma_0))$ implies that $\langle \ell, \eta\rangle \overset{s}{\mapsto} \langle \ell', \eta'\rangle$ also exists in $G(E)$.

Proof. Assume $\ell \xrightarrow{s:\mathtt{C}} \ell'$ is of the form $\ell = \ell_0 \xrightarrow{\sigma_1:c_1} \cdots \xrightarrow{\sigma_n:c_n} \ell_n = \ell'$ such that $s = \sigma_1 \ldots \sigma_n$ and $\mathtt{C} = c_1; \ldots; c_n$. Split this path into sub-paths of the form $\ell_m \xrightarrow{u:\mathtt{H}} \ell_{m+i-1} \xrightarrow{\sigma_{m+i}:c_{m+i}} \ell_{m+i}$ for $0 \le m < m+i \le n$ and $u \in (\Sigma - \Sigma_0)^*, \sigma \in \Sigma_0$ s.t. $u\sigma_{m+i} : c_{m+i}$ is a substring of s. Let $[\ell_m]_\pi \xrightarrow{\sigma_{m+i}:c_{m+i}} [\ell_{m+i}]_\pi$ be a path in $P(E, \Sigma_0)$. Since P is event-observer, we have $\ell_m, \ldots, \ell_{m+i-1} \in [\ell_m]_\pi$ and $\ell_{m+i} \in [\ell_{m+i}]_\pi$. Further, because P is also data-observer, we only need to show that if $\langle [\ell_m]_\pi, \eta_m\rangle \xmapsto{\sigma_{m+i}} \langle [\ell_{m+i}]_\pi, \eta_{m+i}\rangle$ for some η_m, η_{m+i} exists in $G(P(E, \Sigma_0))$, there also exists $\langle \ell_m, \eta_m\rangle \xmapsto{u\sigma_{m+i}} \langle \ell_{m+i}, \eta_{m+i}\rangle$ for some $u \in \Sigma^*$ in $G(E)$ s.t. $\sigma_{m+i} = P(u\sigma_{m+i})$. But, data-observer implies that $\ell_m \overset{u:\mathtt{H}}{\Longrightarrow} \ell_{m+i-1}$ and $\forall \eta_m, \eta_{m+i}$ we have $(\eta_m, \eta_{m+i}) \vDash c \Rightarrow (\eta_m, \eta_{m+i}) \vDash \mathtt{H}; c$. Thus, it must be true that $\langle \ell_m, \eta_m\rangle \overset{u}{\mapsto} \langle \ell_{m+i-1}, \eta_m\rangle \xrightarrow{\sigma_{m+i}} \langle \ell_{m+i}, \eta_{m+i}\rangle$ exists in $G(E)$. For the cases that $\sigma_{m+i} \in \Sigma - \Sigma_0$, the sub-path is not projected and explicit path exists in $G(E)$.

Lemma 2. Consider an EFA E and let $P : \Sigma^* \to \Sigma_0^*$ with $\Sigma_0 \subseteq \Sigma$ be an E-observer. Then for all $s_1, s_2, s_1' \in \Sigma^*$, if $s_1 s_1' \in \mathfrak{S}(E)$ and $P(s_1) = P(s_2)$, then it follows that there exists $s_2' \in \Sigma^*$ s.t. $s_2 s_2' \in \mathfrak{S}(E)$ and $P(s_1') = P(s_2')$.

Proof. Because P is an E-observer and nonblocking, we know that there exists two paths $\ell^\circ \xrightarrow{s_i:\mathtt{C}_i} \ell_i \xrightarrow{t_i:\mathtt{T}_i} \ell_i''$ for some $t_i \in \Sigma^*$ $(i = 1, 2)$ s.t. $\ell_i'' \in L^m, P(t_1) = P(t_2)$. Evidently, taking any path of the form $\ell_i \xrightarrow{u:\mathtt{U}} \ell$ and $\ell_i \xrightarrow{u':\mathtt{U}'} \ell'$ for $u, u' \in \Sigma^*$ such that $u, u' \leq t$, we can see that $P(u) = P(u')$. Consider $s_1' = u$ and $s_2' = u'$ and we obtain the result as required.

Back to the proof of Theorem 1.
(**if**) We shall show that if P is an E-observer then $P(E, \Sigma_0)$ is deterministic. Suppose to the contrary that $P(E, \Sigma_0)$ is nondeterministic. Then, there are two cases: (i) $P(E, \Sigma_0)$ contains two transitions $[\ell]_\pi \xrightarrow{\sigma:c} [\ell_i]_\pi$ $(i = 1, 2)$ for $\sigma \in \Sigma_0$ s.t. $[\ell_1]_\pi \neq [\ell_2]_\pi$. Then, by Lemma 1, there also exist $\ell \xrightarrow{u\sigma u':\mathtt{U};c;\mathtt{U}'} \ell_i$ $(i = 1, 2)$ in E for $u, u' \in (\Sigma - \Sigma_0)^*, \ell \in p_\pi^{-1}([\ell]_\pi), \ell_i \in p_\pi^{-1}([\ell_i]_\pi)$. Because $[\ell_1]_\pi \neq [\ell_2]_\pi$ implies that $\ell_1 \neq \ell_2$. Since P is an E-observer there must exist two paths $\ell_i \xrightarrow{t_i:\mathtt{T}_i} \ell_i'$ for $t_i \in \Sigma^*$ and $\ell_i' \in L^m$ s.t. $P(t_1) \neq P(t_2)$. This is in contradictory to the result of Lemma 2 that if P is an observer then $P(t_1) = P(t_2)$. Hence, it must be true that P is not an observer for E, which is a contradiction. (ii) $P(E, \Sigma_0)$ contain a ε-transition. Then there exist $[\ell_i]_\pi \in L/\pi$ $(i = 1, 2)$ and $[\ell_1]_\pi \neq [\ell_2]_\pi$ such that $[\ell_1]_\pi \xrightarrow{\varepsilon} [\ell_2]_\pi$. Recall that π is quasi-congruence. If this is the only outgoing transition from $[\ell_1]_\pi$ then it must be the case that $\ell_1 \equiv_\pi \ell_2$ for all $\ell_2 \in p_\pi^{-1}([\ell_2]_\pi)$, i.e., $[\ell_1]_\pi = [\ell_2]_\pi$, which is a contradiction to the assumption that $[\ell_1]_\pi \neq [\ell_2]_\pi$. Hence, there must exist also a transition $[\ell_1]_\pi \xrightarrow{\sigma:c} [\ell_2']_\pi$ in $P(E, \Sigma_0)$ for $\sigma \in \Sigma_0, [\ell_2']_\pi \in L/\pi$ such that $[\ell_2]_\pi \neq [\ell_2']_\pi$. Now, by a similar reasoning as in (i), we conclude that P is not an E-observer. Combining (i) and (ii), we conclude that $P(E, \Sigma_0)$ must be Σ-deterministic.
(**Only if**) We shall show that if $P(E, \Sigma_0)$ is Σ-deterministic then P is an E-observer. Let $E \xrightarrow{s} \ell_1$ be a path in E. Assume ℓ_1 can continue in E by a path $\ell_1 \xrightarrow{u\sigma u'} \ell_1'$ for $u, u'(\Sigma - \Sigma_0)^*, \sigma \in \Sigma_0$ s.t. $\ell_1' \in L^m$. There is a corresponding path $[\ell_1]_\pi \xrightarrow{\sigma:c} [\ell_1']_\pi$ in $P(E, \Sigma_0)$. Clearly, $[\ell_1']_\pi \in L^m/\pi$. Let $\ell_2' \in L$ be a location s.t. $\ell_2' \in p_\pi^{-1}([\ell_1']_\pi)$. Since $P(E, \Sigma_0)$ is Σ-deterministic and contains no ε-transitions, by Lemma 1, there is a path $\ell_2 \xrightarrow{w\sigma w':\mathtt{W};c;\mathtt{W}'} \ell_2'$ in E for $w, w' \in (\Sigma - \Sigma_0)^*$ s.t. $\ell_2 \in p_\pi^{-1}([\ell_1]_\pi)$. This implies that $P(w\sigma w') = P(u\sigma u')$. By a similar reasoning, we can show this for all paths in E. Hence, P is E-observer.

Proof of Proposition 1

The sketch the proof as follows. The proof follows from the fact that $\ell \neq \ell^\circ$ and the incoming transition(s) is not a self-loop. Furthermore, by the definition of the paths $\Rightarrow$, we have that the event τ (resp. events τ_j) is unique in E and the guard h has no action and only guards over the internal variables of E. Hence, for the rules **GP1** and **GP2**, straightforwardly we can show that if h is true in some valuation, then immediately we have $c \wedge h \equiv c$. Otherwise, if h is false, the transition is disabled and all its outgoing transitions will be unreachable in E. We have the similar behavior when we propagate h to the conditions on the outgoing transitions, namely, $h \equiv \bot \Rightarrow c \wedge h \equiv \bot$. Regarding the rule **GP3**, clearly, the new locations ℓ_j' are bisimilar (cf. Milner [1989]) to ℓ', namely, $(\forall j)$ ℓ_j' has the same future behavior as ℓ' in E.

REFERENCES

Knut Akesson, Martin Fabian, Hugo Flordal, and Robi Malik. Supremica - An integrated environment for verification, synthesis and simulation of discrete event systems. In *8th Int. Work. Discret. Event Syst.*, pages 384–385, Ann Arbor, MI, USA, July 2006.

Christos G Cassandras and Stéphane Lafortune. *Introduction to Discrete Event Systems.* Springer US, Boston, MA, 2nd edition, 2008.

Lei Feng and W. M. Wonham. On the Computation of Natural Observers in Discrete-Event Systems. *Discret. Event Dyn. Syst.*, 20(1):63–102, October 2008.

Jean-Claude Fernandez. An implementation of an efficient algorithm for bisimulation equivalence. *Sci. Comput. Program.*, 13(2-3):219–236, May 1990.

Jean H. Gallier. *Logic For Computer Science.* Addison-Wesley Wokingham, 2003.

Peyman Gohari and W. M. Wonham. On the complexity of supervisory control design in the RW framework. *IEEE Trans. Syst. Man. Cybern.*, 30(5):643–52, January 2000.

Robin Milner. *Communication and concurrency.* Series in Computer Science. Prentice-Hall, 1989.

Sahar Mohajerani, Robi Malik, and Martin Fabian. Compositional nonblocking verification for extended finite-state automata using partial unfolding. In *IEEE Int. Conf. Autom. Sci. Eng.*, pages 930–935, August 2013.

P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proc. IEEE, Spec. Issue Discret. Event Dyn. Syst.*, 77(1):81–98, 1989.

Klaus Schmidt and Christian Breindl. Maximally Permissive Hierarchical Control of Decentralized Discrete Event Systems. *IEEE Trans. Automat. Contr.*, 56(4): 723–737, April 2011.

Mohammad Reza Shoaei and Bengt Lennartson. Symbolic Interpretation and Execution of Extended Finite Automata. In *12th Int. Work. Discret. Event Syst.*, page 7, May 2014.

Mohammad Reza Shoaei, Lei Feng, and Bengt Lennartson. Abstractions for Nonblocking Supervisory Control of Extended Finite Automata. In *IEEE Int. Conf. Autom. Sci. Eng.*, 2012a.

Mohammad Reza Shoaei, Lei Feng, and Bengt Lennartson. Supervisory control of extended finite automata using transition projection. In *51st IEEE Conf. Decis. Control*, pages 7259–7266, December 2012b.

Markus Skoldstam, Knut Akesson, and Martin Fabian. Modeling of discrete event systems using finite automata with variables. In *46th IEEE Conf. Decis. Control*, pages 3387–3392, 2007.

Rong Su, Jan H. van Schuppen, and Jacobus E. Rooda. Aggregative Synthesis of Distributed Supervisors Based on Automaton Abstraction. *IEEE Trans. Automat. Contr.*, 55(7):1627–1640, July 2010.

K. C. Wong and W. M. Wonham. Hierarchical control of discrete-event systems. *Discret. Event Dyn. Syst.*, 6(3): 241–273, July 1996.

K. C. Wong and W. M. Wonham. Modular control and coordination of discrete-event systems. *Discret. Event Dyn. Syst.*, 8(3):247–297, October 1998.

K. C. Wong and W. M. Wonham. On the Computation of Observers in Discrete-Event Systems. *Discret. Event Dyn. Syst.*, 14(1):55–107, January 2004.

W. M. Wonham. *Supervisory Control of Discrete Event Systems.* Department of Electrical and Computer Engineering, University of Toronto, 2002-2012, 2013.