On the Computation of Supremal Sublanguages Relevant to Supervisory Control

Thomas Moor * **Christine Baier** * **Tae-Sic Yoo** ** **Feng Lin** ***
Stéphane Lafortune ****

* *Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg (e-mail: lrt@rt.eei.uni-erlangen.de)*
** *Idaho National Laboratory, Idaho Falls (e-mail: tae-sic.yoo@inl.gov)*
*** *Dept. of Electrical and Computer Engineering, Wayne State University, Detroit, and School of Electronics and Information Engineering, Tongji University, Shanghai (e-mail: flin@ece.eng.wayne.edu)*
**** *Dept. of Electrical Engineering and Computer Science, The University of Michigan (e-mail: stephane@eecs.umich.edu)*

Abstract: Given a specification language, this paper discusses an iterative procedure for the computation of the supremal sublanguage, that possesses a conjunction of certain closed-loop properties, including controllability, normality and completeness. The iteration is stated in terms of (i) supremal sublanguage operators for each individual property, (ii) prefix-closures, and, (iii) language intersections. Within the iteration, the individual supremal sublanguage operators are only applied on prefix-closed languages, while the overall specification is not required to be prefix-closed. Our main result establishes finite convergence, provided that all parameters are regular.

Keywords: discrete event systems, supervisory control, supremal sublanguages

1. INTRODUCTION

Within the context of supervisory control theory, as initially proposed by Ramadge and Wonham [1987, 1989] and since then intensively studied, supremal sublanguages that possess certain desired closed-loop properties play a key role: the respective properties are used to characterize the achievable closed-loop behaviour, and a maximally permissive supervisor can be stated in terms of the respective supremal sublanguage. Thus, for practical reasons, there is an interest in algorithms for the computation of the respective supremal sublanguage.

The closed-loop properties addressed in the literature include *controllability* [Ramadge and Wonham, 1987, 1989], *normality* [Brandt et al., 1990, Cho and Marcus, 1989] and *completeness* (also known as *liveness*) [Kumar et al., 1992]. However, not every combination is well studied. To the best of our knowledge, no procedure has been reported so far, that computes the supremal controllable *and* normal *and* complete sublanguage. The latter is relevant for the supervision of *sequential behaviours* under partial observation; see also Kumar et al. [1992].

When restricting attention to the case of prefix-closed specifications, the problem of computing the supremal sublanguage is expected to be less difficult. For example, Brandt et al. [1990] establish an elegant formula to compute the controllable and normal sublanguage of a prefix-closed specification. In contrast, the algorithm for the not necessarily prefix-closed case presented in [Cho and Marcus, 1989] is quite involved. This is one motivation for the iterative scheme by Yoo et al. [2002]. While addressing the not necessarily prefix-closed case, the proposed iteration is stated in terms of supremal sublanguages of prefix-closed specifications. Together with the formula from [Brandt et al., 1990], one obtains an overall procedure in terms of regular expressions and projections, which is straightforward to implement; see e.g. UMDES-LIB or libFAUDES.

In this paper, we further develop the approach proposed by Yoo et al. [2002], to address more general conjunctions of closed-loop properties, including controllability, normality and completeness. It is organized as follows. Section 3 formally defines the class of closed-loop properties under consideration and states elementary consequences for the respective supremal sublanguages. Section 4 characterizes the supremal sublanguage as the supremal fixpoint of an operator stated in terms of sublanguages supremal w.r.t. individual closed-loop properties for the prefix-closed case. This leads to an iterative procedure, which, in Section 5, is shown to terminate. In Section 6, we apply our result to practically solve a synthesis problem for a class of sequential behaviours under partial observation.

2. PRELIMINARIES

Let Σ be a *finite alphabet*. 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 two strings $s, r \in \Sigma^*$ there exists $t \in \Sigma^*$ such that $s = rt$, we say r is a *prefix* of s, and write $r \le s$; if in addition $r \neq s$, we say r is a *strict prefix* of s and write $r < s$. A $*$-*language* (or short a *language*) over Σ is a subset $L \subseteq \Sigma^*$.

The *prefix* of a language $L \subseteq \Sigma^*$ is defined by $\operatorname{pre} L := \{r \in \Sigma^* \,|\, \exists\, s \in L : r \le s\}$. The prefix operator is also referred to as the *prefix-closure*, and, a language L is *prefix-closed* (or short *closed*) if $L = \operatorname{pre} L$. 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 *nonconflicting*. Given two languages L and K, we say K is *relatively closed w.r.t. L* if $K = \operatorname{pre}(K) \cap L$. The intersection $\operatorname{pre}(K) \cap L$ is always relatively closed w.r.t. L. If a language K is relatively closed w.r.t. a closed language, then K itself is closed. Given three languages K, E, $L \subseteq \Sigma^*$, such that K is relatively closed w.r.t. E and E is relatively closed w.r.t. L, then K is relatively closed w.r.t. L.

The *natural projection* $p_o \colon \Sigma^* \to \Sigma_o^*$, $\Sigma_o \subseteq \Sigma$, is defined iteratively: (1) let $p_o(\epsilon) := \epsilon$; (2) for $s \in \Sigma^*$, $\sigma \in \Sigma$, let $p_o(s\sigma) := p_o(s)\sigma$ if $\sigma \in \Sigma_o$, or, if $\sigma \notin \Sigma_o$, let $p_o(s\sigma) := p_o(s)$. The set-valued inverse p_o^{-1} of p_o is defined by $p_o^{-1}(r) := \{s \in \Sigma^* \,|\, p_o(s) = r\}$ for $r \in \Sigma_o^*$. When extended 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 $\Sigma_{uc} \subseteq \Sigma$, we say K is *controllable w.r.t. L*, if $K \subseteq \operatorname{pre} L$ and $\operatorname{pre}(K)\Sigma_{uc} \cap \operatorname{pre}(L) \subseteq \operatorname{pre} K$; see e.g. [Ramadge and Wonham, 1987]. Alternative definitions do not explicitly insist in $K \subseteq \operatorname{pre} L$; e.g. Cassandras and Lafortune [2008]. However, when discussing closed-loop behaviours $K \subseteq L$, the additional constraint is obviously fulfilled. With $\Sigma_o \subseteq \Sigma$ the set of observable events, we say K is *normal w.r.t. L*, if $\operatorname{pre} K = p_o^{-1} p_o \operatorname{pre}(K) \cap \operatorname{pre}(L)$. In analogy to controllability, this variant of normality is defined in terms of the prefix of K as in e.g. Cassandras and Lafortune [2008] and in contrast to e.g. Lin and Wonham [1988]. 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. Kumar et al. [1992].

Each one of the properties controllability, normality, completeness, closedness and relative closedness is retained under arbitrary union. Given a family of languages $(K_a)_{a \in A}$, $K_a \subseteq L$ for all $a \in A$, such that a particular combination of the mentioned properties is possessed by each K_a, then the union $\cup_{a \in A} K_a$ possesses the respective properties, too; see, e.g., Lin and Wonham [1988], Ramadge and Wonham [1987, 1989] regarding controllability, Lin and Wonham [1988] regarding normality, and Kumar et al. [1992] for completeness. Note that closedness and relative closedness is also retained under arbitrary intersection.

A *generator* is a tuple $G := (Q, \Sigma, \delta, Q_o, Q_m)$ with the finite *state set Q*, *marked states $Q_m \subseteq Q$*, *initial states $Q_o \subseteq Q$* and the *transition relation $\delta \subseteq Q \times \Sigma \times Q$*. The transition relation is also interpreted as a set-valued map where, for $q \in Q$ and $\sigma \in \Sigma$, $\delta(q,\sigma)$ denotes the set of states q' with $(q,\sigma,q') \in \delta$. By $\delta(q, \epsilon) := \{q\}$ and $\delta(q, s\sigma) := \delta(\delta(q, s), \sigma)$ we extend δ to a set-valued map on $Q \times \Sigma^*$. A state $q \in Q$ is *accessible*, if there exists $s \in \Sigma^*$ such that $q \in \delta(Q_o, s)$. A state $q \in Q$ is *coaccessible*, if there exists $s \in \Sigma^*$ such that $\delta(q, s) \cap Q_m \neq \emptyset$. A generator is *accessible* (*coaccessible*), if all states are accessible (coaccessible). A genartor is *trim* if it is both, accessible and coaccessible. A generator is *deterministic*, if, for each $q \in Q$, $\sigma \in \Sigma$, the sets $\delta(q,\sigma)$ and Q_o have no more than one element.

With a generator G, we associate the *generated* language $\mathrm{L}(G) := \{s \in \Sigma^* \,|\, \delta(Q_o, s) \neq \emptyset\}$ and the *marked* language $\mathrm{L_m}(G) := \{s \in \Sigma^* \,|\, \delta(Q_o, s) \cap Q_m \neq \emptyset\}$. A language is said to be regular, if it is marked by some generator. Throughout this paper, generators are considered to be deterministic. Regarding the marked and generated languages, this assumption is not restrictive. When only the marked language is of concern, a generator may also be assumed to be trim. The *product* $F = G \times H$ of two generators $G = (Q, \Sigma, \delta, Q_o, Q_m)$ and $H = (X, \Sigma, \xi, X_o, X_m)$ is defined by $F = (Z, \Sigma, \zeta, Q_o \times X_o, Q_m \times X_m)$ with $Z = Q \times X$ and $\zeta \subseteq Z \times \Sigma \times Z$, where $((q, x), \sigma, (q', x')) \in \zeta$ if and only if $(q, \sigma, q',) \in \delta$ and $(x, \sigma, x') \in \xi$. The product relates to language intersection: $\mathrm{L}(G \times H) = \mathrm{L}(G) \cap \mathrm{L}(H)$ and $\mathrm{L_m}(G \times H) = \mathrm{L_m}(G) \cap \mathrm{L_m}(H)$.

For generators $G = (Q, \Sigma, \delta, Q_o, Q_m)$ and $H = (X, \Sigma, \xi, X_o, X_m)$, we say H is a *subautomaton* of G and write $H \sqsubseteq G$, if $s \in \mathrm{L}(H)$ implies $\delta(Q_o, s) = \xi(Q_o, s)$. For accessible and deterministic generators, we have $H \sqsubseteq G$ if and only if $X \subseteq Q$, $X_o \subseteq Q_o$ and $\xi \subseteq \delta$; see also [Cho and Marcus, 1989], Lemma 2.1.

The set of *ω-strings* (also *infinite words*) over Σ is denoted $\Sigma^\omega := \{w | w = \sigma_1\sigma_2\sigma_3\cdots$, with $\sigma_i \in \Sigma$ for all $i \in \mathbb{N}\}$. An *ω-language* over Σ is a subset $\mathcal{L} \subseteq \Sigma^\omega$. If for two strings $w \in \Sigma^\omega$, $r \in \Sigma^*$, there exists $v \in \Sigma^\omega$ such that $w = rv$, we say r is a *strict prefix* of w and write $r < w$. The *prefix* of an ω-language $\mathcal{L} \subseteq \Sigma^\omega$ is defined $\operatorname{pre} \mathcal{L} := \{s \in \Sigma^* \,|\, \exists w \in \mathcal{L} : s < w\}$. Note that the prefix of any ω-language is complete. The prefix operator distributes over arbitrary unions of ω-languages. However, for the intersection of two ω-languages $\mathcal{L}$ and $\mathcal{H}$ over Σ, we have $\operatorname{pre}(\mathcal{L} \cap \mathcal{H}) \subseteq \operatorname{pre}(\mathcal{L}) \cap \operatorname{pre}(\mathcal{H})$. If equality holds, we say $\mathcal{L}$ and $\mathcal{H}$ are *nonconflicting*.

The *limit* of a $*$-language $L \subseteq \Sigma^*$ is defined $\lim L := \{w \in \Sigma^\omega | w$ has infinitely many prefixes in $L\}$. If and only if a $*$-language $L \subseteq \Sigma^*$ is complete, we have $\operatorname{pre} \lim L = \operatorname{pre} L$. If and only if a $*$-language $L \subseteq \Sigma^*$ is complete and prefix-closed, we have $\operatorname{pre} \lim L = L$; see [Kumar et al., 1992]. The *topological closure* (or short *closure*) of an ω-language $\mathcal{L}$ is defined by $\operatorname{clo} \mathcal{L} := \lim \operatorname{pre} \mathcal{L}$. In general, we have $\mathcal{L} \subseteq \operatorname{clo} \mathcal{L}$. An ω-language $\mathcal{L}$ is said to be *topologically closed* (or short *closed*) if $\operatorname{clo} \mathcal{L} = \mathcal{L}$, i.e., if $\lim \operatorname{pre} \mathcal{L} = \mathcal{L}$. The limit of a prefix-closed $*$-language is topologically closed. Given two ω-languages $\mathcal{L}, \mathcal{H} \subseteq \Sigma^\omega$, we say $\mathcal{L}$ is *relatively closed w.r.t. $\mathcal{H}$* if $\mathcal{L} = \operatorname{clo}(\mathcal{L}) \cap \mathcal{H}$. The closure operator distributes over finite unions of ω-languages.

In order to define the natural projection $p_o^\omega w \in \Sigma_o^\omega \cup \Sigma_o^*$ of an ω-string $w \in \Sigma^\omega$, let $(s_n) \subseteq \Sigma^*$ be a strictly monotone sequence of prefixes of w, i.e., $s_n < s_{n+1} < w$ for all $n \in \mathbb{N}$. Then, $p_o^\omega w$ is defined as the limit $v \in \Sigma_o^\omega \cup \Sigma_o^*$ of the monotone sequence $(p_o(s_n))$, i.e., for all $n \in \mathbb{N}$ we have $p_o(s_n) < v$ and for all $r < v$ there exists $n \in \mathbb{N}$ with $r < p_o(s_n)$. The set-valued inverse $p_o^{-\omega}$ of p_o^ω is defined by $p_o^{-\omega}(v) := \{w \in \Sigma^\omega \,|\, p_o^\omega(w) = v\}$ for $v \in \Sigma_o^\omega \cup \Sigma_o^*$. When extended to ω-languages, the projection distributes over unions, and the inverse projection distributes over unions and intersections. Projection and inverse projection commute with the prefix-operator. For a prefix-closed $L \subseteq \Sigma^*$ we have $p_o^\omega \lim(L) \cap \Sigma_o^\omega = \lim p_o(L)$, and, for any $L_o \subseteq \Sigma_o^*$, we have $p_o^{-\omega} \lim(L_o) = \lim p_o^{-1}(L_o) \cap p_o^{-\omega}\Sigma_o^\omega$.

3. SUPREMAL SUBLANGUAGES

Rather than to explicitly address e.g. controllability or normality, we give a general discussion of supremal sublanguages for closed-loop properties that are, (A1), retained under arbitrary union and that, (A2), do not refer to task completion modelled by a marked language. Formally, we may represent such a property as a set $\mathcal{A}$ of languages $K \subseteq \Sigma^*$ and impose the following conditions:

(A1) $\mathcal{B} \subseteq \mathcal{A} \quad \Rightarrow \quad \cup_{K \in \mathcal{B}} K \in \mathcal{A}$,
(A2) $K \in \mathcal{A} \quad \Leftrightarrow \quad \operatorname{pre} K \in \mathcal{A}$.

Condition (A1) requires $\mathcal{A}$ to be a complete upper semi-lattice w.r.t. set inclusion. With the convention that the empty union evaluates as the empty set, (A1) implies $\emptyset \in \mathcal{A}$. Condition (A1) is a nearby prerequisite that ensures the existence of supremal sublanguages. Condition (A2) restricts the respective closed-loop property to be characterized in terms of the prefix of the candidate language. For a plant $L \subseteq \Sigma^*$ and the common partitioning $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$, any of the properties controllability, normality and completeness can be represented as a set $\mathcal{A}$ that complies with (A1) and (A2).

Given a closed-loop property $\mathcal{A}$ and a *language-inclusion specification* $E \subseteq \Sigma^*$, we are interested in the supremal sublanguage

$$\langle E\rangle^{\uparrow(\mathcal{A})} := \sup\{K \subseteq E \,|\, K \in \mathcal{A}\}. \qquad (1)$$

Depending of the particular property $\mathcal{A}$, additional assumptions on E may have relevant implications; e.g., for controllability, $E \subseteq L$ can be used to imply $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq L$. However, the general discussion below is valid for any specification $E \subseteq \Sigma^*$.

When viewed as a map with two arguments, $\langle\cdot\rangle^{\uparrow(\cdot)}$ is monotone, i.e., $E_1 \subseteq E_2$ and $\mathcal{A}_1 \subseteq \mathcal{A}_2$ implies $\langle E_1\rangle^{\uparrow(\mathcal{A}_1)} \subseteq \langle E_2\rangle^{\uparrow(\mathcal{A}_2)}$. The following proposition states immediate consequences of conditions (A1) and (A2) for the supremal sublanguage $\langle E\rangle^{\uparrow(\mathcal{A})}$.

Proposition 1. Let $\mathcal{A}$ be a set of languages over Σ that satisfies conditions (A1) and (A2). Then, for any specification $E \subseteq \Sigma^*$, the following holds:

(i) $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq E$, $\langle E\rangle^{\uparrow(\mathcal{A})} \in \mathcal{A}$, and $\langle\langle E\rangle^{\uparrow(\mathcal{A})}\rangle^{\uparrow(\mathcal{A})} = \langle E\rangle^{\uparrow(\mathcal{A})}$,
(ii) $\langle E\rangle^{\uparrow(\mathcal{A})}$ is relatively closed w.r.t. E,
(iii) $\langle \mathrm{pre}\, E\rangle^{\uparrow(\mathcal{A})}$ is closed,
(iv) $\mathrm{pre}\,(\langle E\rangle^{\uparrow(\mathcal{A})}) \subseteq \langle \mathrm{pre}\, E\rangle^{\uparrow(\mathcal{A})}$.

Proof. Ad (i). Recall that for any set $\mathcal{B}$ of languages, $\sup \mathcal{B} = \cup_{K\in\mathcal{B}} K$. This implies $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq E$ and, with (A1), $\langle E\rangle^{\uparrow(\mathcal{A})} \in \mathcal{A}$. For any $H \in \mathcal{A}$, we have $H = \langle H\rangle^{\uparrow(\mathcal{A})}$, and, hence, $\langle\cdot\rangle^{\uparrow(\mathcal{A})}$ is idempotent. Ad (ii) [see also Cho and Marcus [1989], Lemma B.2]. Let $F := \mathrm{pre}\,(\langle E\rangle^{\uparrow(\mathcal{A})}) \cap E$. Referring to (i), observe $\langle E\rangle^{\uparrow(\mathcal{A})} = \langle E\rangle^{\uparrow(\mathcal{A})} \cap E \subseteq F$. To establish the converse inclusion $F \subseteq \langle E\rangle^{\uparrow(\mathcal{A})}$, we observe that $\mathrm{pre}\, F \subseteq \mathrm{pre}\,(\langle E\rangle^{\uparrow(\mathcal{A})}) = \mathrm{pre}\,(\langle E\rangle^{\uparrow(\mathcal{A})} \cap E) \subseteq \mathrm{pre}\, F$. In the latter inclusion we must have equality, in particular $\mathrm{pre}\, F = \mathrm{pre}\,(\langle E\rangle^{\uparrow(\mathcal{A})})$, which implies by (i) and (A2) that $F \in \mathcal{A}$. $F \subseteq E$ is obvious, and, thus, $F \subseteq \langle E\rangle^{\uparrow(\mathcal{A})}$. Ad (iii). By (ii), $\langle \mathrm{pre}\, E\rangle^{\uparrow(\mathcal{A})}$ is relatively closed w.r.t. a closed language and, thus, closed. Ad (iv). Monotonicity of $\langle\cdot\rangle^{\uparrow(\mathcal{A})}$ implies $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq \langle \mathrm{pre}\, E\rangle^{\uparrow(\mathcal{A})}$. Taking the prefix on both sides, (iii) implies (iv). □

Observe that conditions (A1) and (A2) are retained under intersection, i.e., if $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy (A1) and (A2), then so does $\mathcal{A} := \mathcal{A}_1 \cap \mathcal{A}_2$. In particular, Proposition 1 immediately applies not only to controllability, normality and completeness, but also to any conjunction thereof. The following proposition addresses the conjunction of two closed-loop properties explicitly, and, referring to the monotonicity of $\langle\cdot\rangle^{\uparrow(\cdot)}$, is readily applied to finite conjunctions of more than two properties.

Proposition 2. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two sets of languages over Σ that both satisfy conditions (A1) and (A2). Then, for any $E \in \Sigma^*$, we have

(i) $\langle E\rangle^{\uparrow(\mathcal{A}_1\cap\mathcal{A}_2)} \subseteq \langle\langle E\rangle^{\uparrow(\mathcal{A}_1)}\rangle^{\uparrow(\mathcal{A}_2)}$,
(ii) $\langle\langle E\rangle^{\uparrow(\mathcal{A}_1\cap\mathcal{A}_2)}\rangle^{\uparrow(\mathcal{A}_1)} = \langle E\rangle^{\uparrow(\mathcal{A}_1\cap\mathcal{A}_2)}$,
(iii) $\mathrm{pre}\,(\langle\langle E\rangle^{\uparrow(\mathcal{A}_1)}\rangle^{\uparrow(\mathcal{A}_2)}) \subseteq \langle\langle \mathrm{pre}\, E\rangle^{\uparrow(\mathcal{A}_1)}\rangle^{\uparrow(\mathcal{A}_2)}$,
(iv) $\mathrm{pre}\,(\langle E\rangle^{\uparrow(\mathcal{A}_1\cap\mathcal{A}_2)}) \subseteq \langle\langle \mathrm{pre}\, E\rangle^{\uparrow(\mathcal{A}_1)}\rangle^{\uparrow(\mathcal{A}_2)}$.

Proof. Ad (i). By Proposition 1, (i), and monotonicity of $\langle\cdot\rangle^{\uparrow(\cdot)}$ we obtain $\langle E\rangle^{\uparrow(\mathcal{A}_1\cap\mathcal{A}_2)} = \langle\langle E\rangle^{\uparrow(\mathcal{A}_1\cap\mathcal{A}_2)}\rangle^{\uparrow(\mathcal{A}_1\cap\mathcal{A}_2)} \subseteq \langle\langle E\rangle^{\uparrow(\mathcal{A}_1)}\rangle^{\uparrow(\mathcal{A}_2)}$. Ad (ii). The equality follows from $\langle E\rangle^{\uparrow(\mathcal{A}_1\cap\mathcal{A}_2)} \in \mathcal{A}_1$. Ad (iii). The claim is a consequence of Proposition 1, (iv), and monotonicity of $\langle\cdot\rangle^{\uparrow(\cdot)}$. Ad (iv). The inclusion follows by above (i) and (iii) and monotonicity of $\langle\cdot\rangle^{\uparrow(\cdot)}$. □

4. FIXPOINT ITERATION

Given a closed-loop property $\mathcal{A}$ of languages over Σ and a specification $E \subseteq \Sigma^*$, Yoo et al. [2002] represent $\langle E\rangle^{\uparrow(\mathcal{A})}$ as a fixpoint of the operator $\Omega(K) := \langle \mathrm{pre}\, K\rangle^{\uparrow(\mathcal{A})} \cap E$. This representation has the particular benefit that it enables the computation of the supremal sublanguage of a not necessarily prefix-closed specification by means of considerably less involved methods developed for the prefix-closed case.

In this section, we generalize the approach of Yoo et al. [2002] to explicitly address conjunctions of closed-loop properties $\mathcal{A}_1$, $\mathcal{A}_2$, … $\mathcal{A}_m$ over Σ that conform with (A1) and (A2). Given a specification $E \subseteq \Sigma^*$, we define the operator $\Omega\colon \Sigma^* \to \Sigma^*$:

$$\Omega(K) := [\langle\cdot\rangle^{\uparrow(\mathcal{A}_m)} \circ \langle\cdot\rangle^{\uparrow(\mathcal{A}_{m-1})} \circ \cdots \circ \langle\cdot\rangle^{\uparrow(\mathcal{A}_1)}](\mathrm{pre}\, K) \cap E, \quad (2)$$

for $K \subseteq \Sigma^*$, and, with $\circ$ denoting concatenation of maps as in $[f \circ g \circ h](x) := f(g(h(x)))$ for endomorphisms f, g and h. To this end, we note the following properties of Ω.

Proposition 3. Given sets of languages $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_m$ over Σ, each conforming with (A1) and (A2), denote the intersection by $\mathcal{A}$. Consider an arbitrary specification $E \subseteq \Sigma^*$ and the operator Ω defined by Eq. (2). Then, for any $K \subseteq \Sigma^*$, we have that

(i) $\Omega(K)$ is relatively closed w.r.t. E,
(ii) $\Omega(K) \subseteq \mathrm{pre}\,(K) \cap E$,
(iii) $K = \Omega(K) \quad\Rightarrow\quad K \subseteq \langle E\rangle^{\uparrow(\mathcal{A})}$,
(iv) $\langle E\rangle^{\uparrow(\mathcal{A})} = \Omega(\langle E\rangle^{\uparrow(\mathcal{A})})$.

Proof. Ad (i). From Proposition 1, (iii), $\langle\cdot\rangle^{\uparrow(\mathcal{A}_j)}$, $j \le m$, maps closed languages to closed languages. Hence, $[\langle\cdot\rangle^{\uparrow(\mathcal{A}_m)} \circ \cdots \circ \langle\cdot\rangle^{\uparrow(\mathcal{A}_1)}](\mathrm{pre}\, K)$ is closed, and, $\Omega(K)$ is relatively closed w.r.t. E. Ad (ii). For any $j \le m$ and any $H \subseteq \Sigma^*$, we have $\langle H\rangle^{\uparrow(\mathcal{A}_j)} \subseteq H$. Thus, $\Omega(K) \subseteq \mathrm{pre}\,(K) \cap E$. Ad (iii). Provided that $K = \Omega(K)$, we will establish that $K \subseteq E$ and $K \in \mathcal{A}$. The former is obvious. With $\mathrm{pre}\, K = \mathrm{pre}\, \Omega(K) \subseteq [\langle\cdot\rangle^{\uparrow(\mathcal{A}_m)} \circ \cdots \circ \langle\cdot\rangle^{\uparrow(\mathcal{A}_1)}](\mathrm{pre}\, K) \subseteq \mathrm{pre}\, K$, the inclusions turn out as equalities. Since each map $\langle\cdot\rangle^{\uparrow(\mathcal{A}_j)}$ restricts its respective argument, it follows $\mathrm{pre}\, K = \langle \mathrm{pre}\, K\rangle^{\uparrow(\mathcal{A}_j)}$ for each $j \le m$. Referring to (A1), this implies $\mathrm{pre}\, K \in \mathcal{A}_j$, and, by (A2), $K \in \mathcal{A}_j$. Thus, $K \in \mathcal{A}$. Ad (iv). By Proposition 1, (ii), $\langle E\rangle^{\uparrow(\mathcal{A})}$ is relatively closed w.r.t. E. Thus, above (ii) implies $\Omega(\langle E\rangle^{\uparrow(\mathcal{A})}) \subseteq \mathrm{pre}\,(\langle E\rangle^{\uparrow(\mathcal{A})}) \cap E = \langle E\rangle^{\uparrow(\mathcal{A})}$. Conversely, $\langle E\rangle^{\uparrow(\mathcal{A})} = \langle\langle E\rangle^{\uparrow(\mathcal{A})}\rangle^{\uparrow(\mathcal{A})} = \langle\langle E\rangle^{\uparrow(\mathcal{A})}\rangle^{\uparrow(\mathcal{A})} \cap E \subseteq \mathrm{pre}\,(\langle\langle E\rangle^{\uparrow(\mathcal{A})}\rangle^{\uparrow(\mathcal{A})}) \cap E \subseteq \Omega(\langle E\rangle^{\uparrow(\mathcal{A})})$, where the last inclusion is a consequence of Proposition 2, (vi). □

We will establish by (i) and (ii) that the sequence

$$K_0 = E,\; K_{i+1} = \Omega(K_i) \qquad (3)$$

is monotonically decreasing and, hence, converges by definition to the intersection $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$. By (iii) and (iv), $\langle E\rangle^{\uparrow(\mathcal{A})}$ is the supremal fixpoint of Ω. In particular, if K_∞ is a fixpoint, (iii) implies $K_\infty \subseteq \langle E\rangle^{\uparrow(\mathcal{A})}$. The following proposition establishes the converse inclusion.

Proposition 4. Under the same hypothesis as in Proposition 3, and, for all $i \in \mathbb{N}_0$ in Iteration (3), we have that

(i) K_i is relatively closed w.r.t. E,
(ii) $K_{i+1} \subseteq K_i$,
(iii) $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq K_i$.

Proof. Ad (i). For $K_0 = E$ the claim is obviously true, and for $K_{i+1} = \Omega(K_i)$ it is implied by Proposition 3, (i). Ad (ii). Clearly, $K_1 \subseteq E$. Referring to above (i) and Proposition 3, (ii), we have $K_{i+1} = \Omega(K_i) \subseteq \mathrm{pre}\,(K_i) \cap E = K_i$. Ad (iii). For $K_0 = E$ the claim is obviously true. Under the hypothesis that $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq K_i$ holds for some $i \in \mathbb{N}_0$, we will establish the inclusion for $i+1$. Observe by Proposition 1, (ii), and Proposition 2, (ii), that $\langle E\rangle^{\uparrow(\mathcal{A})} =$

$\operatorname{pre}(\langle E\rangle^{\uparrow(\mathcal{A})}) \cap E = \operatorname{pre}([\langle\cdot\rangle^{\uparrow(\mathcal{A}_m)} \circ \cdots \circ \langle\cdot\rangle^{\uparrow(\mathcal{A}_1)}](\langle E\rangle^{\uparrow(\mathcal{A})})) \cap E$. With the induction hypothesis and monotonicity of $\langle\cdot\rangle^{\uparrow(\mathcal{A}_j)}$, we obtain $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq \operatorname{pre}([\langle\cdot\rangle^{\uparrow(\mathcal{A}_m)} \circ \cdots \circ \langle\cdot\rangle^{\uparrow(\mathcal{A}_1)}](K_i)) \cap E \subseteq [\langle\cdot\rangle^{\uparrow(\mathcal{A}_m)} \circ \cdots \circ \langle\cdot\rangle^{\uparrow(\mathcal{A}_1)}](\operatorname{pre} K_i) \cap E = \Omega(K_i)$, where the last inclusion refers to Proposition 2, (iii). Thus we have indeed $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq K_{i+1}$ and (iii) follows by induction. □

By taking intersection over all $i \in \mathbb{N}_0$, (ii) implies that K_∞ is relatively closed w.r.t. E and (iii) implies $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq K_\infty$. Together with Proposition 3, (iii), we have $\langle E\rangle^{\uparrow(\mathcal{A})} = K_\infty$, provided that K_∞ is a fixpoint of Ω. We summarize our results so far.

Theorem 5. Given sets of languages $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_m$ over Σ, such that each conforms with (A1) and (A2), denote the intersection by $\mathcal{A}$, and consider an arbitrary specification $E \subseteq \Sigma^*$. Then, for Ω defined by Eq. (2), Iteration (3) converges to the limit $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$. Furthermore, we have $\langle E\rangle^{\uparrow(\mathcal{A})} \subseteq K_\infty$, where equality holds if and only if K_∞ is a fixpoint of Ω. □

Remark. If the closed-loop properties $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_m$ were complete lattices, Ω would turn out $\cap$-continuous and, thus, K_∞ would be a fixpoint of Ω. However, the properties we want to address are not retained under arbitrary intersection.

5. FINITE CONVERGENCE

For a software implementation of Iteration (3), we from now on assume that for each individual closed-loop property $\mathcal{A}_j$, $j \le m$, the operator $\langle\operatorname{pre}(\cdot)\rangle^{\uparrow(\mathcal{A}_j)}$ retains regularity. If in addition the specification $E \subseteq \Sigma^*$ is regular, so will be the iterate K_i at any step $i \in \mathbb{N}_0$. Note that these assumptions alone neither imply a regular limit K_∞ nor *finite convergence*, i.e., the existence of $n \in \mathbb{N}_0$, such that K_n is a fixpoint of Ω and, hence, $K_\infty = K_n$. If, on the other hand, a fixpoint $K_\infty = K_n$ is indeed reached after a finite number of iterations, Theorem 5 implies $K_n = \langle E\rangle^{\uparrow(\mathcal{A})}$ and Iteration (3) provides means to compute $\langle E\rangle^{\uparrow(\mathcal{A})}$ based on implementations of $\langle\operatorname{pre}(\cdot)\rangle^{\uparrow(\mathcal{A}_j)}$, $j \le m$.

In the literature, algorithms for the computation of supremal sublanguages of regular languages are typically stated as iterations on a generator. In each step of the iteration, the algorithm removes states and/or transitions that conflict with the desired closed-loop property. Since there is never anything added to the iterate generator, finite convergence is obvious, and, the challenge is to establish supremality. Roughly speaking, the latter amounts to a strategic choice of the initial generator that must exhibit a "sufficiently rich" transition structure to realize the supremal sublanguage. Known algorithms for controllability, normality and completeness also share the particular feature that the initial generator can be chosen as the product of two generators, where one realizes the specification. We state condition (A3), which is satisfied for closed-loop properties $\mathcal{A}$ that can be synthesized in the manner just described.

(A3) For any generator H, there exists a generator $G_\mathcal{A}$ with $\mathrm{L}(G_\mathcal{A}) = \Sigma^*$, such that

$$C \sqsubseteq G_\mathcal{A} \times H \quad \Rightarrow \quad (\exists C^\uparrow \sqsubseteq G_\mathcal{A} \times H)[\,\mathrm{L}(C^\uparrow) = \langle \mathrm{L}(C)\rangle^{\uparrow(\mathcal{A})}\,].$$

The technical requirement $\mathrm{L}(G_\mathcal{A}) = \Sigma^*$ prevents the above implication to be trivially satisfied. For a concise discussion regarding the individual closed-loop properties controllability, normality and completeness, consider a plant $L \subseteq \Sigma^*$, a specification $E \subseteq \Sigma^*$, both prefix-closed, with finite automaton realizations $L = \mathrm{L}(G)$ and $E = \mathrm{L}(H)$.

Controllability. The algorithm for computing a realization of the supremal controllable sublanguage, as presented in [Ramadge and Wonham, 1987], effectively starts with a candidate $C_0 = G \times H$ and then successively removes states and transitions which conflict with controllability. It terminates with a realization of the supremal controllable sublanguage of E. To satisfy the implication in (A3), we can choose any $G_\mathcal{A}$ with $G \sqsubseteq G_\mathcal{A}$. Starting from G, it is straightforward to construct $G_\mathcal{A}$, $G \sqsubseteq G_\mathcal{A}$, by inserting transitions to an additional dump-state in order to also fulfill the technical requirement $\mathrm{L}(G_\mathcal{A}) = \Sigma^*$. Note that this does not affect the algorithm, which will on inititialization $C_0 = G \times H$ remove the additional transitions.

Normality. An algorithm for computing a realization of the supremal normal sublanguage is developed in Cho and Marcus [1989], Sections 2 and 3. Following this discussion, a generator $C_0 = R \times R_{\mathrm{obs}}$ can be used as a first candidate for subsequent removal of states and transitions. Here, R_{obs} denotes a so called *observer* for $R = G \times H$ with additional self-loop transitions for unobservable events and with a dump-state for strings $s \notin L \cap E$, i.e., $\mathrm{L}(R_{\mathrm{obs}}) = \Sigma^*$. Based on past observations within Σ_o^*, the state of R_{obs} encodes the available information on the actual state in R; see Cho and Marcus [1989] for further motivation and a detailed construction. For the purpose of this paper, we note that any $G_\mathcal{A}$ with $G \times R_{\mathrm{obs}} \sqsubseteq G_\mathcal{A}$ satisfies the implication in (A3). As for controllability, we construct $G_\mathcal{A}$ such that $G \times R_{\mathrm{obs}} \sqsubseteq G_\mathcal{A}$ and $\mathrm{L}(G_\mathcal{A}) = \Sigma^*$, and, thereby satisfy (A3).

Completeness. To compute the supremal complete sublanguage of $L \cap E$, start with the candidate $C_0 = G \times H$ and repeatedly remove transitions to states with no enabled events. Terminate, when no such transitions exist anymore. The resulting generator realizes the supremal complete sublanguage of $L \cap E$. The described procedure is a special case of the one presented in Kumar et al. [1992], addressing complete and controllable sublanguages. We construct $G_\mathcal{A}$ as for controllability in order to satisfy (A3).

To address conjunctions of controllability, normality and completeness, we consider the particular $G_\mathcal{A}$ from the above discussion of normality. For controllability and completeness, we may interpret $G \times R_{\mathrm{obs}}$ as an alternative plant realization to observe that $G_\mathcal{A}$ uniformly satisfies (A3) for all three properties. In general, we impose the below condition on $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_m$:

(A4) For any generator H, there exists a generator $G_\mathcal{A}$ with $\mathrm{L}(G_\mathcal{A}) = \Sigma^*$, such that for each property $\mathcal{A}_j$, $j \le m$,

$$C \sqsubseteq G_\mathcal{A} \times H \quad \Rightarrow \quad (\exists C^\uparrow \sqsubseteq G_\mathcal{A} \times H)[\,\mathrm{L}(C^\uparrow) = \langle \mathrm{L}(C)\rangle^{\uparrow(\mathcal{A}_j)}\,].$$

Remark. If (A4) is satisfied for $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_m$, then each individual $\mathcal{A}_j$, $j \le m$, satisfies (A3). Vice versa, if each individual property $\mathcal{A}_j$, $j \le m$, satisfies (A3) and if $G_\mathcal{A}$ in (A3) can be chosen uniformly for all H, then $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_m$ can be shown to satisfy (A4).

We now establish that all iterates K_i from Iteration (3) can be realized as subautomata of $G_\mathcal{A}$.

Proposition 6. Let $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_m$ denote sets of languages over Σ that comply with (A1)–(A4), and consider Iteration (3), with Ω defined by Eq. (2). Then, for any specification $E \subseteq \Sigma^*$ with trim realization H, there exists a generator $G_{\mathcal{A},H}$, such that at each step $i \in \mathbb{N}_0$ there exists a trim realization $H_i \sqsubseteq G_{\mathcal{A},H}$ of K_i, i.e., $\mathrm{L}(H_i) = \operatorname{pre} K_i$ and $\mathrm{L}_\mathrm{m}(H_i) = K_i$.

Proof. Let $G_\mathcal{A}$ denote the generator provided by (A4) and assign $G_{\mathcal{A},H} := G_\mathcal{A} \times H$. Furthermore, let $H_0 := \mathrm{Trim}(G_{\mathcal{A},H})$, where $\mathrm{Trim}(\cdot)$ first removes transitions to states that are not coaccessible and then removes states that are not accessible.

In particular, this operation retains the marked language and results in a subautomaton. Thus, H_0 satisfies the claim for $i = 0$. For a proof by induction, assume the claim to hold for some $i \in \mathbb{N}_0$. Thus, there exists a subautomaton $H_i \sqsubseteq G_{\mathcal{A},H}$ such that $\mathrm{pre}\, K_i = \mathrm{L}(H_i)$, $K_i = \mathrm{L_m}(H_i)$. By (A4), $\langle \mathrm{L}(H_i)\rangle^{\uparrow(\mathcal{A}_1)}$ can be generated by a subautomaton $H^1_{i+1} \sqsubseteq G_{\mathcal{A},H}$. Repeating this argument, we obtain a realization $H^m_{i+1} \sqsubseteq G_{\mathcal{A},H}$ of $[\langle\cdot\rangle^{\uparrow(\mathcal{A}_m)} \circ \cdots \circ \langle\cdot\rangle^{\uparrow(\mathcal{A}_1)}](\mathrm{L}(H_i))$. The respective languages are closed, and, hence, we may assume that H^m_{i+1} is accessible and all states are marked; i.e. $\mathrm{L}(H^m_{i+1}) = \mathrm{L_m}(H^m_{i+1})$. Then, by intersection with E, we obtain $K_{i+1} = \Omega(K_i) = \mathrm{L_m}(H^m_{i+1} \times H)$. Note that $H^m_{i+1} \sqsubseteq G_{\mathcal{A},H} = G_{\mathcal{A}} \times H$ implies the existence of a subautomaton $H'_{i+1} \sqsubseteq G_{\mathcal{A},H}$ such that $\mathrm{L_m}(H'_{i+1}) = K_{i+1}$; more specifically, H'_{i+1} can be obtained from the product $H^m_{i+1} \times H$ by renaming states. Finally, let $H_{i+1} := \mathrm{Trim}(H'_{i+1})$ to satisfy the claim for $i + 1$. □

As an immediate consequence of the above proposition and Theorem 5, Iteration (3) is seen to finitely converge to the supremal sublanguage $\langle E\rangle^{\uparrow(\mathcal{A})}$.

Theorem 7. Given sets of languages $\mathcal{A}_1$, $\mathcal{A}_2$, … $\mathcal{A}_m$ over Σ that comply with (A1)–(A4), denote the intersection by $\mathcal{A}$, and, consider a regular specification $E \subseteq \Sigma^*$. Then, for Ω defined by Eq. (2), Iteration (3) finitely converges to the limit $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$ with $K_\infty = \langle E\rangle^{\uparrow(\mathcal{A})}$.

Proof. By Proposition 6, all iterates K_i, $i \in \mathbb{N}_0$, can be realized as subautomata of some generator $G_{\mathcal{A},H}$. Since there exists only a finite number of such subautomata, there can only be finitely many different iterates K_i. Monotonicity from Proposition 4, (ii), then implies finite convergence. □

Remark. Finite convergence does not depend on the implementation of the individual operators $\langle \mathrm{pre}\,(\cdot)\rangle^{\uparrow(\mathcal{A}_j)}$. In particular, our main result is not restricted to the particular algorithms used for the verification of (A3) and (A4).

Remark. Finite convergence does not require E to be relatively closed w.r.t. the plant L. However, when E is relatively closed w.r.t. L, then, by Proposition 4, (i), the limit K_∞ is also relatively closed w.r.t. L. For controllable sublanguages, this addresses the common situation of *nonblocking supervision*.

6. APPLICATION

Controllability, normality and completeness, in conjunction, are closely related to a particular controller synthesis problem for *sequential behaviours*, i.e., plant dynamics that are modelled by ω-languages; see e.g. [Ramadge, 1989, Kumar et al., 1992, Thistle and Wonham, 1994]. We give a concise but self-contained discussion of the respective synthesis problem and motivate our study by a practical solution based on Theorem 7.

Formally, the synthesis problem is given as a tuple $(\Sigma,\ \mathcal{L},\ \mathcal{E})$, where:

(P1) Σ is the overall *alphabet*, with the common partitioning $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$; we impose the requirement $\Sigma_c \subseteq \Sigma_o$, i.e., all controllable events must be observable;
(P2) $\mathcal{L} \subseteq \Sigma^\omega$ is the *plant behaviour*; for the subsequent discussion we require $p_o\mathcal{L} \subseteq \Sigma_o^\omega$, i.e., the plant persistently issues observable events;
(P3) $\mathcal{E} \subseteq \Sigma^\omega$ is the *language-inclusion specification*; for our discussion, we assume $\mathcal{E}$ to be relatively closed w.r.t. $\mathcal{L}$, i.e. the specification must not impose liveness properties other than those possessed by the plant.

Given a *controller* $\mathcal{H}_o \subseteq \Sigma_o^\omega$, we consider $\mathcal{L} \parallel \mathcal{H}_o := \mathcal{L} \cap p_o^{-\omega}\mathcal{H}_o$ the *closed-loop behaviour*. We say, $\mathcal{H}_o$ is a *solution* to the synthesis problem, if the following conditions are satisfied:

(C1) $\mathcal{L}$ and $p_o^{-\omega}\mathcal{H}_o$ are nonconflicting;
(C2) $\mathrm{pre}\,(\mathcal{L}) \cap \mathrm{pre}\, p_o^{-\omega}(\mathcal{H}_o)$ is controllable w.r.t. $\mathrm{pre}\,\mathcal{L}$; and
(C3) $\mathcal{L} \parallel \mathcal{H}_o \subseteq \mathcal{E}$.

All three conditions (C1)–(C3) are retained under arbitrary union of controllers. In particular, a supremal solution $\mathcal{H}_o^{\Uparrow}$ uniquely exists. Furthermore, as a consequence of the specification $\mathcal{E}$ being relatively closed w.r.t. the plant $\mathcal{L}$, the closure of any solution again forms a solution. In particular, $\mathcal{H}_o^{\Uparrow}$ is closed.

For a finite representation, we assume that $\mathcal{L}$ and $\mathcal{E}$ can be expressed as limits $\mathcal{L} = \lim L$ and $\mathcal{E} = \lim E$ of some regular languages $L \subseteq \Sigma^*$ and $E \subseteq \Sigma^*$, respectively. Without loss of generality, we furthermore assume that L is complete. Referring to relative closedness of $\mathcal{E}$, it is readily verified that $\mathcal{E} = \lim(\mathrm{pre}\,(\mathcal{E}) \cap L)$. Thus, without loss of generality, we assume that E is relatively prefix-closed w.r.t. L.

The following fact characterizes the solutions of $(\Sigma,\ \mathcal{L},\ \mathcal{E})$ in terms of closed-loop properties; a proof is provided at the end of this paper.

Fact 8. Let $(\Sigma,\ \mathcal{L},\ \mathcal{E})$ be the above synthesis problem, represented by $\mathcal{L} = \lim L$ and $\mathcal{E} = \lim E$, where L is complete and E is relatively closed w.r.t. L. If $\mathcal{H}_o \subseteq \Sigma_o^\omega$ solves $(\Sigma,\ \mathcal{L},\ \mathcal{E})$, then $K := L \cap p_o^{-1}\mathrm{pre}\,\mathcal{H}_o \subseteq \Sigma^*$ exhibits the following properties:

(L1) K is complete,
(L2) K is controllable w.r.t. L,
(L3) K is normal w.r.t. L,
(L4) $K \subseteq E$, and
(L5) K is relatively prefix-closed w.r.t. L.

Vice versa, for any $K \subseteq \Sigma^*$ that satisfies (L1)–(L5), $\mathcal{H}_o := \lim p_o \mathrm{pre}\, K$ solves $(\Sigma,\ \mathcal{L},\ \mathcal{E})$. In particular, the supremal language $K^\uparrow \subseteq \Sigma^*$, that satisfies (L1)–(L4), also satisfies (L5), and, for $\mathcal{H}_o^\uparrow := \lim p_o\mathrm{pre}\, K^\uparrow$, we have $\mathcal{L} \parallel \mathcal{H}_o^\uparrow = \mathcal{L} \parallel \mathcal{H}_o^{\Uparrow}$. □

Given finite automata realizations of L and E, one can compute $K^\uparrow$ by Iteration (3), where we use three closed-loop properties $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ for controllability, normality and completeness. Theorem 7 guarantees finite convergence with the fixpoint $K_\infty = K^\uparrow$, which by $\mathcal{H}_o^\uparrow := \lim p_o\mathrm{pre}\, K^\uparrow$ solves $(\Sigma,\ \mathcal{L},\ \mathcal{E})$.

For further illustration, consider the plant L and the specification E' given by Figure 1 and Figure 2, respectively, where $\Sigma_c = \{\mathsf{a}, \mathsf{d}, \mathsf{e}\}$ and $\Sigma_{uo} = \{\mathsf{c}\}$.

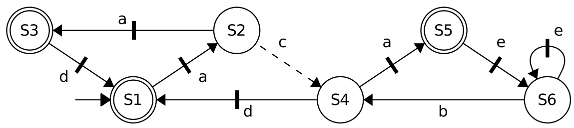

Fig. 1. Plant L

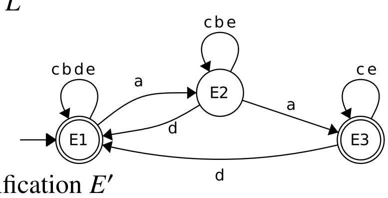

Fig. 2. Specification E'

Technically, the specification E' is not a subset of L, so we initialize Iteration (3) with $K_0 = E := E' \cap L$; see Figure 3. For the particular example, we expect Iteration (3) to, first, disable e in state $\mathsf{S5}$ for controllability; second, to disable a in state $\mathsf{S4}$ for completeness; and, third, to disable a in state $\mathsf{S2}$ for normality.

We have implemented the iteration as a `luafaudes`-script to obtain the fixpoint K_∞ shown in Figure 4.

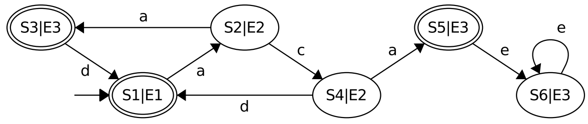

Fig. 3. Iterate $K_0 = E := E' \cap L$

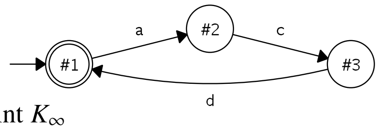

Fig. 4. Fixpoint K_∞

7. CONCLUSION

We have revisited and further developed a uniform approach for the computation of supremal sublanguages that was proposed by Yoo et al. [2002]. A fairly general iteration scheme is shown to finitely converge against the supremal sublanguage that exhibits a conjunction of desired closed-loop properties. For this result, four conditions are imposed on the properties. Condition (A1) requires the property to be retained under union and, thus, assures the supremal sublanguage to exist. Condition (A2) requires the property to address generated languages only and, thus, not to depend on task completion modelled by marked states. Conditions (A3) and (A4) effectively require that for each individual property there exists an algorithm to compute the respective supremal sublanguage and that some a priori knowledge of the resulting state set is available uniformly for all properties. Even though (A3) and (A4) refer only to the prefix-closed case, the iteration is valid also for plants and specifications that are not prefix-closed.

All conditions are fulfilled for controllability, normality, and completeness. Hence, one contribution of this paper is that we can now compute supremal sublanguages that are all controllable, normal and complete, and that we can do so for not necessarily closed specifications and plants. It should also be noted, that the iteration formula is rather simple and that it only refers to supremal sublanguage operators for the individual properties and for the prefix-closed case. This allows for straightforward implementations, which can be used to generate test cases for the validation of alternative approaches.

REFERENCES

R. D. Brandt, V. Garg, R. Kumar, F. Lin, S. I. Marcus, and W. M. Wonham. Formulas for calculating supremal controllable and normal sublanguages. *Systems and Control Letters*, 15: 111–117, 1990.

C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, second edition, 2008.

H. Cho and S. I. Marcus. On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation. *Maths. of Control, Signals & Systems*, 2:47–69, 1989.

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

F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44:173–198, 1988.

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.

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.

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

J. G. Thistle and W. M. Wonham. Supervision of infinite behavior of discrete event systems. *SIAM J. Control and Optimization*, 32:1098–1113, 1994.

T.-S. Yoo, S. Lafortune, and F. Lin. A uniform approach for computing supremal sublanguages arising in supervisory control theory. *Technical Report, Department of EECS, University of Michigan*, 2002.

APPENDIX

For the sake of completeness, we provide a proof for Fact 8. Given a solution $\mathcal{H}_o \subseteq \Sigma_o^\omega$, let $K := L \cap p_o^{-1}\mathrm{pre}\,\mathcal{H}_o$. Obviously, (L5) is satisfied. To verify (L1)–(L4), we first establish

$$\mathrm{pre}\,K = \mathrm{pre}\,(\mathcal{L} \parallel \mathcal{H}_o)\,. \tag{4}$$

For an arbitrary $s \in \mathrm{pre}\,K$, pick $r \in \Sigma^*$ such that $sr \in K$. Hence, $sr \in \mathrm{pre}\,L$ and $sr \in p_o^{-1}\mathrm{pre}\,\mathcal{H}_o = \mathrm{pre}\,p_o^{-\omega}\mathcal{H}_o$. By (C1) we obtain $sr \in \mathrm{pre}\,(\mathcal{L} \cap p_o^{-\omega}\mathcal{H}_o)$, and thus $s \in \mathrm{pre}\,(\mathcal{L} \parallel \mathcal{H}_o)$. For the converse inclusion, consider an arbitrary $s \in \mathrm{pre}\,(\mathcal{L} \parallel \mathcal{H}_o)$ and pick $w \in \Sigma^\omega$ such that $sw \in \mathcal{L} \parallel \mathcal{H}_o$. In particular, we can pick an $r < w$ such that $sr \in L$. With $sr < sw$, observe that $sr \in \mathrm{pre}\,(\mathcal{L} \parallel \mathcal{H}_o) \subseteq \mathrm{pre}\,p_o^{-\omega}\mathcal{H}_o = p_o^{-1}\mathrm{pre}\,\mathcal{H}_o$. This concludes the proof of Eq. (4).

Properties (L1) and (L2) follow immediately from Eq. (4). Regarding (L3), observe $p_o^{-1}p_o\mathrm{pre}\,K = p_o^{-1}p_o\mathrm{pre}\,(L \cap p_o^{-1}\mathrm{pre}\,\mathcal{H}_o) \subseteq p_o^{-1}\mathrm{pre}\,\mathcal{H}_o = \mathrm{pre}\,p_o^{-\omega}\mathcal{H}_o$. With (C1) and Eq. (4), this implies $p_o^{-1}p_o\mathrm{pre}\,(K) \cap \mathrm{pre}\,(L) \subseteq \mathrm{pre}\,p_o^{-\omega}(\mathcal{H}_o) \cap \mathrm{pre}\,(\mathcal{L}) = \mathrm{pre}\,(\mathcal{L} \parallel \mathcal{H}_o) = \mathrm{pre}\,K$. For (L4), observe from (C3) and Eq. (4) that $\mathrm{pre}\,K \subseteq \mathrm{pre}\,E$, and, thus $K = \mathrm{pre}\,(K) \cap L \subseteq \mathrm{pre}\,(E) \cap L = E$.

Vice versa, consider any $K \subseteq \Sigma^*$ that complies with (L1)–(L5) and let $\mathcal{H}_o := \lim p_o\mathrm{pre}\,K$. In order to establish (C1)–(C3), we first show that Eq. (4) again holds true. Observe that $\mathrm{pre}\,(\mathcal{L} \parallel \mathcal{H}_o) = \mathrm{pre}\,(\lim(L) \cap p_o^{-\omega}\lim p_o\mathrm{pre}\,(K)) \subseteq \mathrm{pre}\,(L) \cap \mathrm{pre}\,p_o^{-\omega}\lim p_o\mathrm{pre}\,(K) \subseteq \mathrm{pre}\,(L) \cap p_o^{-1}p_o\mathrm{pre}\,(K)$. By (L3) we obtain $\mathrm{pre}\,(\mathcal{L} \parallel \mathcal{H}_o) \subseteq \mathrm{pre}\,K$. For the converse inclusion, consider an arbitrary $s \in \mathrm{pre}\,K$, and pick $r \in \Sigma^*$ such that $sr \in K$, and, by (L1), $w \in \Sigma^\omega$, such that $srw \in \lim K$. Observe with (L3) and (L5) that $K \subseteq p_o^{-1}p_o(K) \cap L \subseteq p_o^{-1}p_o\mathrm{pre}\,(K) \cap \mathrm{pre}\,(L) \cap L = \mathrm{pre}\,(K) \cap L = K$, and, hence, $srw \in \lim K = \lim(L \cap p_o^{-1}p_oK) \subseteq \mathcal{L} \cap \lim p_o^{-1}p_o\mathrm{pre}\,K = \mathcal{L} \cap p_o^{-\omega}\lim p_o\mathrm{pre}\,K = \mathcal{L} \cap p_o^{-\omega}\mathcal{H}_o$, where the 2nd last equality is by the prerequisite $p_o^\omega\mathcal{L} \subseteq \Sigma_o^\omega$. This concludes the proof of Eq. (4).

For (C1), observe by (L3) and Eq. (4) that $\mathrm{pre}\,(\mathcal{L}) \cap \mathrm{pre}\,p_o^{-\omega}(\mathcal{H}_o) \subseteq \mathrm{pre}\,(L) \cap p_o^{-1}p_o\mathrm{pre}\,(K) = \mathrm{pre}\,K = \mathrm{pre}\,(\mathcal{L} \cap p_o^{-\omega}\mathcal{H}_o)$. Again by Eq. (4), (C2) is an immediate consequence of (L2). Regarding (C3), observe by Eq. (4), by (L4) and by relative closedness of $\mathcal{L}$ w.r.t. $\mathcal{E}$, that $\mathcal{L} \parallel \mathcal{H}_o \subseteq \mathcal{L} \cap \lim\mathrm{pre}\,(\mathcal{L} \parallel \mathcal{H}_o) = \mathcal{L} \cap \lim\mathrm{pre}\,K \subseteq \mathcal{L} \cap \lim\mathrm{pre}\,E = \mathcal{E}$.

The supremal language $K^\uparrow \subseteq \Sigma^*$ with properties (L1)–(L4) is, by Proposition 1, (ii), relatively closed w.r.t. E. Thus, the prerequisite that E is relatively closed w.r.t. L implies (L5). In particular, $\mathcal{H}_o^\uparrow$ is a solution. Given an arbitrary closed solution $\mathcal{H}_o \subseteq \Sigma_o^\omega$, we now show that $\mathcal{L} \parallel \mathcal{H}_o = \lim K$ with $K := L \cap p_o^{-1}\mathrm{pre}\,\mathcal{H}_o$: by (L1) and Eq. (4) we obtain $\mathrm{clo}\lim K = \lim\mathrm{pre}\,K = \mathrm{clo}\,(\mathcal{L} \parallel \mathcal{H}_o)$, to observe with (L5) that $\mathcal{L} \parallel \mathcal{H}_o = \mathcal{L} \cap \mathrm{clo}\,p_o^{-\omega}(\mathcal{H}_o) = \mathcal{L} \cap \mathrm{clo}\,(\mathcal{L} \cap p_o^{-\omega}\mathcal{H}_o) = \lim(L) \cap \mathrm{clo}\lim(K) = \lim(L \cap \mathrm{pre}\,K) = \lim K$. In particular, we have $\mathcal{L} \parallel \mathcal{H}_o \subseteq \mathcal{L} \parallel \mathrm{clo}\,\mathcal{H}_o \subseteq \lim K^\uparrow$ for any solution $\mathcal{H}_o$. Thus, $\lim K^\uparrow = \mathcal{L} \parallel \mathcal{H}_o^\uparrow \subseteq \mathcal{L} \parallel \mathcal{H}_o^\Uparrow \subseteq \lim K^\uparrow$. This implies $\mathcal{L} \parallel \mathcal{H}_o^\uparrow = \mathcal{L} \parallel \mathcal{H}_o^\Uparrow$.