Consistent Abstractions for the Purpose of Supervisory Control

Thomas Moor* Christine Baier* Thomas Wittmann*

Abstract— **A common strategy in the design of discrete event systems is to run synthesis algorithms not on the actual plant model but on abstractions thereof. Depending on the control objectives, certain conditions are imposed on the abstractions, in order to end up with an appropriate supervisory controller for the actual plant. A well known result from the literature is that abstractions obtained by so called *natural observers* can be used for non-blocking supervisory control. In this paper, we conduct a backward-reachability analysis to obtain alternative conditions that imply a non-blocking and complete closed-loop system for an abstraction-based supervisory controller design.**

INTRODUCTION

When the plant model provides more detail than required for the controller design problem at hand, one may resort to a plant abstraction instead. A crucial question in such an *abstraction-based controller design* is whether the resulting controller enforces relevant control objectives not only for the abstraction but also for the original plant model.

More specifically, we consider the situation where the plant model is given as a formal language and the *natural projection* to strings of *high-level events* is considered as a candidate for an abstraction; see also [1, 2]. This setting applies to the design of hierarchical control architectures when a group of plant components, each subject to low-level control [8, 9], are composed and the subsequent task is the synthesis of a supervisor that addresses cooperative behaviour, specified w.r.t. high-level events. In the present paper, we rephrase the question, whether the abstraction-based design solves the original problem, as a requirement imposed on the high-level alphabet and we develop conditions to verify this requirement by a reachability analysis.

Our study relates to [13], where within a general framework the notion of an *observer* is defined and proven to be a sufficient condition for the purpose of non-blocking hierarchical controller synthesis. Variations of the *natural observer* property that explicitly take into account controllability are presented in [2] and address minimal restrictive hierarchical supervision [10]. In [4], it is shown that the observer property is not only sufficient but also necessary for *compositional nonblocking verification*, with a further development to address *compositional synthesis* given in [5]. The conditions developed in the present paper are weaker than the requirement of the natural projection to be an observer. This is demonstrated by an example.

Preliminaries and notational conventions are given in Section I to prepare for the technical problem statement in Section II with a solution based on natural observers in Section III. For the class of control problems under consideration, liveness properties can be alternatively addressed by the reachability analysis presented in Section IV. The implementation of the corresponding reachability operators is outlined in Section V.

* Lehrstuhl für Regelungstechnik, Friedrich-Alexander Universität Erlangen-Nürnberg, Germany. lrt@fau.de

I. PRELIMINARIES AND NOTATION

Let Σ be a *finite alphabet*, i.e., a finite set of symbols $\sigma \in \Sigma$. The *Kleene-closure* Σ^* is the set of finite strings $s = \sigma_1\sigma_2\cdots\sigma_n$, $n \in \mathbb{N}$, $\sigma_i \in \Sigma$, and the *empty string* $\epsilon \in \Sigma^*$, $\epsilon \notin \Sigma$. If, for $s, r \in \Sigma^*$, there exists $t \in \Sigma^*$ such that $s = rt$, we say that r is a *prefix* of s, and write $r \leq s$. A *formal language* (or short a *language*) over Σ is a subset $L \subseteq \Sigma^*$.

The *prefix-closure* (or short *prefix*) of a language $L \subseteq \Sigma^*$ is defined by $\operatorname{pre} L := \{r \in \Sigma^* \mid \exists\, s \in L : r \leq s\}$. A language L is *prefix-closed* (or short *closed*) if $L = \operatorname{pre} L$. For two languages L and K, we say K is *relatively closed w.r.t. L* if $K = (\operatorname{pre} K) \cap L$. The prefix operator distributes over arbitrary unions of languages. However, for the intersection of two languages L and H, we have $\operatorname{pre}(L \cap H) \subseteq (\operatorname{pre} L) \cap (\operatorname{pre} H)$. If equality holds, L and H are said to be *non-conflicting*.

The *natural projection* $\mathrm{p_o}\colon \Sigma^* \to \Sigma_\mathrm{o}^*$, $\Sigma_\mathrm{o} \subseteq \Sigma$, is defined iteratively: (1) let $\mathrm{p_o}\,\epsilon := \epsilon$; (2) for $s \in \Sigma^*$, $\sigma \in \Sigma$, let $\mathrm{p_o}(s\sigma) := (\mathrm{p_o}\, s)\sigma$ if $\sigma \in \Sigma_\mathrm{o}$, or, if $\sigma \notin \Sigma_\mathrm{o}$, let $\mathrm{p_o}(s\sigma) := \mathrm{p_o}\, s$. The set-valued inverse $\mathrm{p_o^{-1}}$ of $\mathrm{p_o}$ is defined by $\mathrm{p_o^{-1}}(r) := \{s \in \Sigma^* \mid \mathrm{p_o}(s) = r\}$ for $r \in \Sigma_\mathrm{o}^*$. When applied to languages, the projection distributes over unions, and the inverse projection distributes over unions and intersections. The prefix operator commutes with projection and inverse projection.

Given two languages L, $K \subseteq \Sigma^*$, and a set of *uncontrollable events* $\Sigma_\mathrm{uc} \subseteq \Sigma$, we say K is *controllable w.r.t. L*, if $(\operatorname{pre} K)\Sigma_\mathrm{uc} \cap (\operatorname{pre} L) \subseteq \operatorname{pre} K$. A language $K \subseteq \Sigma^*$ is *complete*, if for all $s \in \operatorname{pre} K$ there exists $\sigma \in \Sigma$ such that $s\sigma \in \operatorname{pre} K$; see e.g. [3]. Each one of the properties controllability, completeness, closedness and relative closedness is retained under arbitrary union; see, e.g., [8, 9] regarding controllability, and [3] for completeness. Note that closedness and relative closedness is also retained under arbitrary intersection.

Unless otherwise noted, the alphabets Σ, Σ_c, Σ_uc, Σ_o and Σ_uo refer to the *common partitioning* $\Sigma = \Sigma_\mathrm{c} \dot\cup \Sigma_\mathrm{uc} = \Sigma_\mathrm{o} \dot\cup \Sigma_\mathrm{uo}$ in controllable, uncontrollable, observable and unobservable events, respectively.

An *automaton* is a tuple $G = (Q, \Sigma, \delta, q_\mathrm{o}, Q_\mathrm{m})$, with *state set Q*, *initial state* $q_\mathrm{o} \in Q$, *marked states* $Q_\mathrm{m} \subseteq Q$, and the *transition function* $\delta\colon Q \times \Sigma \to Q$ with its common extension to the domain $Q \times \Sigma^*$; i.e., $\delta(q, \epsilon) := q$ and $\delta(q, s\sigma) := \delta(\delta(q, s), \sigma)$. With the automaton G we associate the *marked language* $\mathrm{L_m}(G) := \{s \in \Sigma^* \mid \delta(q_\mathrm{o}, s) \in Q_\mathrm{m}\}$. A language $L \subseteq \Sigma^*$ is said to be *regular*, if it is marked by some automaton with finitely many states. For the automaton $G = (Q, \Sigma, \delta, q_o, Q_m)$, the equivalence relation $\equiv_G$ on Σ^* is defined by $s' \equiv_G s''$ if and only if $\delta(q_o, s') = \delta(q_o, s'')$. Given a language $L \subseteq \Sigma^*$, the equivalence relation $\equiv_L$ on Σ^* is defined by $s' \equiv_L s''$ if and only if $(\forall\ t \in \Sigma^*)[\, s't \in L \Leftrightarrow s''t \in L\,]$. If, for a language $M \subseteq \Sigma^*$, the equivalence relation $\equiv_G$ is at least as fine as $\equiv_M$, then M is marked by the generator $H = (Q, \Sigma, \delta, q_o, X_m)$ with $X_m := \{\delta(q_o, s) |\, s \in M\}$.

II. Problem Statement

For the purpose of this paper, both, the *plant* and the *controller* are modelled as formal languages $L \subseteq \Sigma^*$ and $H \subseteq \Sigma^*$, respectively, with the intersection $K = L \cap H$ as the *closed-loop behaviour*. Moreover, we impose the following conditions on the controller H w.r.t. the plant L and thereby characterise the control problem under consideration.

Definition II.1. Given a *plant* $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc}$, the *controller* $H \subseteq \Sigma^*$ is *admissible w.r.t.* L, if

(H0) H is prefix-closed;

(H1) H is controllable w.r.t. L;

(H2) L and H are non-conflicting; and,

(H3) $(\mathrm{pre}\, L) \cap (\mathrm{pre}\, H)$ is complete. □

It is readily verified that a closed-loop behaviour $K \subseteq L$ can be achieved by a controller H that complies with (H0)–(H2) if and only if K is controllable w.r.t. L and relatively closed w.r.t. L. This corresponds to *non-blocking supervision* as originally proposed by [8, 9]. There, control is exercised by a causal feedback map $V\colon\ \mathrm{pre}\, L \to \Gamma$, which maps the respective past string $s \in \mathrm{pre}\, L$ to a control pattern $\gamma = V(s)$, $\Sigma_{uc} \subseteq \gamma \subseteq \Sigma$, to indicate the set of enabled successor events. In this paper, the controller H is interpreted as a representation of the feedback map V.

The additional condition (H3) is motivated by *sequential behaviours*, i.e., discrete-event systems that continue operation for infinite logic time; see e.g. [3, 7]. To this end, we note that a language $K \subseteq L$ can be achieved as a closed-loop behaviour $K = L \cap H$ with a controller H that complies with (H0)–(H3) if and only if K is controllable w.r.t. L, relatively closed w.r.t. L, and complete. If the latter conditions on K are satisfied, $H := \mathrm{pre}\, K$ is an admissible controller that achieves the closed-loop behaviour K. When a *language inclusion specification* $E \subseteq L$ is given, controller design amounts to the computation of the supremal achievable closed-loop behaviour $K^\uparrow \subseteq E$ in order to extract a corresponding controller $H^\uparrow := \mathrm{pre}\, K^\uparrow$; see also [3, 6].

For the abstraction-based controller design discussed in this paper, the events synchronised with the controller are deliberately restricted to a set of *high-level events* $\Sigma_o \subseteq \Sigma$ and we utilise the projection $L_o := \mathrm{p_o}\, L \subseteq \Sigma_o^*$ as a *plant abstraction*. This setting is motivated by hierarchical control systems, where the plant consists of multiple components, each subject to low-level control, and the remaining task is to design a high-level controller $H_o \subseteq \Sigma_o^*$, that addresses cooperative behaviour expressed by a language inclusion specification $E_o \subseteq \Sigma_o^*$; see e.g. [2, 11, 13]. The effect of a high-level controller H_o on the plant L is represented by the *implementation* $H := \mathrm{p}_o^{-1} H_o$ and we obtain the closed-loop behaviour $K = L \cap H$ to observe

$$K \subseteq \mathrm{p}_o^{-1}(L_o \cap H_o) \quad \text{and} \quad L_o \cap H_o = \mathrm{p_o}\, K. \qquad (1)$$

The latter equality is also referred to as *hierarchical consistency*, and, for the particular closed-loop configuration under investigation, indeed holds automatically; see also [11, 14]. Moreover, the closed-loop behaviour satisfies the specification $E := \mathrm{p}_o^{-1} E_o$, i.e., $K \subseteq E$.

In the worst case, the number of states required to realise L_o is even larger when compared to L, so there may be no computational benefits; see [12]. However, for relevant applications a substantial reduction in the state count can be observed. In such a prospective situation, there remains a crucial question: *does admissibility of H_o w.r.t. the abstraction L_o imply admissibility of the implementation $H := \mathrm{p}_o^{-1} H_o$ w.r.t. the actual plant L?* We rephrase this question as a formal requirement imposed on the abstraction.

Definition II.2. Given a plant $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$, the plant abstraction $L_o := \mathrm{p_o}\, L$ is *consistent for the purpose of controller design* (or short *consistent*), if

$$(\forall\ H_o \subseteq \Sigma_o^*)[\, H_o \text{ is admissible w.r.t. } L_o \Rightarrow \mathrm{p}_o^{-1} H_o \text{ is admissible w.r.t. } L \quad \Box$$

This paper is concerned with conditions under which an abstraction is consistent for the purpose of controller design.

III. A Known Solution

We start our discussion with the observation that the implementation $H := \mathrm{p}_o^{-1} H_o$ in an abstraction-based design is automatically closed (H0) and controllable (H1).

Proposition III.1. Given a plant $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$, let $H_o \subseteq \Sigma_o^*$ be prefix-closed and controllable w.r.t. the plant abstraction $L_o := \mathrm{p_o}\, L$. Then $H := \mathrm{p}_o^{-1} H_o$ is closed and controllable w.r.t. L.

Proof. For closedness, recall that $\mathrm{pre}\, \mathrm{p}_o^{-1} H_o = \mathrm{p}_o^{-1} \mathrm{pre}\, H_o$. Controllability is covered by [14], Theorem 4.1. We provide a simple direct proof for the more specific case at hand. Pick any $s \in \mathrm{pre}\, H = H$ and $\sigma \in \Sigma_{uc}$, such that $s\sigma \in \mathrm{pre}\, L$. In the case of $\sigma \in \Sigma_{uo}$, we obtain $\mathrm{p_o}(s\sigma) = \mathrm{p_o}\, s \in \mathrm{p_o}\, H$ and, thus, $s\sigma \in \mathrm{p}_o^{-1}\, \mathrm{p_o}\, H = \mathrm{p}_o^{-1} H_o = H$. In the case of $\sigma \in \Sigma_o$, we obtain $(\mathrm{p_o}\, s)\sigma = \mathrm{p_o}(s\sigma) \in \mathrm{pre}\, L_o$, and, $\mathrm{p_o}\, s \in H_o$. By controllability of H_o w.r.t. L_o, observe that $\mathrm{p_o}(s\sigma) \in \mathrm{pre}\, H_o = H_o$. Again, we can conclude $s\sigma \in \mathrm{p}_o^{-1} H_o = H$. □

However, the liveness properties (H2) and (H3) are not automatically satisfied and we refer to literature on *non-blocking hierarchical supervisory control*, where a variety of *observer conditions* [13] have been developed for this purpose; see also [2, 10, 11]. For the particular setting of the present paper, we show that *natural observers* provide consistent abstractions in the sense of Definition II.2, and thereby relate our study to the more general framework [13].

Definition III.2. The projection $\mathrm{p_o}\colon \Sigma^* \to \Sigma_\mathrm{o}^*$ is a *natural observer* for a language $L \subseteq \Sigma^*$, if for all $s \in \operatorname{pre} L$ and all $u \in \Sigma_\mathrm{o}^*$ with $(\mathrm{p_o}\, s)u \in \mathrm{p_o}\, L$ there exists $t \in \Sigma^*$ such that $st \in L$ and $\mathrm{p_o}\, t = u$. □

For natural observers, admissibility of H_o w.r.t. L_o indeed implies non-conflictingness (H2) and completeness (H3) for the actual closed-loop system.

Proposition III.3. Given a natural observer $\mathrm{p_o}\colon \Sigma^* \to \Sigma_\mathrm{o}^*$ for the plant $L \subseteq \Sigma^*$, consider a prefix-closed controller candidate $H_\mathrm{o} \subseteq \Sigma_\mathrm{o}^*$. If $L_\mathrm{o} := \mathrm{p_o}\, L$ and H_o are non-conflicting, then so are L and $H := \mathrm{p_o^{-1}}\, H_\mathrm{o}$.
Proof. The claim is covered by [13], Theorem 6. We provide a simple direct proof for the more specific case at hand. Pick any $s \in (\operatorname{pre} L) \cap H$. Clearly, $\mathrm{p_o}\, s \in (\operatorname{pre} L_\mathrm{o}) \cap (\mathrm{p_o}\, \mathrm{p_o^{-1}}\, H_\mathrm{o}) = (\operatorname{pre} L_\mathrm{o}) \cap H_\mathrm{o}$. Since L_o and H_o are non-conflicting, we can choose $u \subseteq \Sigma_\mathrm{o}^*$ such that $(\mathrm{p_o}\, s)u \in L_\mathrm{o} \cap H_\mathrm{o}$. By the natural observer property, there exists $t \in \Sigma^*$ with $st \in L$, $\mathrm{p_o}\, t = u$. The latter implies $st \in \mathrm{p_o^{-1}}\, H_\mathrm{o}$, and, thus, $st \in L \cap H$. □

Proposition III.4. Given a natural observer $\mathrm{p_o}\colon \Sigma^* \to \Sigma_\mathrm{o}^*$ for the plant $L \subseteq \Sigma^*$, consider a prefix-closed controller candidate $H_\mathrm{o} \subseteq \Sigma_\mathrm{o}^*$, and let $H := \mathrm{p_o^{-1}}\, H_\mathrm{o}$. If $L_\mathrm{o} := \mathrm{p_o}\, L$ and H_o are non-conflicting and if $(\operatorname{pre} L_\mathrm{o}) \cap H_\mathrm{o}$ is complete, then $(\operatorname{pre} L) \cap H$ is complete, too.
Proof. Pick any $s \in (\operatorname{pre} L) \cap H$ and observe $\mathrm{p_o}\, s \in (\operatorname{pre} L_\mathrm{o}) \cap H_\mathrm{o}$. By completeness of $(\operatorname{pre} L_\mathrm{o}) \cap H_\mathrm{o}$, we can choose $\sigma \in \Sigma_\mathrm{o}$ such that $(\mathrm{p_o}\, s)\sigma \in (\operatorname{pre} L_\mathrm{o}) \cap H_\mathrm{o}$. As in the proof of the above Proposition III.3, we choose $u \subseteq \Sigma_\mathrm{o}^*$ such that $(\mathrm{p_o}\, s)\sigma u \in L_\mathrm{o} \cap H_\mathrm{o}$, to obtain, by the natural observer property, the existence of $t \in \mathrm{p_o^{-1}}\, u$ such that $s\sigma t \in L \cap H$. □

The above propositions are summarised by the following theorem to identify the natural observer property as a sufficient prerequisite for abstraction-based controller design.

Theorem III.5. If the projection $\mathrm{p_o}\colon \Sigma^* \to \Sigma_\mathrm{o}^*$ is a natural observer for the plant $L \subseteq \Sigma^*$, then the abstraction $L_\mathrm{o} := \mathrm{p_o}\, L$ is consistent for the purpose of controller design. □

Natural observers not only guarantee liveness for the closed loop, but also exhibit favourable properties regarding composed plant models (can be verified componentwise) and the state count of the abstraction (state count is not increased); see e.g. [2, 11]. Moreover, the observer property can be synthesised in polynomial time by extending the alphabet of observabe events; see e.g. [1]. Thus, natural observers and variations thereof [10, 11] can be conceived the preferable approach to abstraction-based controller synthesis.

However, there are situations where no natural observer with a reasonable state count in the abstraction exists, and we demonstrate this by the example Fig. 1, which has been extracted from an real-world application. Here the events x, y, p and a represent alternative work cycles of a processing machine. The event f represents the feed of a workpiece performed by a transport system with buffer capacity two. For the design of a high-level controller to operate the process, a specification is stated in terms of $\Sigma_\mathrm{o} = \{\mathsf{x}, \mathsf{y}, \mathsf{p}, \mathsf{a}\}$, i.e., the only event that one may remove by projection is f. The right-hand side in Fig. 1 shows the resulting abstraction. The example is readily extended for larger buffer capacities, with a constant state count in the abstraction and substantially higher state count in the detailed plant model. To observe that the projection fails to be a natural observer, consider the low-level string $s = \mathsf{ffx}$. Here, the abstraction suggests that $u = \mathsf{pa}$ reaches a marked state, but, for the low-level marking to be reached, two a events are required. As the only option to extend the set of high-level events is to include f, natural observers can not reduce the state count for this example.

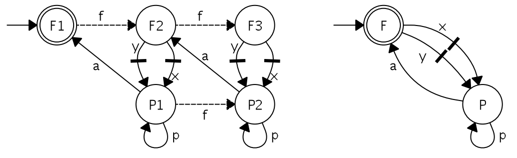

Fig. 1. Plant L and abstraction L_o, resp., with $\Sigma_\mathrm{o} = \{\mathsf{x}, \mathsf{y}, \mathsf{p}, \mathsf{a}, \}$, $\Sigma_\mathrm{c} = \{\mathsf{x}, \mathsf{y}\}$

Even though the proposed projection fails to be a natural observer, any admissible high-level controller H_o must enable at least one of the controllable events x and y for completeness of $(\operatorname{pre} L_\mathrm{o}) \cap H_\mathrm{o}$. This implies that the implementation H can never prevent the actual closed loop to reach a marking. Likewise, completeness is verified by inspection. Thus, we found a consistent abstraction with reduced state count. This demonstrates that, in the context of the above theorem, the natural observer property is restrictive.

IV. Backward Reachability Analysis

The liveness properties non-conflictingness (H2) and completeness (H3) require that any string in the *local closed-loop behaviour* $(\operatorname{pre} L) \cap H$ can be extended by another event, and that a finite number of such extensions can reach a string in $L \cap H$. Thus, when L and H are given, both properties can be verified by a backward-reachability analysis, with target set $M = (\operatorname{pre} L) \cap H$ or $M = L \cap H$, respectively. For the purpose of consistency, L is interpreted as known parameter, whilst H is universally quantified to be any abstraction-based controller. We give a formal definition of the reachability operators under consideration.

Definition IV.1. Given a plant $L \subseteq \Sigma^*$, $\Sigma = \Sigma_\mathrm{c} \dot\cup \Sigma_\mathrm{uc} = \Sigma_\mathrm{o} \dot\cup \Sigma_\mathrm{uo}$, denote $L_\mathrm{o} := \mathrm{p_o}\, L$ the plant abstraction. An operator Ω_1 on $\operatorname{pre} L$, i.e., $\Omega_1(M) \subseteq \operatorname{pre} L$ for $M \subseteq \operatorname{pre} L$, is a *universal one-step-reachability operator*, if, for all $M \subseteq \operatorname{pre} L$ and all controllers $H_\mathrm{o} \subseteq \Sigma_\mathrm{o}^*$ admissible w.r.t. L_o, it holds that

$$(\forall\ s \in \Omega_1(M) \cap \mathrm{p_o^{-1}}\, H_\mathrm{o})(\exists\, \sigma \in \Sigma)[\, s\sigma \in M \cap \mathrm{p_o^{-1}}\, H_\mathrm{o}\,]. \quad (2)$$

Likewise, Ω_* is a *universal star-reachability operator*, if

$$(\forall\ s \in \Omega_*(M) \cap \mathrm{p_o^{-1}}\, H_\mathrm{o})(\exists\, t \in \Sigma^*)[\, st \in M \cap \mathrm{p_o^{-1}}\, H_\mathrm{o}\,]. \quad (3)$$

□

Along with the intention of Definition IV.1, the following proposition states sufficient conditions for liveness properties in an abstraction-based controller design.

Proposition IV.2. Given a plant $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$, denote $L_o := p_o L$ the plant abstraction. Furthermore, let

$$H_o^{\uparrow} := \sup\{ H_o \subseteq \Sigma_o^* \,|\, H_o \text{ is admissible w.r.t. } L_o \}, \tag{4}$$
$$K_{loc}^{\uparrow} := (\mathrm{pre}\, L) \cap (p_o^{-1} H_o^{\uparrow}). \tag{5}$$

If, for a universal one-step-reachability operator Ω_1, we have

$$K_{loc}^{\uparrow} \subseteq \Omega_1(\mathrm{pre}\, L), \tag{6}$$

then the local closed loop $(\mathrm{pre}\, L) \cap (p_o^{-1} H_o)$ is complete for any controller H_o that is admissible w.r.t. L_o. If, for a universal star-reachability operator Ω_*, we have

$$K_{loc}^{\uparrow} \subseteq \Omega_*(L), \tag{7}$$

then L and $p_o^{-1} H_o$ are non-conflicting for any controller H_o that is admissible w.r.t. L_o.

Proof. Recall that properties (H0)–(H3) are retained under arbitrary union. Thus, $H_o^{\uparrow}$ is admissible w.r.t. L_o and constitutes an upper bound to all admissible controllers H_o. In particular, we have $(\mathrm{pre}\, L) \cap (p_o^{-1} H_o) \subseteq (\mathrm{pre}\, L) \cap (p_o^{-1} H_o^{\uparrow}) = K_{loc}^{\uparrow}$. Now assume that (6) is satisfied for Ω_1, and pick an arbitrary $s \in (\mathrm{pre}\, L) \cap (p_o^{-1} H_o)$. In particular, we have $s \in K_{loc}^{\uparrow} \subseteq \Omega_1(\mathrm{pre}\, L) \cap (p_o^{-1} H_o)$. Thus, Definition IV.1 guarantees the existence of $\sigma \in \Sigma$ such that $s\sigma \in \mathrm{pre}\, L \cap (p_o^{-1} H_o)$; i.e, the local closed-loop behaviour is complete. Regarding non-conflictingness, assume that (7) is satisfied for Ω_*, and pick an arbitrary $s \in (\mathrm{pre}\, L) \cap (p_o^{-1} H_o)$. In particular, we have $s \in K_{loc}^{\uparrow} \subseteq \Omega_*(L) \cap (p_o^{-1} H_o)$. Thus, Definition IV.1 guarantees the existence of $t \in \Sigma^*$ such that $st \in L \cap (p_o^{-1} H_o)$; i.e., L and $p_o^{-1} H_o$ are non-conflicting. □

In order to obtain specific universal one-step-reachability operators, we identify strings $s \in \mathrm{pre}\, L$ with an extension $s\sigma \in M \subseteq \mathrm{pre}\, L$ that can not be prevented by the implementation $p_o^{-1} H_o$ of any admissible controller H_o. This is clearly the case when the extension can be chosen as an unobservable event, i.e., $\sigma \in \Sigma_{uo}$, $s\sigma \in M \subseteq \mathrm{pre}\, L$. Referring to controllability of $p_o^{-1} H_o$ (by Proposition III.1), this is also the case for uncontrollable extensions, $\sigma \in \Sigma_{uc}$. Finally, referring to completeness of $(\mathrm{pre}\, L_o) \cap H_o$, we identify the case where all observable successor events that comply with $\mathrm{pre}\, L_o$ reach the target M. The following proposition gives a formal definition of the respective operator for each case.

Proposition IV.3. Given a plant $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$, denote $L_o := p_o L$ the plant abstraction. Then the operators Ω_{1A}, Ω_{1B} and Ω_{1C}, defined by

$$\Omega_{1A}(M) := \{ s \in \mathrm{pre}\, L \,|\, \exists\, \sigma \in \Sigma_{uo} : s\sigma \in M \}, \tag{8}$$
$$\Omega_{1B}(M) := \{ s \in \mathrm{pre}\, L \,|\, \exists\, \sigma \in \Sigma_{uc} : s\sigma \in M \}, \tag{9}$$
$$\Omega_{1C}(M) := \{ s \in \mathrm{pre}\, L \,| \;\; (\forall\, \sigma \in \Sigma_o)[\, p_o\, s\sigma \in \mathrm{pre}\, L_o \;\Rightarrow\; s\sigma \in M \,] \}, \tag{10}$$

for $M \subseteq \mathrm{pre}\, L$, are universal one-step-reachability operators.

Proof. Choose $M \subseteq \mathrm{pre}\, L$ and a controller $H_o \subseteq \Sigma_o^*$ that is admissible w.r.t. L_o, both arbitrarily. For $s \in \Omega_{1A}(M) \cap (p_o^{-1} H_o)$, we can choose $\sigma \in \Sigma_{uo}$, $s\sigma \in M$, and $p_o(s\sigma) = p_o\, s$ implies $s\sigma \in p_o^{-1} H_o$. For $s \in \Omega_{1B}(M) \cap (p_o^{-1} H_o)$, we can choose $\sigma \in \Sigma_{uc}$, $s\sigma \in M$. By Proposition III.1, $p_o^{-1} H_o$ is controllable w.r.t. L and, thus, $s\sigma \in M \subseteq \mathrm{pre}\, L$ implies $s\sigma \in p_o^{-1} H_o$. Finally, consider $s \in \Omega_{1C}(M) \cap (p_o^{-1} H_o)$. Here, we have $p_o\, s \in (\mathrm{pre}\, L_o) \cap H_o$ and, by admissibility of H_o w.r.t. L_o, we can choose $\sigma \in \Sigma_o$ such that $p_o\, s\sigma \in (\mathrm{pre}\, L_o) \cap H_o$. By the definition of Ω_{1C}, this implies $s\sigma \in M \cap (p_o^{-1} H_o)$. □

It follows immediately from Definition IV.1, that the union composition of (arbitrarily many) universal reachability operators again is a universal reachability operator. In particular,

$$\Omega_{1ABC} := \Omega_{1A} \cup \Omega_{1B} \cup \Omega_{1C} \tag{11}$$

is a universal one-step-reachability operator. For the example Fig. 1, we obtain $\mathrm{pre}\, L = \Omega_{1ABC}(\mathrm{pre}\, L)$ and thereby satisfy condition (6) in Proposition IV.2.

Technically, any universal one-step-reachability operator is also a universal star-reachability operator. Since the identity $\Omega_{id}(M) := M$ constitutes a universal star-reachability operator, so does

$$\Omega_{*ABC} := \Omega_{id} \cup \Omega_{1A} \cup \Omega_{1B} \cup \Omega_{1C}\,. \tag{12}$$

The following proposition establishes how iterations of a universal star-reachability operator again yield a universal star-reachability operator.

Proposition IV.4. Given a plant $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$, denote $L_o := p_o L$ the abstraction, and consider a universal star-reachability operator Ω_*. For $M \subseteq \mathrm{pre}\, L$ and $n \in \mathbb{N}$ let $M_0 := M$, $M_{i+1} := \Omega_*(M_i)$, and define Ω_*^n and Ω_*^∞ by

$$\Omega_*^n(M) := M_n \quad\text{and}\quad \Omega_*^\infty(M) := \cup_{i\in\mathbb{N}} M_i\,, \tag{13}$$

respectively. Then Ω_*^n and Ω_*^∞ are universal star-reachability operators.

Proof. Let $M \subseteq \mathrm{pre}\, L$ and $n \in \mathbb{N}$ and choose any controller $H_o \subseteq \Sigma_o^*$ that is admissible w.r.t. L_o. Pick an arbitrary $s \in \Omega_*^n(M) \cap (p_o^{-1} H_o)$. For the case of $n = 0$, we have $st \in M \cap (p_o^{-1} H_o)$ with $t = \epsilon$. If, for some $i \in \mathbb{N}$, we have $s \in M_{i+1} \cap (p_o^{-1} H_o)$, we can choose $t_i \in \Sigma^*$ such that $st_i \in M_i \cap (p_o^{-1} H_o)$. For the case of $n > 0$, we have $s \in M_n \cap (p_o^{-1} H_o)$ and we apply the latter argument n times to obtain $st_n t_{n-1} \cdots t_2 t_1 \in M_0 \cap (p_o^{-1} H_o)$. Thus, Ω_*^n is a universal star-reachability operator. By definition, Ω_*^∞ is the union of universal star-reachability operators. Thus, Ω_*^∞ itself is a universal star-reachability operator. □

Indeed, for the example Fig. 1, iterating Ω_{*ABC} on L yields $\mathrm{pre}\, L$ and, thus, establishes condition (7). We conclude that the respective abstraction is consistent. However, the variation Fig. 2 of our example demonstrates limitations of the operators proposed so far. In Fig. 2, the process starts automatically r and can be completed by the controller with either a or b. Depending on the specification, controllers may do so after a particular number of p events. Universal quantification over all controllers corresponds to an unbounded set of extensions required to reach a marking, once the process has been started. This is not accounted for when iterating Ω_{*ABC}, and we propose an alternative operator.

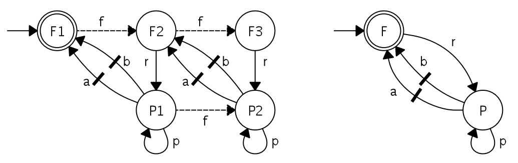

Fig. 2. Plant L and abstraction L_o, resp., with $\Sigma_o = \{\mathsf{r}, \mathsf{p}, \mathsf{a}, \mathsf{b}, \}$, $\Sigma_c = \{\mathsf{a}, \mathsf{b}\}$

Proposition IV.5. Given a plant $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$, denote $L_o := \mathrm{p_o}\, L$ the plant abstraction. Consider the operator Ω_{*D}, defined by

$$\Omega_{*D}(M) := \{\, s \in \operatorname{pre} L \,| \\ (\forall\ u \in \Sigma_o^*)[\, \mathrm{p_o}\, su \in L_o \ \Rightarrow\ (s\, \mathrm{p_o^{-1}} \operatorname{pre} u) \cap M \neq \emptyset \,]\,\}, \quad (14)$$

for $M \subseteq \operatorname{pre} L$. Then Ω_{*D} is a universal star-reachability operator.

Proof. Choose any $M \subseteq \operatorname{pre} L$ and any controller $H_o \subseteq \Sigma_o^*$ that is admissible w.r.t. L_o. For $s \in \Omega_{*D}(M) \cap (\mathrm{p_o^{-1}}\, H_o)$, we have $\mathrm{p_o}\, s \in (\operatorname{pre} L_o) \cap H_o$ and, by admissibility of H_o w.r.t. L_o, we can choose $u \in \Sigma_o^*$ such that $\mathrm{p_o}\, su \in L_o \cap H_o$. Observe that $s\, \mathrm{p_o^{-1}} \operatorname{pre} u \subseteq \mathrm{p_o^{-1}}\, H_o$. Moreover, by the definition of Ω_{*D}, we can choose $t \in \mathrm{p_o^{-1}} \operatorname{pre} u$, such that $st \in M$, and, hence, $st \in M \cap \mathrm{p_o^{-1}}\, H_o$. □

Observe that $M \subseteq \Omega_{*D}(M)$ for all $M \subseteq \operatorname{pre} L$. Thus, we define Ω_{*ABCD} by

$$\Omega_{*ABCD} := \Omega_{1A} \cup \Omega_{1B} \cup \Omega_{1C} \cup \Omega_{*D}\,, \quad (15)$$

and obtain, by Propositions IV.4 and IV.5, the universal star-reachability operator

$$\Omega^{\infty}_{*ABCD} := \cup_{i \in \mathbb{N}} \Omega^{i}_{*ABCD}\,. \quad (16)$$

Indeed, for both examples Fig. 1 and Fig. 2, the iteration yields $\operatorname{pre} L = \Omega^{\infty}_{*ABCD}(L)$, and, thus, consistency is established for the respective abstractions. The following theorem summarises Propositions IV.2–IV.5 as our main result.

Theorem IV.6. Given a plant $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$, denote $L_o := \mathrm{p_o}\, L$ the plant abstraction, and $K^{\uparrow}_{\mathrm{loc}} \subseteq \Sigma^*$ the upper bound on the local closed loop from Eq. (5). If

$$K^{\uparrow}_{\mathrm{loc}} \subseteq \Omega_{1ABC}(\operatorname{pre} L) \cap \Omega^{\infty}_{*ABCD}(L)\,, \quad (17)$$

for the operators Ω_{1ABC} and Ω^{∞}_{*ABCD} from Eq. (11) and Eq. (16), then the abstraction L_o is consistent for the purpose of controller design. □

Both examples Fig. 1 and Fig. 2 demonstrate, that our condition (17) can be satisfied even when $\mathrm{p_o}$ fails to be a natural observer. In general, our condition is not more restrictive than the requirement of $\mathrm{p_o}$ to be a natural observer.

Theorem IV.7. If the projection $\mathrm{p_o}\colon \Sigma^* \to \Sigma_o^*$ is a natural observer for the plant $L \subseteq \Sigma^*$, then condition (17) in Theorem IV.6 is satisfied.

Proof. To show that $K^{\uparrow}_{\mathrm{loc}} \subseteq \Omega_{1ABC}(\operatorname{pre} L)$, pick an arbitrary $s \in K^{\uparrow}_{\mathrm{loc}} \subseteq \operatorname{pre} L$. We distinguish two cases. First, if there exists $\sigma \in \Sigma_{uo} \cup \Sigma_{uc}$ with $s\sigma \in \operatorname{pre} L$, we obtain $s \in \Omega_{1A}(\operatorname{pre} L) \cup \Omega_{1B}(\operatorname{pre} L)$, and, hence $s \in \Omega_{1ABC}(\operatorname{pre} L)$. For the second case, we have that $s\sigma \in \operatorname{pre} L$ implies $\sigma \in \Sigma_o$. Here, we establish that $s \in \Omega_{1C}(\operatorname{pre} L)$. Pick an arbitrary $\rho \in \Sigma_o$ such that $\mathrm{p_o}\, s\rho \in \operatorname{pre} L_o$. Then there exists $u \in \Sigma_o^*$ with $\mathrm{p_o}\, s\rho u \in L_o$ and we refer to the natural observer property to obtain $t \in \mathrm{p_o^{-1}}(\rho u)$ such that $st \in L$. In particular, the first event in t must be observable, and, hence, match ρ. This implies $s\rho \in \operatorname{pre} L$, and, hence, $s \in \Omega_{1C}(\operatorname{pre} L)$. With both cases, we conclude $K^{\uparrow}_{\mathrm{loc}} \subseteq \Omega_{1ABC}(\operatorname{pre} L)$. We now show that $K^{\uparrow}_{\mathrm{loc}} \subseteq \Omega_{*D}(L)$. Pick an arbitrary $s \in K^{\uparrow}_{\mathrm{loc}} \subseteq \operatorname{pre} L$. For any $u \in \Sigma_o^*$ such that $\mathrm{p_o}\, su \in L_o$, we refer to the natural observer property in order to choose $t \in \mathrm{p_o^{-1}}\, u$ such that $st \in L$. In particular, $(s\, \mathrm{p_o^{-1}}\, u) \cap L \neq \emptyset$, and, hence, $s \in \Omega_{*D}(L)$. Clearly, $\Omega_{*D}(L) \subseteq \Omega^{\infty}_{*ABCD}(L)$, and we obtain $K^{\uparrow}_{\mathrm{loc}} \subseteq \Omega^{\infty}_{*ABCD}(L)$. □

V. Implementation

Under the assumption that the plant L and the target set M are regular with known finite automata realisations, we outline a possible implementation of the reachability operators from the previous section. In our discussion, we establish that each operator retains regularity and identify a state set suitable for the realisation of the respective results.

Proposition V.1. Given a plant $L \subseteq \Sigma^*$, denote the plant abstraction $L_o := \mathrm{p_o}\, L$. Then, for any target set $M \subseteq \operatorname{pre} L$ and any s', $s'' \in \Sigma^*$, we have

$$s' \equiv_L s'',\ s' \equiv_{\mathrm{p_o^{-1}} L_o} s'' \text{ and } s' \equiv_M s'' \quad \Rightarrow \quad s' \equiv_{\Omega(M)} s''\,,$$

where Ω denotes either one of the operators Ω_{1A}, Ω_{1B}, Ω_{1C} and Ω_{*D}, defined by Propositions IV.3 and IV.5, respectively.

Proof. For each of the operators under consideration, we have to show that for all s', $s'' \in \Sigma^*$ which are equivalent w.r.t. $\equiv_L$, $\equiv_{\mathrm{p_o^{-1}} L_o}$ and $\equiv_M$, and for all $t \in \Sigma^*$, we have

$$s't \in \Omega(M) \ \Leftrightarrow\ s''t \in \Omega(M)\,.$$

Note that, $s' \equiv_L s''$ implies $s' \equiv_{\operatorname{pre} L} s''$, and, likewise, we obtain $s' \equiv_{\operatorname{pre} \mathrm{p_o^{-1}} L_o} s''$. Thus, for the two operators Ω_{1A} and Ω_{1B}, the claim follows immediately from

$$st \in \Omega_{1A}(M) \ \Leftrightarrow\ (\, st \in \operatorname{pre} L \text{ and } (\,\exists\, \sigma \in \Sigma_{uo})[\, st\sigma \in M \,]\,)\,,$$
$$st \in \Omega_{1B}(M) \ \Leftrightarrow\ (\, st \in \operatorname{pre} L \text{ and } (\,\exists\, \sigma \in \Sigma_{uc})[\, st\sigma \in M \,]\,)\,.$$

We turn to Ω_{1C} and use that, for all $s \in \Sigma^*$ and all $\sigma \in \Sigma_o$,

$$\mathrm{p_o}\, s\sigma \in \operatorname{pre} L_o \ \Leftrightarrow\ s\sigma \in \operatorname{pre} \mathrm{p_o^{-1}}\, L_o$$

to obtain

$$st \in \Omega_{1C}(M) \quad \Leftrightarrow \quad (\ st \in \operatorname{pre} L \ \text{ and} \\ (\,\forall\ \sigma \in \Sigma_o)[\, st\sigma \in \operatorname{pre} \mathrm{p_o^{-1}}\, L_o \ \Rightarrow\ st\sigma \in M \,]\ )\,.$$

This implies that $s't \in \Omega_{1C}(M)$ if and only if $s''t \in \Omega_{1C}(M)$. Regarding Ω_{*D}, we use that, for all $s \in \Sigma^*$ and all $u \in \Sigma_o^*$,

$$(\, \mathrm{p_o}\, su \in L_o \ \Rightarrow\ (s\, \mathrm{p_o^{-1}} \operatorname{pre} u) \cap M \neq \emptyset \,) \\ \Leftrightarrow \quad (\,\exists\, v \in \operatorname{pre} \mathrm{p_o^{-1}}\, u\,)[\, su \in \mathrm{p_o^{-1}}\, L_o \ \Rightarrow\ sv \in M \,]\,,$$

to obtain for all $t \in \Sigma^*$ that

$$st \in \Omega_{*D}(M) \quad \Leftrightarrow \quad (\ st \in \operatorname{pre} L \ \text{ and} \\ (\,\forall\ u \in \Sigma_o^*\,)(\,\exists\, v \in \operatorname{pre} \mathrm{p_o^{-1}}\, u\,)[\, stu \in \mathrm{p_o^{-1}}\, L_o \ \Rightarrow\ stv \in M \,]\ )\,.$$

Hence, $s't \in \Omega_{*D}(M)$ if and only if $s''t \in \Omega_{*D}(M)$. □

Given finite automata realisations of L and M, well-known procedures for the product composition, the projection and the inverse projection can be used to set up a finite automaton G with associated equivalence relation $\equiv_G$ on Σ^* that is at least as fine as the equivalences associated with the languages L, $\mathrm{p}_\mathrm{o}^{-1} L_\mathrm{o}$ and M; i.e., for all s', $s'' \in \Sigma^*$ we may assume that

$$s' \equiv_G s'' \quad \Rightarrow \quad s' \equiv_L s'' \,,\, s' \equiv_{\mathrm{p}_\mathrm{o}^{-1} L_\mathrm{o}} s'' \text{ and } s' \equiv_M s'' \,.$$

Then, each of the languages L, $\mathrm{pre}\, L$, $\mathrm{p}_\mathrm{o}^{-1} L_\mathrm{o}$, $\mathrm{pre}\, \mathrm{p}_\mathrm{o}^{-1} L_\mathrm{o}$ and M can be represented by an automaton H which matches G except for the set of marked states; i.e., each of the above languages can be effectively represented by a set of marked states and the transition relation of G.

By the above proposition, the same is true for the resulting languages $\Omega_{1\mathrm{A}}(M)$, $\Omega_{1\mathrm{B}}(M)$, $\Omega_{1\mathrm{C}}(M)$ and $\Omega_{*\mathrm{D}}(M)$, and an implementation of either operator can be obtained by identifying the respective set of marked states. For $\Omega_{1\mathrm{A}}(M)$ and $\Omega_{1\mathrm{B}}(M)$, this can be done by traversing the state set of G and testing for the existence of a transition with an event from Σ_uo and Σ_uc, respectively, to a successor state that is marked for M. Regarding $\Omega_{1\mathrm{C}}(M)$, we obtain the corresponding set of marked states by testing whether all observable events enabled for $\mathrm{pre}\, \mathrm{p}_\mathrm{o}^{-1} L_\mathrm{o}$ have a successor state that is marked for M. Regarding $\Omega_{*\mathrm{D}}(M)$, we refer to the following characterisation:

$$\begin{aligned} &(\forall\ u \in \Sigma_\mathrm{o}^*)[\, \mathrm{p}_\mathrm{o}\, su \in L_\mathrm{o} \ \Rightarrow\ (s\, \mathrm{p}_\mathrm{o}^{-1}\, \mathrm{pre}\, u) \cap M \neq \emptyset \,] \\ \Leftrightarrow\quad & L_\mathrm{o} \cap (\mathrm{p}_\mathrm{o}\, s\Sigma_\mathrm{o}^*) \ \subseteq\ L_\mathrm{o} \cap \mathrm{p}_\mathrm{o}(\,(s\Sigma^*) \cap M\,)\Sigma_\mathrm{o}^* \,. \end{aligned}$$

Applying the inverse projection on both sides, each term can be evaluated referring to a copy G and represented by a set of marked states. Note that the defining condition of $\Omega_{*\mathrm{D}}(M)$ is a variation of the observer property for a restriction of G. Thus, the methodology proposed in [1] is expected to be applicable to obtain a more efficient verification procedure.

All described procedures are of polynomial order in the state count of G. Thus, provided that the projection at hand does reduce the state count in the abstraction, the operators can be evaluated efficiently. If, on the other hand, the projection at hand does not reduce the state count, the operators are of limited interest anyway.

Implementations of $\Omega_{*\mathrm{ABC}}^\infty(M)$ and $\Omega_{*\mathrm{ABCD}}^\infty(M)$ as iterations of $\Omega_{*\mathrm{ABC}}$ and $\Omega_{*\mathrm{ABCD}}$, respectively, can be terminated when a fixpoint is attained. Here, the following proposition guarantees finite termination.

Proposition V.2. Given a regular plant $L \subseteq \Sigma^*$, denote the plant abstraction $L_\mathrm{o} := \mathrm{p}_\mathrm{o}\, L$. Then, for any regular target set $M \subseteq \mathrm{pre}\, L$, the iterations $\Omega_{*\mathrm{ABC}}^\infty(M)$ and $\Omega_{*\mathrm{ABCD}}^\infty(M)$, defined by Proposition IV.4, reach a fixpoint after finitely many iterations.

Proof. Let Ω_* denote either one of the operators $\Omega_{*\mathrm{ABC}}^\infty$ or $\Omega_{*\mathrm{ABCD}}^\infty$. Recall that $N \subseteq \Omega_*(N)$ for all $N \subseteq \mathrm{pre}\, L$. Thus, the sequence $(M_i)_{i\in\mathbb{N}}$ defined by $M_0 := M$, $M_{i+1} := \Omega_*(M_i)$, is monotonously increasing and bounded by $\mathrm{pre}\, L$. Pick any s', $s'' \in \Sigma^*$ with $s' \equiv_L s''$, $s' \equiv_{\mathrm{p}_\mathrm{o}^{-1} L_\mathrm{o}} s''$ and $s' \equiv_M s''$. By Proposition V.1, we have $s' \equiv_{M_{i+1}} s''$ if $s' \equiv_{M_i} s''$. This implies $s' \equiv_{M_i} s''$ for all i. In particular, there is a finite upper bound for the minimal number of states required to realise any iterate M_i. Thus, the sequence $(M_i)_{i\in\mathbb{N}}$ takes only finitely many different values. By monotonicity, a fixpoint must be attained after a finite number of iterations. □

CONCLUSION

We have developed an alternative condition to guarantee that an abstraction-based controller design yields a valid solution for the actual plant. As with known conditions for this purpose, our result refers to the particular plant at hand and accounts for any subsequent abstraction-based controller design. In our study, we consider abstractions that are obtained by natural projections to a high-level alphabet. The control objectives under consideration include the common controllability condition as well as two liveness properties. The latter are addressed by a reachability analysis where the controller is interpreted as an unknown parameter. We demonstrate by example that our condition can be satisfied for projections that fail to be natural observers, and, we show that any natural observer satisfies our alternative condition.

REFERENCES

[1] L. Feng and W. M. Wonham. On the computation of natural observers in discrete-event systems. *Discrete Event Dynamic Systems*, 20:63–102, 2010.

[2] L. Feng and W.M. Wonham. Supervisory control architecture for discrete-event systems. *IEEE Transactions on Automatic Control*, 53(6):1449–1461, 2008.

[3] R. Kumar, V. Garg, and S. I. Marcus. On supervisory control of sequential behaviors. *IEEE Transactions on Automatic Control*, 37:1978–1985, 1992.

[4] R. Malik, H. Flordal, and P. Pena. Conflicts and projections. *1st IFAC Workshop on Dependable Control of Discrete Systems (DCDS)*, pages 63–68, 2007.

[5] S. Mohajerani, R. Malik, S. Ware, and M. Fabian. On the use of observation equivalence in synthesis abstraction. *3rd Int. Workshop on Dependable Control of Discrete Systems*, 2011.

[6] T. Moor, Ch. Baier, T.-S. Yoo, F. Lin, and S. Lafortune. On the computation of supremal sublanguages relevant to supervisory control. *Workshop on Discrete Event Systems (WODES)*, pages 175–180, 2012.

[7] P. J. Ramadge. Some tractable supervisory control problems for discrete-event systems modeled by büchi automata. *IEEE Transactions on Automatic Control*, 34:10–19, 1989.

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

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

[10] K. Schmidt and C. Breindl. Maximally permissive hierarchical control of decentralized discrete event systems. *IEEE Transactions on Automatic Control*, 56(4):723–737, 2011.

[11] K. Schmidt, T. Moor, and S. Perk. Nonblocking hierarchical control of decentralized discrete event systems. *IEEE Transactions on Automatic Control*, 53(10):2252–2265, 2008.

[12] K. C. Wong. On the complexity of projections of discrete-event systems. In *In IEE Workshop on Discrete Event Systems*, pages 201–208, 1998.

[13] K. C. Wong and W. M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems*, 6:241–306, 1996.

[14] H. Zhong and W. M. Wonham. On the consistency of hierarchical supervision in discrete-event systems. *IEEE Transactions on Automatic Control*, 35:1125–1134, 1990.