Marked-String Accepting Observers for the Hierarchical and Decentralized Control of Discrete Event Systems

Klaus Schmidt and Thomas Moor
Universität Erlangen-Nürnberg
Lehrstuhl für Regelungstechnik
Cauerstraße 7 - 91058 Erlangen, Germany
{klaus.schmidt,thomas.moor}@rt.eei.uni-erlangen.de

Abstract

The paper[1] extends previous work, where we develop a control theory for the nonblocking hierarchical control of decentralized discrete event systems (DES). These results are based on two technical conditions for the hierarchical abstraction: it has to be (i) *marked string accepting* and (ii) *locally nonblocking*.

In this paper, we investigate the systematic construction of the hierarchical abstraction. Starting from an initial natural projection which need not fulfill (i) and (ii), we provide an algorithm to compute the hierarchical abstraction with the coarsest equivalence kernel finer than that of the initial natural projection, and such that (i) and (ii) hold. Our approach extends the work in [11], where the authors compute observers for the hierarchical control of DES.

I. Introduction

Recent approaches for the control of large-scale discrete event systems employ hierarchical control architectures for reducing the computational complexity of supervisor synthesis [2], [4], [5], [7], [8], [10], [12]. In hierarchical architectures, controller synthesis is based on a plant abstraction (high-level model), which is supposed to be less complex than the original plant model (low-level model). The main question is how to derive the plant abstraction and the low-level supervisor implementation of a

[1]This technical report is an extended version of [9].

high-level controller such that the closed-loop system in the low level is nonblocking and satisfies the expected behavior in the high level.

All of the above approaches assume that the high-level observation is given. [2] use a two-level control hierarchy such that hierarchical consistent and nonblocking control are guaranteed by construction. In [4], [5], [7], [10], it is required that certain sufficient conditions for nonblocking and hierarchically consistent control hold. However, little is known about how to choose the high-level observations systematically such that these conditions are fulfilled.

A first result in this direction is elaborated in [11] as an extension to the theory of observers in [12]. An observer with the coarsest possible equivalence kernel that is finer than that of an initial *causal reporter map* is computed. Nevertheless, the choice of the initial reporter map is not obvious.

In this paper, we consider the hierarchical and decentralized architecture presented in [8], where the overall system is modeled by the synchronous product of decentralized subsystems. A natural projection, where the *shared events* of the decentralized subsystems must be contained in the high-level alphabet, is used for hierarchical abstraction. For nonblocking and hierarchically consistent control, it is required that this natural projection is (i) *locally nonblocking* and (ii) *marked string accepting*. The problem to be solved is to find a natural projection such that the shared events are contained in the high-level alphabet and (i) and (ii) are fulfilled. Similar to the observer algorithm in [11], we develop a procedure to determine such a natural projection with the coarsest possible equivalence kernel starting from the natural projection on the shared events.

The outline of the paper is as follows. Basic notations and definitions of supervisory control theory are recalled in Section II. Section III discusses the features of the hierarchical and decentralized control approach in [8] and formalizes the problem statement. Our algorithm is developed and illustrated with an example in Section IV. Section V elaborates how the algorithm can be applied to build an architecture for nonblocking hierarchical and decentralized supervisory control.

II. PRELIMINARIES

We recall basics from supervisory control theory [1], [13].

For a finite alphabet Σ, the set of all finite strings over Σ is denoted Σ^*. We write $s_1 s_2 \in \Sigma^*$ for the concatenation of two strings $s_1, s_2 \in \Sigma^*$. We write $s_1 \leq s$ when s_1 is a *prefix* of s, i.e. if there exists a string $s_2 \in \Sigma^*$ with $s = s_1 s_2$. The empty string is denoted $\varepsilon \in \Sigma^*$, i.e. $s\varepsilon = \varepsilon s = s$ for all $s \in \Sigma^*$. A *language* over Σ is a subset $M \subseteq \Sigma^*$. The *prefix closure* of M is defined by $\overline{M} := \{s_1 \in \Sigma^* | \exists s \in M \text{ s.t. } s_1 \leq s\}$. A language M is *prefix closed* if $M = \overline{M}$.

The *natural projection* $p_i : \Sigma^* \to \Sigma_i^*$, $i = 1, 2$, for the (not necessarily disjoint) union $\Sigma = \Sigma_1 \cup \Sigma_2$ is defined iteratively: (1) let $p_i(\varepsilon) := \varepsilon$; (2) for $s \in \Sigma^*$, $\sigma \in \Sigma$, let $p_i(s\sigma) := p_i(s)\sigma$ if $\sigma \in \Sigma_i$, or $p_i(s\sigma) := p_i(s)$ otherwise. The set-valued inverse of p_i is denoted $p_i^{-1} : \Sigma_i^* \to 2^{\Sigma^*}$. The *synchronous product* $M_1||M_2 \subseteq \Sigma^*$ of two languages $M_i \subseteq \Sigma_i^*$ is $M_1||M_2 = p_1^{-1}(M_1) \cap p_2^{-1}(M_2) \subseteq \Sigma^*$.

A *finite automaton* is a tuple $G = (X, \Sigma, \delta, x_0, X_\mathrm{m})$, with the finite set of *states* X; the finite alphabet of *events* Σ; the partial *transition function* $\delta : X \times \Sigma \to X$; the *initial state* $x_0 \in X$; and the set of *marked states* $X_\mathrm{m} \subseteq X$. We write $\delta(x, \sigma)!$ if δ is defined at (x, σ). In order to extend δ to a partial function on $X \times \Sigma^*$, recursively let $\delta(x, \varepsilon) := x$ and $\delta(x, s\sigma) := \delta(\delta(x, s), \sigma)$, whenever both $x' = \delta(x, s)$ and $\delta(x', \sigma)!$. $L(G) := \{s \in \Sigma^* : \delta(x_0, s)!\}$ and $L_\mathrm{m}(G) := \{s \in L(G) : \delta(x_0, s) \in X_\mathrm{m}\}$ are the *closed* and *marked language* generated by the finite automaton G, respectively. For any string $s \in L(G)$, $\Sigma(s) := \{\sigma | s\sigma \in L(G)\}$ is the set of eligible events after s. A formal definition of the synchronous composition of two automata G_1 and G_2 can be taken from e.g. [1]. Note that $L_\mathrm{m}(G_1||G_2) = L_\mathrm{m}(G_1)||L_\mathrm{m}(G_2)$.

In a supervisory control context, we write $\Sigma = \Sigma_c \cup \Sigma_u$, $\Sigma_c \cap \Sigma_u = \emptyset$, to distinguish *controllable* (Σ_c) and *uncontrollable* (Σ_uc) events. A *control pattern* is a set γ, $\Sigma_\mathrm{uc} \subseteq \gamma \subseteq \Sigma$, and the set of all control patterns is denoted $\Gamma \subseteq 2^\Sigma$. A *supervisor* is a map $S \colon L(G) \to \Gamma$, where $S(s)$ represents the set of enabled events after the occurrence of string s; i.e. a supervisor can disable controllable events only. The language $L(S/G)$ generated by G under supervision S is iteratively defined by (1) $\varepsilon \in L(S/G)$ and (2) $s\sigma \in L(S/G)$ iff $s \in L(S/G), \sigma \in S(s)$ and $s\sigma \in L(G)$. Thus, $L(S/G)$ represents the behavior of the *closed-loop system*. To take into account the marking of G, let $L_\mathrm{m}(S/G) := L(S/G) \cap L_\mathrm{m}(G)$. The closed-loop system is *nonblocking* if $\overline{L_\mathrm{m}(S/G)} = L(S/G)$, i.e. if each string in $L(S/G)$ is the prefix of a marked string in $L_\mathrm{m}(S/G)$.

A language M is said to be controllable w.r.t. $L(G)$ if there exists a supervisor S such that $\overline{M} = L(S/G)$. The set of all languages that are controllable w.r.t. $L(G)$ is denoted $\mathcal{C}(L(G))$. Furthermore, the set $\mathcal{C}(L(G))$ is closed under arbitrary union. In particular, for every *specification* language E there uniquely exists a *supremal controllable sublanguage* of E w.r.t. $L(G)$, which is formally defined as $\kappa_{L(G)}(E) := \cup\{K \in \mathcal{C}(L(G)) |\ K \subseteq E\}$. A supervisor S that leads to a closed-loop behavior $\kappa_{L(G)}(E)$ is said to be *maximally permissive*.

A language E is $L_\mathrm{m}(G)$-*closed* if $\overline{E} \cap L_\mathrm{m}(G) = E$ and the set of $L_\mathrm{m}(G)$-closed languages is denoted $\mathcal{F}_{L_\mathrm{m}(G)}$. For specifications $E \in \mathcal{F}_{L_\mathrm{m}(G)}$, the plant $L(G)$ is nonblocking under maximally permissive supervision.

III. Hierarchical Control Approach

In [8], a hierarchical approach for the control of decentralized DES as illustrated in Figure 1 is developed.

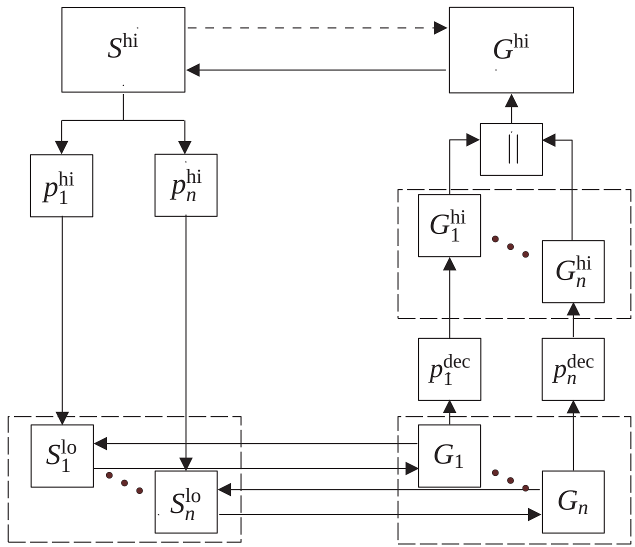

Fig. 1. Hierarchical architecture

Decentralized DES $||_{i=1}^n G_i$ are represented by finite automata G_i, $i = 1,\ldots,n$ with the respective alphabets Σ_i. The overall system with the alphabet $\Sigma := \bigcup_{i=1}^n \Sigma_i$ is defined as $G := ||_{i=1}^n G_i$. High-level abstractions G_i^{hi} of the low-level subsystems G_i are computed by evaluating the natural projections $p_i^{\mathrm{dec}} : \Sigma_i^* \to (\Sigma_i^{\mathrm{hi}})^*$ of the low-level languages $L(G_i)$ and $L_{\mathrm{m}}(G_i)$ such that $L(G_i^{\mathrm{hi}}) = p_i^{\mathrm{dec}}(L(G_i))$ and $L_{\mathrm{m}}(G_i^{\mathrm{hi}}) = p_i^{\mathrm{dec}}(L_{\mathrm{m}}(G_i))$. We require

a. the high-level alphabets Σ_i^{hi} are chosen such that $\bigcup_{j\neq i}^n (\Sigma_i \cap \Sigma_j) \subseteq \Sigma_i^{hi} \subseteq \Sigma_i$, i.e. Σ_i^{hi} contains all events shared with other components.

The overall high-level model G^{hi} is defined such that $L(G^{\mathrm{hi}}) := p^{\mathrm{hi}}(L(G))$ and $L_{\mathrm{m}}(G^{\mathrm{hi}}) = p^{\mathrm{hi}}(L_{\mathrm{m}}(G))$ with the natural projection $p^{\mathrm{hi}} : \Sigma^* \to (\bigcup_{i=1}^n \Sigma_i^{\mathrm{hi}})$. Using assumption a., it can be shown [10] that $G^{\mathrm{hi}} = ||_{i=1}^n G_i^{\mathrm{hi}}$. This means that instead of deriving the high-level model G^{hi} from the overall low-level model G, a parallel composition of the decentralized high-level models G_i^{hi} can be evaluated. The tuple $(||_{i=1}^n G_i, ||_{i=1}^n G_i^{\mathrm{hi}})$ is denoted a *decentralized projected DES*. A nonblocking high-level supervisor S^{hi} for G^{hi} and a high-level specification $E^{\mathrm{hi}} \subseteq L_{\mathrm{m}}(G^{\mathrm{hi}})$ is implemented by decentralized low-level supervisors S_i^{lo}. The decentralized supervisors exist if

b. the high-level languages $L(G_i^{\mathrm{hi}})$ are mutually controllable (see [6]).

The hierarchical and decentralized control architecture guarantees nonblocking and hierarchically consistent control if

c. the decentralized low-level – high-level tuples (G_i, G_i^{hi}) are locally nonblocking and marked string accepting as stated in Definition 3.1 and 3.2.

The approach is computationally efficient as both the abstraction and the supervisor implementation do not require the computation of the overall system and it can be shown that the high-level models always have less states than the low-level models [8].

From the perspective of each individual subsystem G_i, nonblocking control is based on two different types of conditions. Verifying mutual controllability of the high-level languages $L(G_i^{\mathrm{hi}})$ (condition b.) involves the other subsystems. Different from this, the locally nonblocking and the marked string accepting condition (c.) exclusively depend on the behavior of each particular tuple (G_i, G_i^{hi}), denoted *projected system* (PS), and the choice of the high-level alphabet (condition a.).

The latter *structural* conditions, which only depend on the system structure of each projected system, are investigated. For notational convenience, we replace (G_i, G_i^{hi}) by (H, H^{hi}) with the alphabets Σ and Σ^{hi} and the natural projection $p^{\mathrm{hi}} : \Sigma^* \to (\Sigma^{\mathrm{hi}})^*$.

A PS (H, H^{hi}) is locally nonblocking if for all low-level strings $s \in L(H)$ and for all high-level events $\sigma \in \Sigma^{\mathrm{hi}}$, which are feasible after the corresponding high-level string $p^{\mathrm{hi}}(s)$, a local path starting from s exists, such that σ can occur.

Definition 3.1 (Locally Nonblocking Condition): Let (H, H^{hi}) be a PS. The string $s^{\mathrm{hi}} \in L(H^{\mathrm{hi}})$ is locally nonblocking if for all $s \in L(H)$ with $p^{\mathrm{hi}}(s) = s^{\mathrm{hi}}$ and $\forall \sigma \in \Sigma^{\mathrm{hi}}(s^{\mathrm{hi}})$, $\exists u \in (\Sigma - \Sigma^{\mathrm{hi}})^*$ s.t. $su\sigma \in L(H)$. (H, H^{hi}) is locally nonblocking if s^{hi} is locally nonblocking for all $s^{\mathrm{hi}} \in L(H^{\mathrm{hi}})$.

For formulating the marked string accepting condition, the set of *exit strings* is needed. For a given PS (H, H^{hi}) and a high-level string $s^{\mathrm{hi}} \in L(H^{\mathrm{hi}})$, the set of exit strings $L_{\mathrm{ex},s^{\mathrm{hi}}}$ is the set of corresponding low-level strings which have a high-level successor event, i.e. $L_{\mathrm{ex},s^{\mathrm{hi}}} := \{s \in L(H) | p^{\mathrm{hi}}(s) = s^{\mathrm{hi}} \wedge (\exists \sigma \in \Sigma^{\mathrm{hi}} \text{ s.t. } s\sigma \in L(H))\} \subseteq \Sigma^*$.

Marked string acceptance guarantees that if the high-level system passes a marked string, the low-level system also has to pass a marked string.

Definition 3.2 (Marked String Acceptance): Let $(H, H^{\rm hi})$ be a PS. The string $s^{\rm hi} \in L_{\rm m}(H^{\rm hi})$ is marked string accepting[2] if for all $s \in L_{{\rm ex},s^{\rm hi}}$

$$\exists s' \leq s \text{ with } p^{\rm hi}(s') = s^{\rm hi} \text{ and } s' \in L_{\rm m}(H). \tag{1}$$

$(H, H^{\rm hi})$ is marked string accepting if $s^{\rm hi}$ is marked string accepting for all $s^{\rm hi} \in L_{\rm m}(H^{\rm hi})$.

According to condition a., the choice of the high-level alphabets $\Sigma_i^{\rm hi}$ is restricted by $\bigcup_{j \neq i}(\Sigma_i \cap \Sigma_j) \subseteq \Sigma_i^{hi} \subseteq \Sigma_i$. To keep the high-level model $H_i^{\rm hi}$ small, a natural candidate is $\Sigma_i^{\rm hi} = \bigcup_{j \neq i}(\Sigma_i \cap \Sigma_j)$. However, choosing this $\Sigma_i^{\rm hi}$, the locally nonblocking and the marked string accepting condition need not be fulfilled. An intuitive solution to this problem is presented in the following example.

Example 3.1: The PS $(H, H^{\rm hi})$ for H in Figure 2 and $\Sigma^{\rm hi} := \{\alpha, \beta, \gamma, \delta, \varphi, \psi\}$ is marked string accepting but not locally nonblocking. After $s^{\rm hi} = \psi$, the high-level events ξ and β are feasible. Yet, $s = \mathtt{a}\psi\mathtt{h}$ cannot be extended with $u \in (\Sigma - \Sigma^{\rm hi})^*$ such that $su\xi \in L(H)$ which violates Definition 3.1.

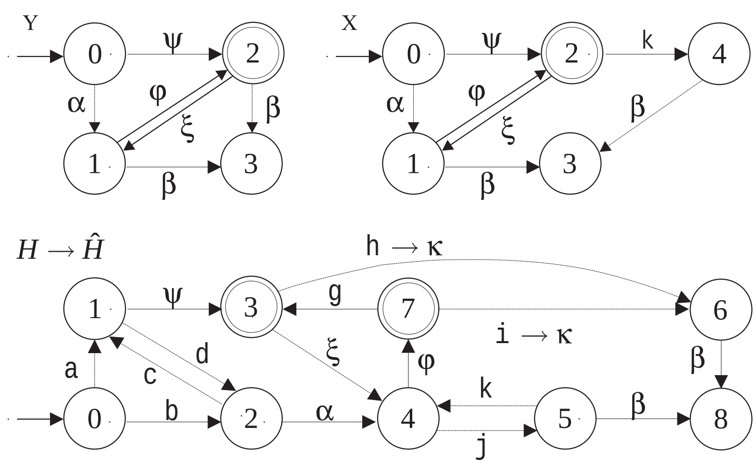

Fig. 2. Automaton with relabeling

A solution to the problem is obtained if the low-level transitions from state 3 and 7 to state 6 are relabeled κ (as indicated in Figure 2) and the new high-level alphabet $\hat{\Sigma}^{\rm hi} = \Sigma^{\rm hi} \cup \{\kappa\}$ is used.[3] The PCS $(\hat{H}, \hat{H}^{\rm hi})$ fulfills condition c.

Thus, the question arises if there is a systematic way to determine $\hat{\Sigma}^{\rm hi}$ such that condition c. holds by adding high-level observations. The next section provides an algorithm for computing the minimal $\hat{\Sigma}^{\rm hi}$

[2]Note that $s^{\rm hi} \in L(H^{\rm hi}) - L_{\rm m}(H^{\rm hi}) \Rightarrow (p^{\rm hi})^{-1}(s^{\rm hi}) \cap L_{\rm m}(H) = \emptyset$.

[3]relabeling in H just changes the observation sent to the high level.

meeting condition c. The corresponding natural projection is called an *msa-observer*.

IV. Computation of MSA-Observers

We first present basic results from set theory which are used to prove the existence of msa-observers.

A. Basic Notation

We denote $\mathcal{E}(M)$ the set of all equivalence relations on the set M. For $\mu \in \mathcal{E}(M)$, $[m]_\mu$ is the equivalence class containing $m \in M$. The set of equivalence classes of μ is written as $M/\mu := \{[m]_\mu | m \in M\}$ and the canonical projection $\mathrm{cp}_\mu : M \to M/\mu$ maps an element $m \in M$ to its equivalence class $[m]_\mu$. Let $f : M \to N$ be a function. The equivalence relation $\ker f$ is the kernel of f and is defined as follows: for $m, m' \in M$, $m \equiv m' \mod \ker f$ iff $f(m) = f(m')$.

Given two equivalence relations η and μ on M, $\eta \leq \mu$, i.e. η refines μ, if $m \equiv m' \mod \eta \Rightarrow m \equiv m' \mod \mu$ for all $m, m' \in M$. With the partial order $\leq$, we denote $\vee$ and $\wedge$ as the join and the meet operation of the lattice $\mathcal{E}(M)$.

Let M and N be sets and $f : M \to 2^N$ be a function. Also assume $\varphi \in \mathcal{E}(N)$. The equivalence relation $\varphi \circ f$ on M is defined for all $m, m' \in M$:[4]

$$m \equiv m' \mod \varphi \circ f \Leftrightarrow \mathrm{cp}_\varphi(f(m)) = \mathrm{cp}_\varphi(f(m')),$$

Now let $f : M \to 2^M$. $\varphi \in \mathcal{E}(M)$ is called a *quasi-congruence* for (M, f) if $\varphi \leq \varphi \circ f$. The quasi-congruences for f form a complete upper semilattice of the lattice $\mathcal{E}(M)$ [13]. Furthermore, if $\mu, \eta \in \mathcal{E}(M)$ s.t. $\mu \leq \eta$, the equivalence relation $\eta/\mu \in \mathcal{E}(M/\mu)$ is defined s.t. for $m, m' \in M$

$$[m]_\mu \equiv [m']_\mu \mod \eta/\mu \Leftrightarrow m = m' \mod \eta. \tag{2}$$

B. Existence

In this section, the problem discussed in Section III is formally stated and solved for the PS (H, H^{hi}). The set of transitions of the automaton H is denoted $T_H := \{(x, \sigma, x') \in X \times \Sigma \times X | x' = \delta(x, \sigma)\}$. A relabeling from H to $\hat{H}$ is a function $r : T_H \to T_{\hat{H}}$ with $\mathrm{r}((x, \sigma, x')) = (x, \hat{\sigma}, x')$, where $\sigma \in \Sigma$ and $\hat{\sigma} \in \hat{\Sigma}$.

We recall the following result on the prefix-closure function $\mathrm{pre} : \Sigma^* \to 2^{\Sigma^*}$ with $\mathrm{pre}(s) = \overline{\{s\}}$ for $s \in \Sigma^*$ [11]. The kernel $\ker p^{\mathrm{hi}}$ of the natural projection p^{hi} for $L(H)$ is a quasi-congruence for $(L(H), \mathrm{pre})$, i.e. if $s, s' \in L(H)$, then $p^{\mathrm{hi}}(s) = p^{\mathrm{hi}}(s') \Rightarrow p^{\mathrm{hi}}(\mathrm{pre}(s)) = p^{\mathrm{hi}}(\mathrm{pre}(s'))$. Also, for any quasi-congruence μ on

[4] The natural extension of cp_φ to sets is used.

$(L(H),\mathrm{pre})$, there is a relabeling $r : T_H \to T_{\hat{H}}$ with the natural projection $\hat{p}^{\mathrm{hi}} : \hat{\Sigma}^* \to (\hat{\Sigma}^{\mathrm{hi}})^*$ for $L(\hat{H})$ such that $\ker \hat{p}^{\mathrm{hi}} = \mu$.

Based on the above notions, the problem in Section III is formalized.

Problem 1: Let H be an automaton and p^{hi} be the natural projection. Find (i) the coarsest quasi-congruence μ on $(L(H),\mathrm{pre})$ that is finer than $\ker p^{\mathrm{hi}}$, and (ii) a relabeling $r : T_H \to T_{\hat{H}}$ and a natural projection $\hat{p}^{\mathrm{hi}} : \hat{\Sigma}^* \to \hat{\Sigma}^{\mathrm{hi}}$ with $\ker \hat{p}^{\mathrm{hi}} = \mu$ and such that $(\hat{H},\hat{H}^{\mathrm{hi}})$ fulfills condition 3., i.e. it is locally nonblocking and marked string accepting.

Regarding Definition 3.1 and 3.2, the following two postsets for languages are needed to find a quasi-congruence as stated in Problem 1.

Definition 4.1 (Postsets): Let H and p^{hi} be as above and let $M \subseteq L(H)$. The *local postset* of $s \in L(H)$ is $\mathrm{pos}_M(s) := \{u \in (\Sigma - \Sigma^{\mathrm{hi}})^*\Sigma(\Sigma - \Sigma^{\mathrm{hi}})^* | su \in M\}$. The *marked string accepting postset* of $s \in L(H)$ is defined as

$$\mathrm{pos}_M^{\mathrm{msa}}(s) := \begin{cases} \emptyset & \text{if (1) holds} \\ & \forall s_{\mathrm{ex}} \in L_{\mathrm{ex},p^{\mathrm{hi}}(s)} \text{ s.t. } s \le s_{\mathrm{ex}} \\ \mathrm{pos}_M(s) & \text{else} \end{cases}$$

The local postset contains all extensions of s with at most one event in Σ^{hi}. The marked string accepting postset distinguishes strings which violate Definition 3.2. $\mathrm{pos}_M^{\mathrm{msa}}$ maps these strings to the local postset of s. Strings which agree with Definition 3.2 are mapped to the empty set.

The *marked string accepting (msa)-observer* is introduced for formulating Lemma 4.1.

Definition 4.2 (M-MSA-Observer): The natural projection $p^{\mathrm{hi}} : \Sigma^* \to (\Sigma^{\mathrm{hi}})^*$ with $\Sigma^{\mathrm{hi}} \subseteq \Sigma$ is an M-msa-observer for the automaton H with $M \subseteq L(H)$ if $\ker p^{\mathrm{hi}}$ is a quasi-congruence for $(L(H),\mathrm{pre})$, $(L(H),\mathrm{pos}_M)$ and $(L(H),\mathrm{pos}_M^{\mathrm{msa}})$.

The relevance of the M-msa-observer is elaborated in the next Lemma. If the map p^{hi} is a $L(H)-$msa-observer for the language $L(H)$, then the corresponding PS (H,H^{hi}) is locally nonblocking and marked string accepting.

Lemma 4.1 (MSA and LNB): Let H, p^{hi} and H^{hi} be defined as above. The natural projection p^{hi} is a $L(H)$-msa-observer for H if and only if (H,H^{hi}) is locally nonblocking and marked string accepting.

Proof: "$\Rightarrow$": It holds that $\ker p^{\mathrm{hi}}$ is a quasi-congruence for $\mathrm{pos}_{L(H)}$ and $\mathrm{pos}_{L(H)}^{\mathrm{msa}}$.

Let $s^{\mathrm{hi}} \in L(H^{\mathrm{hi}})$ and $s \in L(H)$, s.t. $s^{\mathrm{hi}} = p^{\mathrm{hi}}(s)$ and assume $\sigma \in \Sigma^{\mathrm{hi}}(s^{\mathrm{hi}})$. As $s^{\mathrm{hi}} \in p^{\mathrm{hi}}(L(H))$, there is a $s' \in L(H)$ s.t. $s'\sigma \in L(H)$ and $p^{\mathrm{hi}}(s') = s^{\mathrm{hi}}$. As $\ker p^{\mathrm{hi}}$ is a quasi-congruence for $\mathrm{pos}_{L(H)}$, we have that $p^{\mathrm{hi}}(s') = p^{\mathrm{hi}}(s) \Rightarrow p^{\mathrm{hi}}(\mathrm{pos}_{L(H)}(s')) = p^{\mathrm{hi}}(\mathrm{pos}_{L(H)}(s))$. With $\sigma \in p^{\mathrm{hi}}(\mathrm{pos}_{L(H)}(s'))$, it follows that $\sigma \in p^{\mathrm{hi}}(\mathrm{pos}_{L(H)}(s))$. But then there is a $u \in (\Sigma - \Sigma^{\mathrm{hi}})^*$ s.t. $su\sigma \in L(H)$. As $s \in L(H)$ was arbitrary, (H,H^{hi}) is locally nonblocking.

Let $s^{\mathrm{hi}} \in L_{\mathrm{m}}(H^{\mathrm{hi}})$ and $s \in L_{\mathrm{ex},s^{\mathrm{hi}}}$ s.t. there is no $s' \leq s$ with $p^{\mathrm{hi}}(s') = s^{\mathrm{hi}}$ and $s' \in L_{\mathrm{m}}(H)$. Then $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s) = \mathrm{pos}_{L(H)}(s) \neq \emptyset$. However, as $s^{\mathrm{hi}} \in L_{\mathrm{m}}(H^{\mathrm{hi}})$, there is $s_{\mathrm{m}} \in L_{\mathrm{m}}(H)$ s.t. $p^{\mathrm{hi}}(s_{\mathrm{m}}) = s^{\mathrm{hi}}$. As $s \in L_{\mathrm{ex},s^{\mathrm{hi}}}$, there is $\sigma \in \Sigma^{\mathrm{hi}}(s^{\mathrm{hi}})$. Then there is $s'_{\mathrm{m}} \geq s_{\mathrm{m}}$ s.t. $s'_{\mathrm{m}} \in L_{\mathrm{ex},s^{\mathrm{hi}}}$ since (H, H^{hi}) is locally nonblocking (see above), which means that $\mathrm{pos}_{L(H)}(s_{\mathrm{m}}) = \emptyset$. Then $p^{\mathrm{hi}}(s) = p^{\mathrm{hi}}(s_{\mathrm{m}})$ and $p^{\mathrm{hi}}(\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s)) \neq p^{\mathrm{hi}}(\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s_{\mathrm{m}}))$. As this contradicts the assumption that p^{hi} is a quasi-congruence for $\mathrm{pos}^{\mathrm{msa}}_{L(H)}$ such s does not exist. With Definition 3.2, (H, H^{hi}) is marked string accepting.

"$\Leftarrow$": We assume that (H, H^{hi}) is locally nonblocking and marked string accepting.
Let $s, s' \in L(H)$ s.t. $p^{\mathrm{hi}}(s) = p^{\mathrm{hi}}(s')$ and assume that $u \in \mathrm{pos}_{L(H)}(s)$. There are two cases. If $p^{\mathrm{hi}}(u) = \varepsilon$, then, $u' = \varepsilon \in \mathrm{pos}_{L(H)}(s')$ and $p^{\mathrm{hi}}(s'u') = p^{\mathrm{hi}}(su)$. If $p^{\mathrm{hi}}(u) = \sigma$, then $\sigma \in \Sigma^{\mathrm{hi}}(p^{\mathrm{hi}}(s))$. As (H, H^{hi}) is locally nonblocking, there is $u' \in (\Sigma - \Sigma^{\mathrm{hi}})^*$ s.t. $s'u'\sigma \in L(H)$ which means $u'\sigma \in \mathrm{pos}_{L(H)}(s')$.
As this holds for any $u \in \mathrm{pos}_{L(H)}(s)$, $p^{\mathrm{hi}}(\mathrm{pos}_{L(H)}(s')) = p^{\mathrm{hi}}(\mathrm{pos}_{L(H)}(s))$. Thus p^{hi} is a quasi-congruence for $\mathrm{pos}_{L(H)}$.

Let $s, s' \in L(H)$ s.t. $p^{\mathrm{hi}}(s) = p^{\mathrm{hi}}(s') := s^{\mathrm{hi}}$. Again there are two cases. In the first case, $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s) = \emptyset$. Then for all $s_{\mathrm{ex}} \in L_{\mathrm{ex},s^{\mathrm{hi}}}$ s.t. $s_{\mathrm{ex}} \geq s$, Equation (1) holds. If there is no such s_{ex}, then the locally nonblocking condition implies that $\Sigma^{\mathrm{hi}}(s^{\mathrm{hi}}) = \emptyset$ and thus there is also no $s'_{\mathrm{ex}} \in L_{\mathrm{ex},s^{\mathrm{hi}}}$ s.t. $s'_{\mathrm{ex}} \geq s'$ which means that $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s') = \emptyset$. In case there is s_{ex} as defined above, the locally nonblocking condition states that there is a $s'_{\mathrm{ex}} \in L_{\mathrm{ex},s^{\mathrm{hi}}}$ s.t. $s'_{\mathrm{ex}} \geq s'$. As (H, H^{hi}) is marked string accepting, Equation (1) holds for all such s'_{ex}. Consequently $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s') = \emptyset$. In the second case $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s) \neq \emptyset$. Because of the above proof $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s') \neq \emptyset$. (If $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s') = \emptyset$, then also $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s) = \emptyset$) According to Definition 4.1, it is the case that $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s) = \mathrm{pos}_{L(H)}(s)$ and $\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s') = \mathrm{pos}_{L(H)}(s')$. As $\mathrm{pos}_{L(H)}(s') = \mathrm{pos}_{L(H)}(s)$ was shown above, also $p^{\mathrm{hi}}(\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s')) = p^{\mathrm{hi}}(\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s))$.

In both cases $p^{\mathrm{hi}}(\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s')) = p^{\mathrm{hi}}(\mathrm{pos}^{\mathrm{msa}}_{L(H)}(s))$ and consequently p^{hi} is a quasi-congruence for $\mathrm{pos}^{\mathrm{msa}}_{L(H)}$. Together, p^{hi} is an $L(H)$-msa-observer for H. ■

Considering Lemma 4.1 and Problem 1, we want to determine the coarsest quasi-congruence which is finer than the kernel $\ker p^{\mathrm{hi}}_{\mathrm{in}}$ of an initial natural projection $p^{\mathrm{hi}}_{\mathrm{in}}$.

$$\begin{aligned} \pi^*_{\mathrm{msa}} \quad := \quad & \sup\{\pi \in \mathcal{E}(L(H)) | \pi \leq (\ker p^{\mathrm{hi}}_{\mathrm{in}}) \wedge \\ & (\pi \circ \mathrm{pre}) \wedge (\pi \circ \mathrm{pos}_{L(H)}) \wedge (\pi \circ \mathrm{pos}^{\mathrm{msa}}_{L(H)}\}. \end{aligned} \tag{3}$$

The supremal element π^*_{msa} exists as the quasi-congruences form a complete upper semilattice of the lattice $\mathcal{E}(L(H))$.

Theorem 4.1: $\mu = \pi^*_{\mathrm{msa}}$ in Equation (3) is the quasi-congruence which solves Problem 1 (i).

C. Algorithmic Computation

Having shown the existence of the solution to Problem 1 (i), the corresponding msa-observer in Problem 1 (ii) is determined in this section. The algorithm follows the iterative procedure in [11].

Let μ be an equivalence relation on the state set X of H with the quotient set $Y := X/\mu$ and the associated canonical projection $\mathrm{cp}_\mu : X \to Y$. The initial state and the marked states in the quotient are $y_0 = \mathrm{cp}_\mu(x_0)$ and $Y_\mathrm{m} = \mathrm{cp}_\mu(X_\mathrm{m})$, respectively. Also let $\Sigma^\mathrm{hi} \subseteq \Sigma$ and $\Sigma^\mathrm{hi} \notin \Sigma$ be an additional label. The induced transition function $\nu : Y \times (\Sigma^\mathrm{hi} \cup \{\Sigma^\mathrm{hi}\}) \to 2^Y$ on the quotient is defined as

$$\nu(y,\sigma) := \begin{cases} \{\mathrm{cp}_\mu(\delta(x,\sigma)) | x \in \mathrm{cp}_\mu^{-1}(y)\} & \text{if } \sigma \in \Sigma^\mathrm{hi} \\ \{\mathrm{cp}_\mu(\delta(x,\gamma)) | \gamma \in (\Sigma - \Sigma^\mathrm{hi}), & \\ x \in \mathrm{cp}_\mu^{-1}(y)\} - \{y\} & \text{if } \sigma = \Sigma^\mathrm{hi} \end{cases}$$

We call $H_{\mu,\Sigma^\mathrm{hi}} := (Y, \Sigma^\mathrm{hi} \cup \{\Sigma^\mathrm{hi}\}, \nu, y_0, Y_\mathrm{m})$ the *quotient automaton* of H for Σ^hi and μ.

In order to determine the msa-observer and similar to the postsets in Definition 4.1, the *successor event transition function* and the *nonmarked transition function* are used.

Definition 4.3: Let H and $\Sigma^\mathrm{hi} \subseteq \Sigma$ be as above. Let $x = \delta(x_0, s)$ for $s \in L(H)$. The successor event transition function $\Delta_\sigma : X \to 2^X$ is defined for $\sigma \in \Sigma^\mathrm{hi}$ as

$$\Delta_\sigma(x) := \{\delta(x,u) | u \in \mathrm{pos}_{L(H)}(s) \cap (\Sigma - \Sigma^\mathrm{hi})^* \sigma (\Sigma - \Sigma^\mathrm{hi})^*\}. \tag{4}$$

The nonmarked transition function $\Delta_\mathrm{nm} : X \to 2^X$ is

$$\Delta_\mathrm{nm}(x) := \begin{cases} \bigcup_{\sigma \in \Sigma^\mathrm{hi}} \Delta_\sigma(x) & \text{if } \mathrm{pos}^\mathrm{msa}_{L(H)}(s) \neq \emptyset \\ \emptyset & \text{else} \end{cases} \tag{5}$$

With (4) and (5), the coarsest quasi-congruence μ_H for H and Σ^hi can be evaluated as

$$\mu_H := \sup\{\mu \in \mathcal{E}(X) | \mu \leq \bigwedge_{\sigma \in \Sigma^\mathrm{hi} \cup \{\mathrm{nm}\}} (\mu \circ \Delta_\sigma)\}. \tag{6}$$

An efficient algorithm for computing μ_H is given in [3]. Based on μ_H, Theorem 4.2 establishes the relation between the quotient $H_{\mu_H,\Sigma^\mathrm{hi}}$ and a $L(H)$-msa-observer.

Theorem 4.2: Let H and p^hi be given as above and let μ_H be the quasi-congruence in Equation (6). p^hi is an $L(H)$-msa-observer iff $H_{\mu_H,\Sigma^\mathrm{hi}}$ is deterministic and contains no σ_0-transitions. In this case, $H_{\mu_H,\Sigma^\mathrm{hi}}$ is a minimal state recognizer of $p^\mathrm{hi}(L_\mathrm{m}(H))$ and can be computed in polynomial time.

In order to prove Theorem 4.2, we introduce the *Nerode equivalence* $\equiv_L$ for a language $L \subseteq \Sigma^*$ [13]. Let $s, s' \in \overline{L}$. Then

$$s \equiv s' \mod \equiv_L \text{ iff } \forall u \in \Sigma^* : su \in L \Leftrightarrow s'u \in L. \tag{7}$$

For an automaton H, we define the equivalence relation $\mu_X \in \mathcal{E}(L(H))$ by $s \equiv s' \mod \mu_X \Leftrightarrow \delta(x_0, s) = \delta(x_0, s')$. Then, it holds that if the projection p^{hi} is an msa-observer, the Nerode equivalence on the projected language $p^{\text{hi}}(L_{\text{m}}(H))$ is coarser than μ_X, i.e., $[\equiv_{p^{\text{hi}}(L_{\text{m}}(H))} \circ p^{\text{hi}}]$ partitions the state space of H.

Lemma 4.2: Let H, μ_X be defined as above, and let p^{hi} be an msa-observer. It holds that

$$\mu_X \leq [\equiv_{p^{\text{hi}}(L_{\text{m}}(H))} \circ p^{\text{hi}}]. \tag{8}$$

Lemma 4.2 is shown in Appendix A. Based on this result, we define the equivalence relation $\mu_{p^{\text{hi}}(L_{\text{m}}(H))} := [\equiv_{p^{\text{hi}}(L_{\text{m}}(H))} \circ p^{\text{hi}}]/\mu_H \in \mathcal{E}(L(H)/\mu_X)$, where states in X are equivalent if they exhibit the same future marked behavior under the projection p^{hi}. This statement is further formalized in the following proposition. Here, the function Δ_σ is extended to strings as follows. For $x \in X$, and $t\sigma \in (\Sigma^{\text{hi}})^*$, $\Delta_\varepsilon(x) = \{x\}$ and $\Delta_{t\sigma}(x) = \bigcup\{\Delta_\sigma(x') | x' \in \Delta_t(x)\}$.

Proposition 4.1: Let H, $\mu_{p^{\text{hi}}(L_{\text{m}}(H))}$ be defined as above, let p^{hi} be an msa-observer, and $x, x' \in X$. Then $x \equiv x' \mod \mu_{p^{\text{hi}}(L_{\text{m}}(H))}$ iff

1) $\forall t \in (\Sigma^{\text{hi}})^* - \{\varepsilon\} : \Delta_t(x) \cap X_{\text{m}} \neq \emptyset \Leftrightarrow \Delta_t(x') \cap X_{\text{m}} \neq \emptyset$
2) $\Delta_{\text{nm}}(x) \neq \emptyset \Leftrightarrow \Delta_{\text{nm}}(x') \neq \emptyset$.

Furthermore, $\mu_{p^{\text{hi}}(L_{\text{m}}(H))} = \mu_H$.

The proof of Proposition 4.1 is provided in Appendix A. Now, Theorem 4.2 can be shown.

Proof: "'$\Rightarrow$"': p^{hi} is an msa-observer. It has to be shown that $H_{\mu_H,\Sigma^{\text{hi}}}$ is deterministic and contains no σ_0-transitions.

We first prove that $H_{\mu_H,\Sigma^{\text{hi}}}$ is deterministic. Assume the contrary and denote the state set and the transition function of $H_{\mu_H,\Sigma^{\text{hi}}}$ as Y and ν, respectively. Then, there are $y, \hat{y}, \hat{y}' \in Y$, $\sigma \in \Sigma^{\text{hi}}$ s.t. $\{\hat{y}, \hat{y}'\} \subseteq \nu(y, \sigma)$, and there exist $x, x', \hat{x}, \hat{x}' \in X$ with $\{x, x'\} \subseteq cp_{\mu_H}^{-1}(y)$, $\hat{x} \in cp_{\mu_H}^{-1}(\hat{y})$, $\hat{x}' \in cp_{\mu_H}^{-1}(\hat{y}')$ and $\delta(x, \sigma) = \hat{x}$, $\delta(x', \sigma) = \hat{x}'$ according to the QA construction. Let $s, s' \in \Sigma^*$ s.t. $\delta(x_0, s) = x$ and $\delta(x_0, s') = x'$. Then, $\hat{x} \not\equiv \hat{x}' \mod \mu_H$ implies that $\hat{x} \not\equiv \hat{x}' \mod \mu_{p^{\text{hi}}(L_{\text{m}}(H))}$, i.e., w.l.o.g., there is $t \in (\Sigma^{\text{hi}})^*$ s.t. $p^{\text{hi}}(s\sigma)t \in p^{\text{hi}}(L_{\text{m}}(H))$ but $p^{\text{hi}}(s'\sigma)t \notin p^{\text{hi}}(L_{\text{m}}(H))$. But then, $p^{\text{hi}}(s)\sigma t \in p^{\text{hi}}(L_{\text{m}}(H))$, while $p^{\text{hi}}(s')\sigma t \notin p^{\text{hi}}(L_{\text{m}}(H))$ contradicts that $x \equiv x' \mod \mu_{p^{\text{hi}}(L_{\text{m}}(H))}$.

We now show that $H_{\mu_H,\Sigma^{\text{hi}}}$ does not have σ_0-transitions. Assume the contrary. Then, there are $y, \hat{y} \in Y$ s.t. $\hat{y} \in \nu(y, \Sigma^{\text{hi}})$ and $x, \hat{x} \in X$, $\sigma \in \Sigma - \Sigma^{\text{hi}}$ s.t. $x \in cp_{\mu_H}^{-1}(y)$, $\hat{x} \in cp_{\mu_H}^{-1}(\hat{y})$, and $\hat{x} = \delta(x, \sigma)$ according to the QA construction. Again, since $x \not\equiv \hat{x} \mod \mu_H$, also $x \not\equiv \hat{x} \mod \mu_{p^{\text{hi}}(L_{\text{m}}(H))}$. Then, w.l.o.g., there is $t \in (\Sigma^{\text{hi}})^*$ s.t. $p^{\text{hi}}(s)t \in p^{\text{hi}}(L_{\text{m}}(H))$ but $p^{\text{hi}}(s\sigma)t \notin p^{\text{hi}}(L_{\text{m}}(H))$. Since $p^{\text{hi}}(s\sigma)t = p^{\text{hi}}(s)t$, this leads to contradiction.

"'$\Leftarrow$"': $H_{\mu_H,\Sigma^{\text{hi}}}$ is deterministic and contains no σ_0-transitions. It has to be shown that p^{hi} is an msa-observer.

We first prove that p^{hi} is an $L(H)$-observer. Let $s \in L(H)$ and $p^{\text{hi}}(s)t \in p^{\text{hi}}(L(H))$ for $t \in (\Sigma^{\text{hi}})^*$. Let $x = \delta(x_0, s)$ and $y := cp_{\mu_H}(x)$. Then, there are two cases. If $\nu(y,t)!$, then $\Delta_t(x) \neq \emptyset$. Hence, there is $u \in \Sigma^*$ s.t. $su \in L(H)$ and $p^{\text{hi}}(su) = p^{\text{hi}}(s)t$. If $\nu(y,t)$ does not exist, there must be $y' \neq y$ s.t. $\nu(y',t)!$ and $x' \in cp_{\mu_H}^{-1}(y')$ with $\delta(x_0, s') = x'$ and $p^{\text{hi}}(s') = p^{\text{hi}}(s)$. But this is only possible if $H_{\mu_H,\Sigma^{\text{hi}}}$ is nondeterministic or contains σ_0-transitions, which leads to contradiction.

We finally show that p^{hi} fulfills (1) in Definition 3.2. Let $t \in p^{\text{hi}}(L_{\text{m}}(H))$ and $s\sigma \in (p^{\text{hi}})^{-1}(t)\Sigma^{\text{hi}} \cap L(H)$. It has to be shown that there is $s' \leq s$ with $p^{\text{hi}}(s') = p^{\text{hi}}(s)$ and $s' \in L_{\text{m}}(H)$. We write $x := \delta(x_0, s)$ and $y := cp_{\mu_H}(x)$. Since $t \in p^{\text{hi}}(L_{\text{m}}(H))$, there is $\hat{s} \in L_{\text{m}}(H)$ s.t. $p^{\text{hi}}(\hat{s}) = t$. Let $\hat{x} := \delta(x_0, \hat{s})$ and $\hat{y} := cp_{\mu_H}(\hat{x})$. Then $\Delta_{\text{nm}}(\hat{x}) = \emptyset$. If $\hat{y} = y$, it must hold that $\Delta_{\text{nm}}(x) = \emptyset$ since μ_H is a quasi-congruence for Δ_{nm}. Then, there exists $s' \leq s$ with $p^{\text{hi}}(s') = p^{\text{hi}}(s)$ and $s' \in L_{\text{m}}(H)$ with the definition of Δ_{nm}. If $\hat{y} \neq y$, we have that $p^{\text{hi}}(s) = p^{\text{hi}}(\hat{s})$ but $y = cp_{\mu_H}(\delta(x_0, s)) \neq \hat{y} = cp_{\mu_H}(\delta(x_0, \hat{s}))$. Then, $H_{\mu_H,\Sigma^{\text{hi}}}$ is nondeterministic or contains σ_0-transitions, which leads to contradiction. ■

The remaining question is how to proceed if $H_{\mu_H,\Sigma^{\text{hi}}}$ is nondeterministic or has σ_0-transitions. Algorithm 1 solves this problem by relabeling transitions in H using $H_{\mu_H,\Sigma^{\text{hi}}}$.

Algorithm 1 (MSA-Observer): **Input:** H, Σ^{hi}.

1. compute μ_H according to Equation (6).
2. compute $H_{\mu_H,\Sigma^{\text{hi}}}$.
3. **if** $H_{\mu_H,\Sigma^{\text{hi}}}$ is deterministic and has no σ_0-transitions
   - $\hat{H} = H$, $\hat{\Sigma}^{\text{hi}} = \Sigma^{\text{hi}}$
   - **terminate**

   else
   - $(\hat{H}, \hat{\Sigma}^{\text{hi}}) = \text{relabel}_{\mu_H}(H, H_{\mu_H,\Sigma^{\text{hi}}}, \Sigma^{\text{hi}})$
   - $H = \hat{H}$, $\Sigma^{\text{hi}} = \hat{\Sigma}^{\text{hi}}$
   - **go to** Step 1.

Output: $\hat{H}, \hat{\Sigma}^{\text{hi}}$.

The relabeling function $\text{relabel}_{\mu_H}(H, H_{\mu_H,\Sigma^{\text{hi}}}, \Sigma^{\text{hi}})$ is implemented by the subsequent algorithm.

Algorithm 2 (relabeling): **Input:** H, $H_{\mu_H,\Sigma^{\text{hi}}}$, Σ^{hi}.

1. $\overline{r} : T_{H_{\mu_H,\Sigma^{\text{hi}}}} \to T_{\hat{H}_{\mu_H,\hat{\Sigma}^{\text{hi}}}}$ relabels $H_{\mu_H,\Sigma^{\text{hi}}}$ to $\hat{H}_{\mu_H,\hat{\Sigma}^{\text{hi}}}$ over $\hat{\Sigma}^{\text{hi}}$ with the following restrictions:
   - $\overline{r}((y,\sigma,y')) = (y,\hat{\sigma},y')$ and $\sigma \neq \hat{\sigma} \Rightarrow \hat{\sigma} \notin (\Sigma \cup \{\Sigma^{\text{hi}}\})$, i.e. always relabel with new labels.
   - if $(y,\hat{\sigma},y') = r(y,\sigma,y')$ and $(z,\hat{\gamma},z') = r(z,\gamma,z')$ with $\sigma \neq \gamma$, then $\hat{\sigma} \neq \hat{\gamma}$, i.e. transitions with different original event labels have different new labels.

2. $r: T_H \to T_{\hat{H}}$ relabels H to $\hat{H}$ according to $\overline{r}$. Assume $(x,\sigma,x') \in T_H$.
   - if $\sigma \in \Sigma^{\text{hi}}$ and $\overline{r}((\text{cp}_{\mu_H}(x),\sigma,\text{cp}_{\mu_H}(x'))) = (\text{cp}_{\mu_H}(x),\hat{\sigma},\text{cp}_{\mu_H}(x'))$ with $\sigma \neq \hat{\sigma}$
   $$\Rightarrow r((x,\sigma,x')) = (x,\hat{\sigma},x').$$
   - if $\sigma \notin \Sigma^{\text{hi}}$ and $\overline{r}((\text{cp}_{\mu_H}(x),\Sigma^{\text{hi}},\text{cp}_{\mu_H}(x'))) = (\text{cp}_{\mu_H}(x),\hat{\sigma},\text{cp}_{\mu_H}(x'))$
   $$\Rightarrow r((x,\sigma,x')) = (x,\hat{\sigma},x').$$

Output: $\hat{H}$, $\hat{\Sigma}^{\text{hi}}$.

The application of Algorithm 1 results in the main theorem of this section.

Theorem 4.3: Algorithm 1 with H and Σ^{hi} terminates in at most $|X|$ steps. If the algorithm stops with the automaton $\hat{H}$ and the alphabet $\hat{\Sigma}^{\text{hi}}$, then it holds for the kernel of the natural projection p^{hi} for $L(\hat{H})$ that $\ker \hat{p}^{\text{hi}} = \pi^*_{\text{msa}}$.

This means that given an automaton H and a high-level alphabet Σ^{hi}, the observer algorithm returns a natural projection $\hat{p}^{\text{hi}}$ for the relabeled automaton $\hat{H}$ such that $(\hat{H},\hat{H}^{\text{hi}})$ is locally nonblocking and marked string accepting. Before extending this result to decentralized DES in Section V, we illustrate Algorithm 1 in Example 4.1.

Example 4.1: Let H be as in Figure 3 with the high-level alphabet $\Sigma^{\text{hi}} = \{\alpha,\beta\}$. We follow the procedure in Algorithm 1. The quasi-congruence μ_H in (6) evaluates to $\mu_H = \{\{0,1,2\},\{3,6,7\},\{4,5\},\{8\}\}$ (for example compare $\Delta_{\text{nm}}(3) = \Delta_{\text{nm}}(6) = \Delta_{\text{nm}}(7) = \emptyset$ and $\Delta_{\text{nm}}(4) = \Delta_{\text{nm}}(5) = \{8\}$). The quotient automaton $H_{\mu_H,\Sigma^{\text{hi}}}$ is shown in Figure 3. It has a nondeterministic event α in state (0,1,2) and two σ_0-transitions. Thus, the corresponding transitions must be relabled in $H_{\mu_H,\Sigma^{\text{hi}}}$ and in H according to Algorithm 2. As an example, we choose $\overline{r}\big(((0,1,2),\alpha,(3,6,7))\big) = ((0,1,2),\psi,(3,6,7))$ and thus $r\big((1,\alpha,3)\big) = (1,\psi,3)$. The resulting PS $(\hat{H},\hat{H}^{\text{hi}})$ with the high-level alphabet $\hat{\Sigma} = \{\alpha,\beta,\varphi,\xi,\psi\}$ is equal to the PS (H,H^{hi}) in Example 3.1. Thus, after one more iteration, the observer algorithm terminiates with the solution $(\hat{H},\hat{\Sigma}^{\text{hi}})$ in Example 3.1.

V. CONSISTENT RELABELING OF DECENTRALIZED DES

The algorithms in Section IV-C provide a method to compute a relabeling and a locally nonblocking and marked string accepting natural projection for a single PS (H,H^{hi}). As the control architecture introduced in Section III involves decentralized projected systems $(||_{i=1}^n G_i, ||_{i=1}^n G_i^{\text{hi}})$, the effect of relabeling one automaton G_k on the overall synchronous behavior has to be investigated. To this end, consider a transition

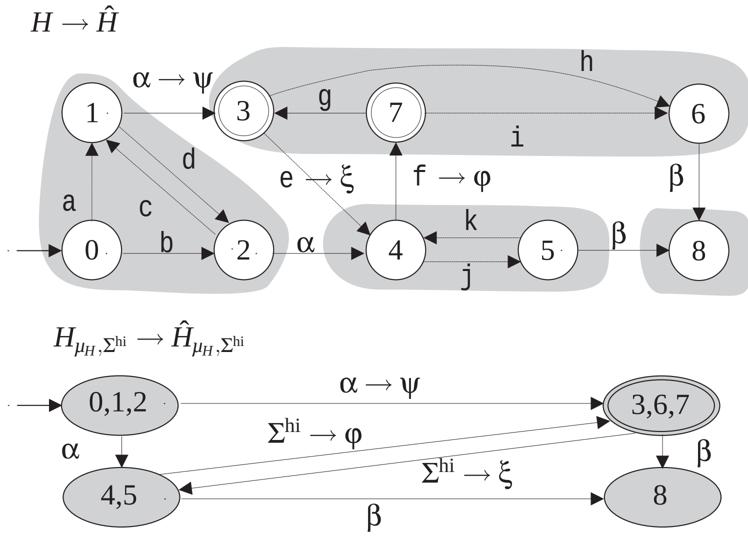

Fig. 3. Illustration of the msa-observer algorithm

$q_k = (x_1, \sigma, x_2) \in T_{G_k}$ which is relabled to (x_1, τ, x_2) in $T_{\hat{G}_k}$, i.e. $r_k((x_1, \tau, x_2)) = q_k$. If σ is not contained in any of the other alphabets, that is $\sigma \notin \Sigma_i$ for all $i \neq k$, there is no effect on the other subsystems as σ occurs asynchronously. In case that $\sigma \in \Sigma_i$ for some $i \neq k$, a relabeling of σ in T_{G_k} changes the synchronous behavior of the decentralized subsystems. We can bypass this problem by adding a new transition containing the event τ for any transition containing σ in the subsystems G_i, $i \neq k$. The following definitions formalize this idea.

The function R_k denotes the map from the relabled events to their original events.

Definition 5.1: Let G_k be an automaton with the relabled automaton $\hat{G}_k$. The map $R_k : \hat{\Sigma}_k \to \Sigma_k$ is defined as

$$R_k(\tau) = \begin{cases} \sigma & \text{if } \exists q_k = (x_1, \tau, x_2) \in T_{\hat{G}_k} \text{ s.t.} \\ & r_k(q) = (x_1, \sigma, x_2) \neq q, \\ \tau & \text{else.} \end{cases}$$

$\overline{R}_k : \hat{\Sigma}_k^* \to \Sigma_k^*$ is the extension of R_k to strings with $\overline{R}_k(\varepsilon) = \varepsilon$ and $\overline{R}_k(\hat{s}\tau) = \overline{R}_k(\hat{s}) R_k(\tau)$ for $\hat{s} \in \hat{\Sigma}_k^*$ and $\tau \in \hat{\Sigma}_k$.

Definition 5.2 (Consistent relabeling): Let $(||_{i=1}^n G_i, ||_{i=1}^n G_i^{\text{hi}})$ be a decentralized projected DES and let $\hat{G}_k$ be a relabeling of G_k with R_k according to Definition 5.1 and the high-level alphabet $\hat{\Sigma}_k^{\text{hi}}$.[5] The tuple $(\hat{G}_i, \hat{\Sigma}_i^{\text{hi}})$, $i \neq k$ is a consistent relabeling of $(G_i, \Sigma_i^{\text{hi}})$ w.r.t. $(\hat{G}_k, \hat{\Sigma}_k^{\text{hi}})$ if (i) $\hat{\Sigma}_i^{\text{hi}} = \Sigma_i^{\text{hi}} \cup \{\tau \in \hat{\Sigma}_k^{\text{hi}} | R_k(\tau) \in \Sigma_i\}$

[5]The corresponding natural projection is $\hat{p}_i^{\text{dec}} : \hat{\Sigma}_i^* \to (\hat{\Sigma}_i^{\text{hi}})^*$.

and (ii) for all $\tau \in \hat{\Sigma}_k$ and $\forall q_i \in T_{G_i}$ such that $q_i = (x_1, R_k(\tau), x_2)$, it holds that $(x_1, \tau, x_2) \in T_{\hat{G}_i}$. The DPS $(||_{i=1}^n \hat{G}_i, ||_{i=1}^n \hat{G}_i^{\mathrm{hi}})$ is a consistent relabeling of $(||_{i=1}^n G_i, ||_{i=1}^n G_i^{\mathrm{hi}})$ w.r.t. $(\hat{G}_k, \hat{\Sigma}_k^{\mathrm{hi}})$ if each tuple $(\hat{G}_i, \hat{\Sigma}_i^{\mathrm{hi}})$, $i \neq k$ is a consistent relabeling of $(G_i, \Sigma_i^{\mathrm{hi}})$ w.r.t. $(\hat{G}_k, \hat{\Sigma}_k^{\mathrm{hi}})$.

It is readily observed, that for all $i = 1, \ldots, n$, it is true that $\overline{R}_k(L(\hat{G}_i)) = L(G_i)$. Yet, it has to be shown that the synchronous behavior of the decentralized systems is not changed by the consistent relabeling. Lemma 5.1 provides this result.

Lemma 5.1 (Consistent relabeling): Let $(||_{i=1}^n \hat{G}_i, ||_{i=1}^n \hat{G}_i^{\mathrm{hi}})$ be a consistent relabeling of $(||_{i=1}^n G_i, ||_{i=1}^n G_i^{\mathrm{hi}})$ w.r.t. $(\hat{G}_k, \hat{\Sigma}_k^{\mathrm{hi}})$ and define the natural projections $p_i : \Sigma^* \to \Sigma_i^*$ and $\hat{p}_i : \hat{\Sigma}^* \to \hat{\Sigma}_i^*$. Then

$$\overline{R}_k(L(\hat{G})) = \overline{R}_k(||_{i=1}^n L(\hat{G}_i)) = ||_{i=1}^n L(G_i) = L(G), \tag{9}$$

$$\overline{R}_k(L(\hat{G}^{\mathrm{hi}})) = \overline{R}_k(||_{i=1}^n L(\hat{G}_i^{\mathrm{hi}})) = ||_{i=1}^n L(G_i^{\mathrm{hi}}) = L(H^{\mathrm{hi}}), \tag{10}$$

and the same equivalence holds for the respective marked languages.

Proof: It has to be shown that (i) $\overline{R}_k(||_{i=1}^n L(\hat{G}_i)) \subseteq ||_{i=1}^n L(G_i)$ and $\overline{R}_k(||_{i=1}^n L(\hat{G}_i^{\mathrm{hi}})) \subseteq ||_{i=1}^n L(G_i^{\mathrm{hi}})$ and (ii) $\overline{R}_k(||_{i=1}^n L(\hat{G}_i)) \supseteq ||_{i=1}^n L(G_i)$ and $\overline{R}_k(||_{i=1}^n L(\hat{G}_i^{\mathrm{hi}})) \supseteq ||_{i=1}^n L(G_i^{\mathrm{hi}})$.

(i) First we show that $\overline{R}_k(\hat{p}_i(\hat{s})) = p_i(\overline{R}_k(\hat{s}))$ by induction. Let $\hat{s} = \varepsilon$. Then $\overline{R}_k(\hat{p}_i(\varepsilon)) = \varepsilon = p_i(\overline{R}_k(\varepsilon))$. Now assume that $\overline{R}_k(\hat{p}_i(\hat{s})) = p_i(\overline{R}_k(\hat{s}))$ holds for $\hat{s} \in L(\hat{G})$ and let $\tau \in \hat{\Sigma}$ s.t. $\hat{s}\tau \in L(\hat{G})$. There are three different cases.

1. $\tau \in \Sigma_i \cap \hat{\Sigma}_i$, i.e. $\overline{R}_k(\tau) = \tau$. Then $\overline{R}_k(\hat{p}_i(\hat{s}\tau)) = \overline{R}_k(\hat{p}_i(\hat{s})\tau) = \overline{R}_k(\hat{p}_i(\hat{s}))\tau = p_i(\overline{R}_k(\hat{s}))\tau = p_i(\overline{R}_k(\hat{s})\tau) = p_i(\overline{R}_k(\hat{s}\tau))$.
2. $\tau \in \hat{\Sigma}_i - \Sigma_i$, i.e. $\overline{R}_k(\tau) = \sigma \in \Sigma_i$. Then $\overline{R}_k(\hat{p}_i(\hat{s}\tau)) = \overline{R}_k(\hat{p}_i(\hat{s})\tau) = \overline{R}_k(\hat{p}_i(\hat{s}))\sigma = p_i(\overline{R}_k(\hat{s}))\sigma = p_i(\overline{R}_k(\hat{s})\sigma) = p_i(\overline{R}_k(\hat{s}\tau))$.
3. $\tau \notin \Sigma_i \cup \hat{\Sigma}_i$, i.e. $\overline{R}_k(\tau) = \tau' \notin \Sigma_i$. Then $\overline{R}_k(\hat{p}_i(\hat{s}\tau)) = \overline{R}_k(\hat{p}_i(\hat{s})) = p_i(\overline{R}_k(\hat{s})) = p_i(\overline{R}_k(\hat{s})\tau') = p_i(\overline{R}_k(\hat{s}\tau))$.

Now we assume that $s \in \overline{R}_k(||_{i=1}^n L(\hat{G}_i))$. Then there is a $\hat{s} \in ||_{i=1}^n L(\hat{G}_i)$ s.t. $\overline{R}_k(\hat{s}) = s$ and consequently $\forall i$, $i = 1, \ldots, n$ it holds that $\hat{p}_i(\hat{s}) \in L(\hat{G}_i)$. As $\overline{R}_k(\hat{p}_i(\hat{s})) = p_i(\overline{R}_k(\hat{s}))$, it follows that $p_i(s) = \overline{R}_k(\hat{p}_i(\hat{s})) \in \overline{R}_k(L(\hat{G}_i))$ $\forall i$. Thus, $s \in ||_{i=1}^n \overline{R}_k(L(\hat{G}_i)) = ||_{i=1}^n L(G_i)$.

(ii) The reverse direction is also proven by induction. It holds that $\varepsilon \in ||_{i=1}^n L(G_i)$ and $\varepsilon \in \overline{R}_k(||_{i=1}^n L(\hat{G}_i))$. Assume $s \in ||_{i=1}^n L(G_i)$ and $s \in \overline{R}_k(||_{i=1}^n L(\hat{G}_i))$ and let $\sigma \in \Sigma$ with $s\sigma \in ||_{i=1}^n L(G_i)$. We note that there is $\hat{s} \in ||_{i=1}^n L(\hat{G}_i)$ s.t. $\overline{R}_k(\hat{s}) = s$. It has to be shown that there is $\tau \in \hat{\Sigma}$ s.t. $\overline{R}_k(\tau) = \sigma$ and $\hat{s}\tau \in ||_{i=1}^n L(\hat{G}_i)$, i.e. $\hat{p}_i(\hat{s}\tau) \in L(\hat{G}_i)$ for all $i = 1, \ldots, n$. There are two cases.

1. $\sigma \notin \Sigma_k \cup \hat{\Sigma}_k$ or $R_k(\sigma) = \sigma$. In both cases no relabeling of the event σ is performed. As $p_i(s)\sigma \in L(G_i)$ for all i s.t. $\sigma \in \Sigma_i$, also $\hat{p}_i(\hat{s}\sigma) \in L(\hat{G}_i)$ because of Definition 5.2. But then $\hat{p}_i(\hat{s}\tau) \in L(\hat{G}_i)$ for all $i = 1, \ldots, n$ if $\tau = \sigma$ is chosen.

2. $\sigma \in \Sigma_k$ and there is $\tau \in \hat{\Sigma}_k$ s.t. $\overline{R}_k(\tau) = \sigma \neq \tau$. As for all i s.t. $\sigma \in \Sigma_i$, $p_i(s)\sigma \in L(G_i)$, Definition 5.2 implies that $\hat{p}_i(\hat{s}\tau) \in L(\hat{G}_i)$. Thus, $\hat{p}_i(\hat{s}\tau) \in L(\hat{G}_i)$ for all $i = 1, \ldots, n$.

$\overline{R}_k(L(\hat{G}^{\mathrm{hi}})) = L(G^{\mathrm{hi}})$ and the proof for the marked languages follow with an analogous argument. ∎

A further beneficial property of the consistent relabeling is stated in Lemma 5.2. Besides the language equivalence, also the locally nonblocking and marked string accepting condition are preserved.

Lemma 5.2: Let $(||_{i=1}^n \hat{G}_i, ||_{i=1}^n \hat{G}_i^{\mathrm{hi}})$ be a consistent relabeling of $(||_{i=1}^n G_i, ||_{i=1}^n G_i^{\mathrm{hi}})$ w.r.t. $(\hat{G}_k, \hat{\Sigma}_k^{\mathrm{hi}})$. If the projected system (G_i, G_i^{hi}) is marked string accepting and locally nonblocking, then the projected system $(\hat{G}_i, \hat{G}_i^{\mathrm{hi}})$ is also marked string accepting and locally nonblocking.

Proof: Assume that $\hat{s}^{\mathrm{hi}} \in L(\hat{G}_i^{\mathrm{hi}})$ and $\tau \in \hat{\Sigma}_i^{\mathrm{hi}}(\hat{s}^{\mathrm{hi}})$. Then $s^{\mathrm{hi}} := \overline{R}_k(\hat{s}^{\mathrm{hi}}) \in L(G_i^{\mathrm{hi}})$ and because of the choice of $\hat{\Sigma}_i^{\mathrm{hi}}$, $R_k(\tau) \in \Sigma_i^{\mathrm{hi}}$. Now suppose that $\hat{s} \in L(\hat{G}_i)$ s.t. $\hat{p}_i^{\mathrm{dec}}(\hat{s}) = \hat{s}^{\mathrm{hi}}$. Then also $s := \overline{R}(\hat{s}) \in L(G_i)$ and $p_i^{\mathrm{dec}}(s) = s^{\mathrm{hi}}$. As (G_i, G_i^{hi}) is locally nonblocking, it follows that there is a $u \in (\Sigma_i - \Sigma_i^{\mathrm{hi}})^*$ s.t. $suR_k(\tau) \in L(G_i)$. Because of Definition 5.2, $\hat{s}u\tau \in L(\hat{G}_i)$. As this holds for arbitrary $\hat{s}^{\mathrm{hi}} \in L(\hat{G}_i^{\mathrm{hi}})$ and $\hat{s} \in L(\hat{G}_i)$, $(\hat{G}_i, \hat{G}_i^{\mathrm{hi}})$ is locally nonblocking.

Now let $\hat{s}^{\mathrm{hi}} \in L_{\mathrm{m}}(\hat{G}_i^{\mathrm{hi}})$ and $\hat{s} \in \hat{L}_{i,\mathrm{ex},\hat{s}^{\mathrm{hi}}}$. Then $\overline{R}_k(\hat{s}^{\mathrm{hi}}) \in L_{\mathrm{m}}(G_i^{\mathrm{hi}})$ and $s := \overline{R}_k(\hat{s}) \in L_{i,\mathrm{ex},\overline{R}_k(\hat{s}^{\mathrm{hi}})}$ because of Definition 5.2. As (G_i, G_i^{hi}) is marked string accepting, there is a $s' \leq s$ with $p_i^{\mathrm{dec}}(s') = \overline{R}_k(\hat{s}^{\mathrm{hi}})$ and $s' \in L_{\mathrm{m}}(G_i)$. Because of Lemma 5.1, there is a $\hat{s}' \in L_{\mathrm{m}}(\hat{G}_i)$ s.t. $\overline{R}_k(\hat{s}') = s'$. With Definition, 5.2 $\hat{s}' \leq \hat{s}$ and $\hat{p}_i^{\mathrm{dec}}(\hat{s}') = \hat{s}^{\mathrm{hi}}$. Hence, $(\hat{G}_i, \hat{G}_i^{\mathrm{hi}})$ is marked string accepting. ∎

Using Lemma 5.1 and Lemma 5.2, we develop an iterative relabeling algorithm. As stated in Theorem 5.1, it results in a decentralized projected DES which is suitable for hierarchical and decentralized control according to [8].

Algorithm 3 (Decentralized relabeling):

Input: $(||_{i=1}^n G_i, ||_{i=1}^n G_i^{\mathrm{hi}})$

1. Initialize $k = 0$.
2. $k := k + 1$,
   compute $L(G_k)$-msa-observer $\hat{p}_k^{\mathrm{hi}}$ for $(\hat{G}_k, \hat{G}_k^{\mathrm{hi}})$ from (G_k, G_k^{hi}) using Algorithm 1,
   determine $\overline{R}_k$ as in Definition 5.1.
3. compute $(||_{i=1}^n \hat{G}_i, ||_{i=1}^n \hat{G}_i^{\mathrm{hi}})$ as consistent relabeling of $(||_{i=1}^n G_i, ||_{i=1}^n G_i^{\mathrm{hi}})$ w.r.t. $(\hat{G}_k, \hat{\Sigma}_k^{\mathrm{hi}})$ according to Definition 5.2.
4. **if** $k = n$
   - **terminate**

   else

- $(||_{i=1}^{n} G_i, ||_{i=1}^{n} G_i^{\mathrm{hi}}) := (||_{i=1}^{n} \hat{G}_i, ||_{i=1}^{n} \hat{G}_i^{\mathrm{hi}})$.

- **go to** step 2.

Output: $(||_{i=1}^{n} \hat{G}_i, ||_{i=1}^{n} \hat{G}_i^{\mathrm{hi}})$, $\{\overline{R}_1, \ldots, \overline{R}_n\}$.

Theorem 5.1: Let $(||_{i=1}^{n} G_i, ||_{i=1}^{n} G_i^{\mathrm{hi}})$ be a decentralized projected DES and let $(||_{i=1}^{n} \hat{G}_i, ||_{i=1}^{n} \hat{G}_i^{\mathrm{hi}})$ be the output of Algorithm 3 applied to $(||_{i=1}^{n} G_i, ||_{i=1}^{n} G_i^{\mathrm{hi}})$. Then all projected systems $(\hat{G}_i, \hat{G}_i^{\mathrm{hi}})$ are marked string accepting and locally nonblocking. Additionally, $\overline{R}_1 \circ \cdots \circ \overline{R}_n(L(\hat{G})) = L(G)$ and $\overline{R}_1 \circ \cdots \circ \overline{R}_n(L(\hat{G}^{\mathrm{hi}})) = L(G^{\mathrm{hi}})$.

Proof: The proof of Theorem 5.1 follows by successive application of Lemma 5.1 and Lemma 5.2. ■

Theorem 5.1 suggests the following hierarchical control design for decentralized DES $||_{i=1}^{n} G_i$. Starting from the natural projection p_i^{dec} on the set of shared events $\Sigma_i^{\mathrm{hi}} := \bigcup_{j \neq i}(\Sigma_i \cap \Sigma_j)$, Algorithm 3 can be applied to the decentralized projected DES $(||_{i=1}^{n} G_i, ||_{i=1}^{n} G_i^{\mathrm{hi}})$. As all PSs $(\hat{G}_i, \hat{G}_i^{\mathrm{hi}})$ of the resulting decentralized projected DES $(||_{i=1}^{n} \hat{G}_i, ||_{i=1}^{n} \hat{G}_i^{\mathrm{hi}})$ are locally nonblocking and marked string accepting, the hierarchical and decentralized approach in [8] can be applied. The following example illustrates the procedure.

Example 5.1: Let $||_{i=1}^{2} G_i$ be the decentralized DES with G_1 and G_2 as in Figure 4. The initial natural projection on the shared events is $p_i^{\mathrm{dec}} : \Sigma_i^* \to (\Sigma_1 \cap \Sigma_2)^*$, where $\Sigma_1 \cap \Sigma_2 = \{\alpha, \beta\}$. It results in the decentralized projected system $(||_{i=1}^{2} G_i, ||_{i=1}^{2} G_i^{\mathrm{hi}})$. We apply Algorithm 3 to $(||_{i=1}^{2} G_i, ||_{i=1}^{2} G_i^{\mathrm{hi}})$. Observing that (G_1, G_1^{hi}) is locally nonblocking and marked string accepting (step 2.), no relabeling has to be performed for $k = 1$ in step 3. Thus, the next iteration for $k = 2$ again starts with the original decentralized projected system $(||_{i=1}^{2} G_i, ||_{i=1}^{2} G_i^{\mathrm{hi}})$. As G_2 equals H in Example 4.1, the msa-observer $\hat{p}_2^{\mathrm{dec}}$ computed in step 2. of the algorithm, is the same as $\hat{p}^{\mathrm{hi}}$ in Example 3.1. Because of this reason, the relabled automaton $\hat{G}_2$ in Figure 4 equals $\hat{H}$ in Example 3.1. In step 3., G_1 has to be relabled according to Definition 5.2. It holds that $\hat{\Sigma}_1^{\mathrm{hi}} = \Sigma_1^{\mathrm{hi}} \cup \{\psi\}$, as $R_2(\psi) = \alpha \in \Sigma_1^{\mathrm{hi}}$. As the transition $(1, \alpha, 2)$ is in T_{G_1}, the transition $(1, \psi, 2)$ has to be added to $T_{\hat{G}_1}$ to comply with Definition 5.2 (see Figure 4). The algorithm terminates in step 4. with $(||_{i=1}^{2} \hat{G}_i, ||_{i=1}^{2} \hat{G}_i^{\mathrm{hi}})$. Note that both $(\hat{G}_1, \hat{G}_1^{\mathrm{hi}})$ and $(\hat{G}_2, \hat{G}_2^{\mathrm{hi}})$ are locally nonblocking and marked string accepting. It can also be verified that the synchronous behavior of the decentralized systems is not changed by the relabeling procedure.

VI. Conclusions

A hierarchical and decentralized control architecture which reduces the computational complexity of DES controller synthesis for large-scale composed systems was elaborated in [8]. Nonblocking and hier-

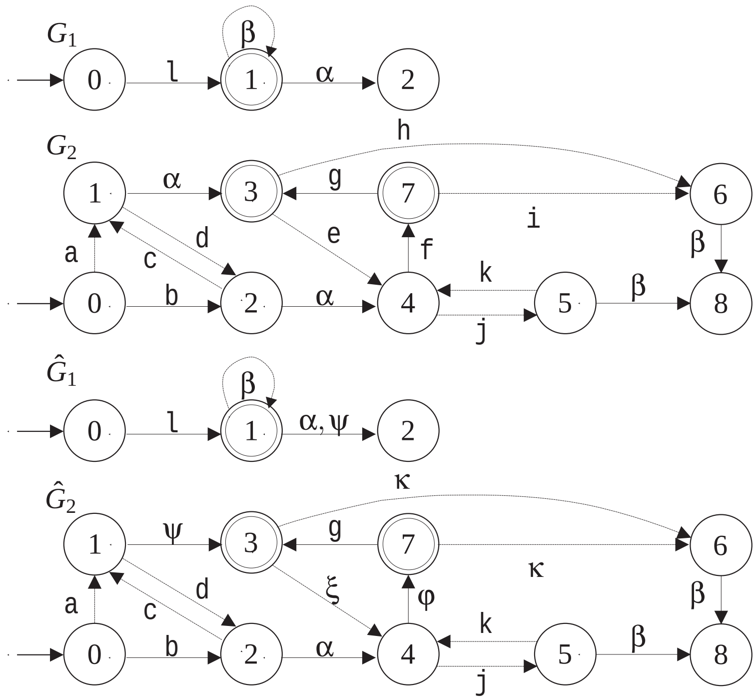

Fig. 4. Application of Algorithm 3 to $||_{i=1}^{2} G_i$

archically consistent control can be guaranteed if the natural projection used for hierarchical sbstraction is (i) *locally nonblocking* and (ii) *marked string accepting* for each subsystem. In this paper we investigated the problem of automatically determining a natural projection such that (i) and (ii) are fulfilled. To this end, we first provided an algorithm which computes the natural projection with the coarsest equivalence kernel that is finer than that of an initial natural projection for an individual subsystem. In our case, the initial natural projection is given by the natural projection on the *shared events* of the composed system. Using this fact and applying the above method for all subsystems of a given composed system, we developed an algorithm which computes the coarsest hierarchical abstraction complying with the method for large-scale composed systems in [8].

APPENDIX A

PROOF OF PROPOSITION 4.1 AND LEMMA 4.2

We first elaborate properties of msa-observers.

Lemma A.1: Let $p^{\mathrm{hi}}:\Sigma^* \to (\Sigma^{\mathrm{hi}})^*$ be an msa-observer for the language $L \subseteq \Sigma^*$, and $s \in \overline{L}$. Then the following holds.

1) if $p^{\mathrm{hi}}(s) \in p^{\mathrm{hi}}(L)$, then either $\exists u \in (\Sigma-\Sigma^{\mathrm{hi}})^*$ s.t. $su \in L$ or $\exists \tilde{s} \le s$ s.t. $p^{\mathrm{hi}}(\tilde{s}) = p^{\mathrm{hi}}(s)$ and $\tilde{s} \in L$.
2) if $t \in (\Sigma^{\mathrm{hi}})^* - \{\varepsilon\}$ and $p^{\mathrm{hi}}(s)t \in p^{\mathrm{hi}}(L)$, then $\exists u \in \Sigma^*$ s.t. $su \in L$ and $p^{\mathrm{hi}}(su) = p^{\mathrm{hi}}(s)t$.

Proof: Let $s \in \overline{L}$. To show 1), we assume that $p^{\mathrm{hi}}(s) \in p^{\mathrm{hi}}(L)$. Then, either $\exists u \in (\Sigma-\Sigma^{\mathrm{hi}})^*$ s.t. $su \in L$ or there is no such u. In the latter case, it must hold that there are $\tilde{u} \in (\Sigma-\Sigma^{\mathrm{hi}})^*$, $\tilde{u}' \in \Sigma^*$ and $\sigma \in \Sigma^{\mathrm{hi}}$ s.t. $s\tilde{u}\sigma\tilde{u}' \in L$, since $s \in \overline{L}$. Considering that p^{hi} is an msa-observer, there must be $\tilde{s} \le s\tilde{u}$ s.t. $p^{\mathrm{hi}}(\tilde{s}) = p^{\mathrm{hi}}(s\tilde{u})$ and $\tilde{s} \in L$. But then, also $\tilde{s} \le s$ and $p^{\mathrm{hi}}(\tilde{s}) = p^{\mathrm{hi}}(s)$.

To show 2), let $t \in (\Sigma^{\mathrm{hi}})^* - \{\varepsilon\}$ and $p^{\mathrm{hi}}(s)t \in p^{\mathrm{hi}}(L)$. Since p^{hi} is an $\overline{L}$-observer, there are $\tilde{u} \in \Sigma^*$, $\sigma \in \Sigma^{\mathrm{hi}}$ s.t. $s\tilde{u}\sigma \in \overline{L}$ and $p^{\mathrm{hi}}(s\tilde{u}\sigma) = p^{\mathrm{hi}}(s)t$. Then, either $\exists \tilde{u}' \in (\Sigma-\Sigma^{\mathrm{hi}})^*$ s.t. $s\tilde{u}\sigma\tilde{u}' \in L$ or there is no such $\tilde{u}'$. In the first case, $u = \tilde{u}\sigma\tilde{u}'$ complies with 2). In the second case, there must be $\hat{u} \in (\Sigma-\Sigma^{\mathrm{hi}})^*$, $\hat{u}' \in \Sigma^*$ and $\hat{\sigma} \in \Sigma^{\mathrm{hi}}$ s.t. $s\tilde{u}\sigma\hat{u}\hat{\sigma}\hat{u}' \in L$. But then, $\tilde{s} \notin L$ for all $s\tilde{u}\sigma \le \tilde{s} \le s\tilde{u}\sigma\hat{u}$, which contradicts that p^{hi} is an msa-observer. ∎

Now, Lemma 4.2 can be proved.

Proof: It has to be shown that $s \equiv s' \mod \mu_H \Rightarrow p^{\mathrm{hi}}(s) \equiv_{p^{\mathrm{hi}}(L_{\mathrm{m}}(H))} p^{\mathrm{hi}}(s')$, i.e., $\forall t \in (\Sigma^{\mathrm{hi}})^*$ it must hold that $p^{\mathrm{hi}}(s)t \in p^{\mathrm{hi}}(L_{\mathrm{m}}(H)) \Leftrightarrow p^{\mathrm{hi}}(s')t \in p^{\mathrm{hi}}(L_{\mathrm{m}}(H))$. Assume that $x := \delta(x_0,s) = \delta(x_0,s')$ and $t \in (\Sigma^{\mathrm{hi}})^*$ s.t. $p^{\mathrm{hi}}(s)t \in p^{\mathrm{hi}}(L_{\mathrm{m}}(H))$. As p^{hi} is an msa-observer, Lemma A.1 implies that there is either $u \in \Sigma^*$ s.t. $su \in L_{\mathrm{m}}(H)$ and $p^{\mathrm{hi}}(su) = p^{\mathrm{hi}}(s)t$ or $\tilde{s} \le s$, $\tilde{u} \in (\Sigma-\Sigma^{\mathrm{hi}})^*$ s.t. $s = \tilde{s}\tilde{u}$ and $\tilde{s} \in L_{\mathrm{m}}(H)$. In the first case, $\delta(x,u)!$, and hence, $s'u \in L_{\mathrm{m}}(H)$ and $p^{\mathrm{hi}}(s'u) = p^{\mathrm{hi}}(s')t \in p^{\mathrm{hi}}(L_{\mathrm{m}}(H))$. In the second case, because of Lemma A.1, there must be $\tilde{s}' \le s'$ and $\tilde{u}' \in (\Sigma-\Sigma^{\mathrm{hi}})^*$ s.t. $s' = \tilde{s}'\tilde{u}'$ and $\tilde{s}' \in L_{\mathrm{m}}(H)$. Then, also $p^{\mathrm{hi}}(s') = p^{\mathrm{hi}}(\tilde{s}') \in p^{\mathrm{hi}}(L_{\mathrm{m}}(H))$, which concludes the proof. ∎

In order to prove Proposition 4.1, we establish three lemmas.

Lemma A.2: Let $\mu \in \mathcal{E}(X)$ be a quasi-congruence for $(X, \{\Delta_\sigma, \sigma \in \Sigma^{\mathrm{hi}}\} \cup \Delta_{\mathrm{nm}})$, and $x, x' \in X$. Then $x \equiv x' \mod \mu$ implies that

1) $\forall t \in (\Sigma^{\mathrm{hi}})^* - \{\varepsilon\} : \Delta_t(x) \cap X_{\mathrm{m}} \neq \emptyset \Leftrightarrow \Delta_t(x') \cap X_{\mathrm{m}} \neq \emptyset$
2) $\Delta_{\mathrm{nm}}(x) \neq \emptyset \Leftrightarrow \Delta_{\mathrm{nm}}(x') \neq \emptyset$.

Proof: To show 1), assume that $t \in (\Sigma^{\mathrm{hi}})^*$ s.t. $\Delta_t(x) \cap X_{\mathrm{m}} \neq \emptyset$. Assume $\hat{x} \in \Delta_t(x) \cap X_{\mathrm{m}}$. We first show that there is $\hat{x}' \in \Delta_t(x')$ s.t. $\hat{x} \equiv \hat{x}' \mod \mu$ by induction. Let $t = \sigma_1 \cdots \sigma_m$, where $\sigma_1 = \varepsilon$ and $\sigma_i \in \Sigma^{\mathrm{hi}}$ for

$i = 2,\ldots,m$. Then, there are $u_i \in (\Sigma - \Sigma^{\rm hi})^*$, $i = 1,\ldots,m$ s.t. $\hat{x} = \delta(x,\sigma_1 u_1 \cdots \sigma_m u_m)$. We denote $x_i := \delta(x,\sigma_1 u_1 \cdots \sigma_i)$ for $i = 1,\ldots,m$.

As the induction base, we observe that $x_1 = x \in \Delta_\varepsilon(x_1)$, $x'_1 := x' \in \Delta_\varepsilon(x'_1)$ and $x_1 \equiv x'_1 \mod \mu$. Now, assume that for some $i \in \{1,\ldots,m-1\}$, $x_i \in \Delta_{\sigma_1\cdots\sigma_i}(x_1)$, $x'_1 \in \Delta_{\sigma_1\cdots\sigma_i}(x'_1)$ and $x_i \equiv x'_i \mod \mu$. Then, $x_{i+1} \in \Delta_{\sigma_{i+1}}(x_i)$. As μ is a quasi-congruence for $\Delta_{\sigma_{i+1}}$, there must be $x'_{i+1} \in \Delta_{\sigma_{i+1}}(x'_i)$ s.t. $x_{i+1} \equiv x'_{i+1} \mod \mu$. Hence, $x_{i+1} \in \Delta_{\sigma_1\cdots\sigma_{i+1}}(x_1)$, $x'_{i+1} \in \Delta_{\sigma_1\cdots\sigma_{i+1}}(x'_1)$, and $x_{i+1} \equiv x'_{i+1} \mod \mu$. But then, induction on the length of t shows that there is $x'_m \in \Delta_t(x')$ s.t. $x_m \equiv x'_m \mod \mu$. In addition, $\hat{x} \in \Delta_t(x) \cap X_{\rm m}$ implies that $\Delta_{\rm nm}(\hat{x}) = \emptyset$. As μ is a quasi-congruence for $\Delta_{\rm nm}$, also $\Delta_{\rm nm}(x'_m) = \emptyset$. Respecting the definition of $\Delta_{\rm nm}$ combined with the fact that H is nonblocking, there must be $u' \in (\Sigma - \Sigma^{\rm hi})^*$ s.t. $\hat{x}' := \delta(x'_m, u') \in X_{\rm m}$. Since $\hat{x}' \in \Delta_t(x')$, we arrive at $\hat{x}' \in \Delta_t(x') \cap X_{\rm m}$, i.e., $\Delta_t(x') \cap X_{\rm m} \neq \emptyset$.

For 2), assume that $\Delta_{\rm nm}(x) \neq \emptyset$. Thus, there is $\hat{x} \in \Delta_{\rm nm}(x)$. Since μ is a quasi-congruence for $\Delta_{\rm nm}$, there must be $\hat{x}' \in \Delta_{\rm nm}(x')$ s.t. $\hat{x} \equiv \hat{x}' \mod \mu$. Hence, $\Delta_{\rm nm}(x') \neq \emptyset$. ■

Lemma A.3: Let $p^{\rm hi}$ be an msa-observer. Then, $\mu_{p^{\rm hi}(L_{\rm m}(H))}$ is a quasi-congruence for $(X, \{\Delta_\sigma, \sigma \in \Sigma^{\rm hi}\} \cup \Delta_{\rm nm})$.

Proof: Let $x, x' \in X$ s.t. $x \equiv x' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$, and $s, s' \in L(H)$ s.t. $x = \delta(x_0, s)$ and $x' = \delta(x_0, s')$.

First assume that $\sigma \in \Sigma^{\rm hi}$ s.t. $\hat{x} \in \Delta_\sigma(x)$. Then, there are $u, \hat{u} \in (\Sigma - \Sigma^{\rm hi})^*$ s.t. $\hat{x} = \delta(x, u\sigma\hat{u})$ and $p^{\rm hi}(su\sigma\hat{u}) = p^{\rm hi}(s)\sigma$. It has to be shown that $\exists \hat{x}' \in \Delta_\sigma(x')$ s.t. $\hat{x} \equiv \hat{x}' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$. Assume that $t \in (\Sigma^{\rm hi})^*$ s.t. $p^{\rm hi}(s)\sigma t \in p^{\rm hi}(L_{\rm m}(H))$. Since $p^{\rm hi}(s) \equiv_{p^{\rm hi}(L_{\rm m}(H))} p^{\rm hi}(s')$ according to the definition of $\mu_{p^{\rm hi}(L_{\rm m}(H))}$, and $p^{\rm hi}(s)\sigma t \in p^{\rm hi}(L_{\rm m}(H))$, also $p^{\rm hi}(s')\sigma t \in p^{\rm hi}(L_{\rm m}(H))$. Since $p^{\rm hi}$ is an msa-observer, it follows from Lemma A.1 that there are $u', \hat{u}' \in (\Sigma - \Sigma^{\rm hi})^*$, $\tilde{u}' \in \Sigma^*$ s.t. $s'u'\sigma\hat{u}'\tilde{u}' \in L_{\rm m}(H)$ and $p^{\rm hi}(s'u'\sigma\hat{u}'\tilde{u}') = p^{\rm hi}(s')\sigma t$. Then, $su\sigma\hat{u} \equiv su'\sigma\hat{u}' \mod \equiv_{p^{\rm hi}(L_{\rm m}(H))} \circ p^{\rm hi}$. Hence, with $\hat{x}' := \delta(x', u'\sigma\hat{u}')$, we have that $\hat{x} \equiv \hat{x}' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$. As $\sigma \in \Sigma^{\rm hi}$ was arbitrary, this implies that $\mu_{p^{\rm hi}(L_{\rm m}(H))}$ is a quasi-congruence for $(X, \{\Delta_\sigma | \sigma \in \Sigma^{\rm hi}\})$.

Now assume that $\hat{x} \in \Delta_{\rm nm}(x)$. Then, $\hat{x} \in \Delta_\sigma(x)$ for some $\sigma \in \Sigma^{\rm hi}$, and as shown above, there is $\hat{x}' \in \Delta_\sigma(x')$ s.t. $\hat{x} \equiv \hat{x}' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$. It remains to show that $\hat{x}' \in \Delta_{\rm nm}(x')$. Since $\hat{x} \in \Delta_{\rm nm}(x)$, there are $\tilde{x} \in X$, $u, v \in (\Sigma - \Sigma^{\rm hi})^*$, $\tilde{\sigma} \in \Sigma^{\rm hi}$ s.t. $x = \delta(\tilde{x}, u)$, $\delta(\tilde{x}uv\tilde{\sigma})!$ and $\tilde{u} \le uv \Rightarrow \delta(\tilde{x}, \tilde{u}) \notin X_{\rm m}$. Since $p^{\rm hi}$ is an msa-observer, this implies that $p^{\rm hi}(s) \notin p^{\rm hi}(L_{\rm m}(H))$. Noting that $p^{\rm hi}(s) \equiv_{p^{\rm hi}(L_{\rm m}(H))} p^{\rm hi}(s')$, also $p^{\rm hi}(s') \notin p^{\rm hi}(L_{\rm m}(H))$. Hence, there are $\tilde{x}' \in X$, $u', v' \in (\Sigma - \Sigma^{\rm hi})^*$, $\tilde{\sigma}' \in \Sigma^{\rm hi}$ s.t. $x' = \delta(\tilde{x}', u')$, $\delta(\tilde{x}'u'v'\tilde{\sigma}')!$ and $\tilde{u}' \le u'v' \Rightarrow \delta(\tilde{x}', \tilde{u}') \notin X_{\rm m}$. Then, $\Delta_\sigma(x') \subseteq \Delta_{\rm nm}(x')$, i.e., $\hat{x}' \in \Delta_{\rm nm}(x')$. ■

Lemma A.4: Assume that 1) and 2) in Lemma A.2 are fulfilled for $x, x' \in X$, and $p^{\rm hi}$ is an msa-observer. Then $x \equiv x' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$.

Proof: Let $s, s' \in L(H)$ s.t. $x = \delta(x_0, s)$, $x' = \delta(x_0, s')$, and assume that $p^{\rm hi}(s)t \in p^{\rm hi}(L_{\rm m}(H))$. We have to show that $p^{\rm hi}(s')t \in p^{\rm hi}(L_{\rm m}(H))$.

If $t = \varepsilon$, we have that $p^{\rm hi}(s) \in p^{\rm hi}(L_{\rm m}(H))$. Since $p^{\rm hi}$ is an msa-observer, either there is $\tilde{s} \leq s$ with $\tilde{s} \in L_{\rm m}(H)$ and $p^{\rm hi}(\tilde{s}) = p^{\rm hi}(s)$ or there is $\hat{s} \geq s$ with $\hat{s} \in L_{\rm m}(H)$ and $p^{\rm hi}(\hat{s}) = p^{\rm hi}(s)$. Considering that this holds for any s s.t. $\delta(x_0, s) = x$, it can be concluded that $\Delta_{\rm nm}(x) = \emptyset$ according to the definition of $\Delta_{\rm nm}$. Then, with 2), $\Delta_{\rm nm}(x') = \emptyset$, which implies that there is no $\tilde{x}' \in X$, $uv \in (\Sigma - \Sigma^{\rm hi})^*$, $\sigma' \in \Sigma^{\rm hi}$ s.t. $x = \delta(\tilde{x}', u)$ and $\delta(\tilde{x}', uv\sigma')!$ and $u' \leq uv \Rightarrow u' \notin L_{\rm m}(H)$. If $p_0(s')\sigma' \notin p_0(L(H))$ for all $\sigma \in \Sigma_0$, the fact that H is assumed to be nonblocking implies that there is a $v' \in (\Sigma - \Sigma^{\rm hi})^*$ s.t. $s'v' \in L_{\rm H}$. Otherwise, there is $\sigma' \in \Sigma^{\rm hi}$ s.t. $p_0(s')\sigma' \in p_0(L(H))$. Hence, either there is $\tilde{s}' \leq s'$ with $\tilde{s}' \in L_{\rm m}(H)$ and $p^{\rm hi}(\tilde{s}') = p^{\rm hi}(s')$ or $\hat{s}' \geq s'$ with $p^{\rm hi}(\hat{s}') = p^{\rm hi}(s')$ and $\hat{s}' \in L_{\rm m}(H)$. It follows that $p^{\rm hi}(s')t = p^{\rm hi}(s') \in p^{\rm hi}(L_{\rm m}(H))$.

If $t \neq \varepsilon$, there is $u \in \Sigma^*$ s.t. $su \in L_{\rm m}(H)$ and $p^{\rm hi}(su) = p^{\rm hi}(s)t$, since $p^{\rm hi}$ is an msa-observer (see Lemma A.1). Hence, $\Delta_t(x) \cap X_{\rm m} \neq \emptyset$. With 1), we have that $\Delta_t(x') \cap X_{\rm m} \neq \emptyset$. Thus, there is $u' \in \Sigma^*$ s.t. $s'u' \in L_{\rm m}(H)$ and $p^{\rm hi}(s'u') = p^{\rm hi}(s')t \in p^{\rm hi}(L_{\rm m}(H))$. That is, $s \equiv s' \mod \equiv_{p^{\rm hi}(L_{\rm m}(H))} \circ p^{\rm hi}$, and hence $x \equiv x' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$. ■

We now prove Proposition 4.1.

Proof: "‘$\Rightarrow$"’: Assume that $x \equiv x' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$. Since $\mu_{p^{\rm hi}(L_{\rm m}(H))}$ is a quasi-congruence according to Lemma A.3, Lemma A.2 ensures that 1) and 2) hold.

"‘$\Leftarrow$"’: If 1) and 2) hold, $x \equiv x' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$ follows from Lemma A.4.

To show that $\mu_{p^{\rm hi}(L_{\rm H})} = \mu_H$, let $\mu' \in \mathcal{E}(X)$ be a quasi-congruence for $(X, \{\Delta_\sigma, \sigma \in \Sigma^{\rm hi}\} \cup \Delta_{\rm nm})$ and $x \equiv x' \mod \mu'$ for $x, x' \in X$. Then, Lemma A.2 suggests that 1) and 2) hold. Hence, Lemma A.4 implies that $x \equiv x' \mod \mu_{p^{\rm hi}(L_{\rm m}(H))}$, i.e., $\mu' \leq \mu_{p^{\rm hi}}$. Hence, $\mu_{p^{\rm hi}(L_{\rm m}(H))}$ is the coarsest quasi-congruence for

$$(X, \{\Delta_\sigma, \sigma \in \Sigma^{\rm hi}\} \cup \Delta_{\rm nm})$$

. ■

REFERENCES

[1] C.G Cassandras and S. Lafortune. Introduction to discrete event systems. *Kluwer*, 1999.

[2] A.E.C. da Cunha, J.E.R. Cury, and B.H. Krogh. An assume guarantee reasoning for hierarchical coordination of discrete event systems. *Workshop on Discrete Event Systems*, 2002.

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

[4] B. Gaudin and H. Marchand. Efficient computation of supervisors for loosely synchronous discrete event systems: A state-based approach. *IFAC World Congress*, 2005.

[5] R.J. Leduc. Hierarchical interface based supervisory control. *PhD thesis, Department of Electrical and Computer Engineering, University of Toronto*, 2002.

[6] S-H. Lee and K.C. Wong. Structural decentralised control of concurrent DES. *European Journal of Control*, 35:1125–1134, October 2002.

[7] C. Ma. Nonblocking supervisory control of state tree structures. *Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Toronto*, 2004.

[8] K. Schmidt. Hierarchical control of decentralized discrete event systems: Theory and application. *Phd-thesis, Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg*, 2005.

[9] K. Schmidt and T. Moor. Computation of marked string accepting observers for discrete event systems. *Workshop on Discrete Event Systems (WODES)*, 2006.

[10] K. Schmidt, T. Moor, and S. Perk. A hierarchical architecture for nonblocking control of discrete event systems. *Mediterranean Conference on Control and Automation*, 2005.

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

[12] K.C. Wong and W.M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*, 1996.

[13] W.M Wonham. Notes on control of discrete event systems. *Department of Electrical Engineering, University of Toronto*, 2004.