Discrete Event Dyn Syst (2018) 28:269–287
DOI 10.1007/s10626-017-0250-0

Characterizations and effective computation of supremal relatively observable sublanguages

Kai Cai[1] · Renyuan Zhang[2] · W. M. Wonham[3]

Received: 3 September 2016 / Accepted: 18 May 2017 / Published online: 4 July 2017

Abstract Recently we proposed *relative observability* for supervisory control of discrete-event systems under partial observation. Relative observability is closed under set unions and hence there exists the supremal relatively observable sublanguage of a given language. In this paper we present a new characterization of relative observability, based on which an operator on languages is proposed whose largest fixpoint is the supremal relatively observable sublanguage. Iteratively applying this operator yields a monotone sequence of languages; exploiting the linguistic concept of *support* based on Nerode equivalence, we prove for regular languages that the sequence converges finitely to the supremal relatively observable sublanguage, and the operator is effectively computable. Moreover, for the purpose of control, we propose a second operator that in the regular case computes the supremal relatively observable and controllable sublanguage.

This work was supported in part by JSPS KAKENHI Grant no. JP16K18122 and Program to Disseminate Tenure Tracking System, MEXT, Japan; the National Nature Science Foundation, China, Grant no. 61403308; the Natural Sciences and Engineering Research Council, Canada, Grant no. NSERC-DG-480599.

This article belongs to the Topical Collection: *Special Issue on Diagnosis, Opacity and Supervisory Control of Discrete Event Systems*
Guest Editors: Christos G. Cassandras and Alessandro Giua

✉ Kai Cai
kai.cai@eng.osaka-cu.ac.jp

Renyuan Zhang
ryzhang@nwpu.edu.cn

W. M. Wonham
wonham@ece.utoronto.ca

[1] Department of Electrical & Information Engineering, Osaka City University, Osaka, Japan

[2] School of Automation, Northwestern Polytechnical University, Xian, China

[3] Department of Electrical & Computer Engineering, University of Toronto, Toronto, Canada

Keywords Supervisory control · Partial-observation · Relative observability · Regular language · Nerode equivalence relation · Support relation · Discrete-event systems · Automata

1 Introduction

In Cai et al. (2015) we proposed *relative observability* for supervisory control of discrete-event systems (DES) under partial observation. The essence of relative observability is to set a fixed ambient language relative to which the standard observability conditions (Lin and Wonham 1988) are tested. Relative observability is proved to be stronger than observability (Lin and Wonham 1988; Cieslak et al. 1988), weaker than normality (Lin and Wonham 1988; Cieslak et al. 1988), and closed under arbitrary set unions. Therefore the supremal relatively observable sublanguage of a given language exists, and we developed an automaton-based algorithm to compute the supremal sublanguage. Other recent work on supervisory control under partial observation is reported in Yin and Lafortune (2016) and Cai et al. (2016).

In this paper and its conference precursor (Cai and Wonham 2016), we present a new characterization of relative observability. The original definition of relative observability in Cai et al. (2015) was formulated in terms of *strings*, while the new characterization is given in *languages*. Based on this characterization, we propose an operator on languages, whose largest fixpoint is precisely the supremal relatively observable sublanguage. Iteratively applying this operator yields a monotone sequence of languages. In the case where the relevant languages are regular, we prove that the sequence converges finitely (albeit with exponential computing effort) to the supremal relatively observable sublanguage, and the operator is effectively computable.

This new computation scheme for the supremal sublanguage is given entirely in terms of languages, and the convergence proof systematically exploits the concept of *support* (Wonham 2016, Section 2.8) based on Nerode equivalence relations (Hopcroft and Ullman 1979). The solution therefore separates out the linguistic essence of the problem from the implementational aspects of state computation using automaton models. This approach is in the same spirit as Wonham and Ramadge (1987) for controllability, namely operator fixpoint and successive approximation.

Moreover, the proposed language-based scheme allows more straightforward implementation, as compared to the automaton-based algorithm in Cai et al. (2015). In particular, we show that the language operator used in each iteration of the language-based scheme may be decomposed into a series of standard or well-known language operations (e.g. complement, union, subset construction); therefore off-the-shelf algorithms may be suitably assembled to implement the computation scheme. Our previous experience with the automaton-based algorithm in Cai et al. (2015) suggests that computing the supremal relatively observable sublanguage is fairly delicate and thus prone to error. Hence, it is advantageous to have two algorithms at hand so that one can double check the computation results, thereby ensuring presumed correctness based on consistency.

Finally, for the purpose of supervisory control under partial observation, we combine relative observability with controllability. In particular, we propose an operator which in the regular case effectively computes the supremal relatively observable and controllable sublanguage.

We note that Alves et al. (2016) recently proposed an algorithm that also computes the supremal relatively observable sublanguage and has exponential complexity with a lower-degree polynomial multiplier than our language-based algorithm (see Remark 2 below for details). This algorithm, like the one in Cai et al. (2015), is automaton-based; thus it provides little insight into the linguistic meaning of the operations at each iteration, and may not be decomposable into a set of well-known language computations. We also note that Moor et al. (2012) studied a general scheme of combining operators that iteratively compute supremal sublanguages with different properties. In this scheme, however, individual operators are applied only to prefix-closed languages. Consequently the scheme is generally not applicable to our problem where marked languages are considered to address nonblocking.

The rest of the paper is organized as follows. In Section 2 we present a new characterization of relative observability, and an operator on languages that yields an iterative scheme to compute the supremal relatively observable sublanguage. In Section 3 we prove that in the case of regular languages, the iterative scheme generates a monotone sequence of languages that is finitely convergent to the supremal relatively observable sublanguage. In Section 4 we combine relative observability and controllability, and propose an operator that effectively computes the supremal relatively observable and controllable sublanguage. Finally in Section 5 we state conclusions.

This paper extends its conference precursor (Cai and Wonham 2016) in the following respects. (1) In the main result of Section 3, Theorem 7, the bound on the size of the supremal sublanguage is tightened and the corresponding proof given. (2) The effective computability of the proposed operator is shown in Section 3.3. (3) Relative observability is combined with controllability in Section 4, and a new operator is presented that effectively computes the supremal relatively observable and controllable sublanguage.

2 Characterizations of relative observability and its supremal element

In this section, the concept of relative observability proposed in Cai et al. (2015) is first reviewed. Then we present a new characterization of relative observability, together with a fixpoint characterization of the supremal relatively observable sublanguage.

2.1 Relative observability

Let Σ be a finite event set. A string $s \in \Sigma^*$ is a *prefix* of another string $t \in \Sigma^*$, written $s \leq t$, if there exists $u \in \Sigma^*$ such that $su = t$. Let $L \subseteq \Sigma^*$ be a language. The *(prefix) closure* of L is $\overline{L} := \{s \in \Sigma^* \mid (\exists t \in L)\ s \leq t\}$. For partial observation, let the event set Σ be partitioned into Σ_o, the observable event subset, and Σ_{uo}, the unobservable subset (i.e. $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$). Bring in the *natural projection* $P : \Sigma^* \to \Sigma_o^*$ defined according to

$$\begin{aligned} P(\epsilon) &= \epsilon, \quad \epsilon \text{ is the empty string;} \\ P(\sigma) &= \begin{cases} \epsilon, & \text{if } \sigma \notin \Sigma_o, \\ \sigma, & \text{if } \sigma \in \Sigma_o; \end{cases} \\ P(s\sigma) &= P(s)P(\sigma), \quad s \in \Sigma^*, \sigma \in \Sigma. \end{aligned} \tag{1}$$

In the usual way, P is extended to $P : Pwr(\Sigma^*) \to Pwr(\Sigma_o^*)$, where $Pwr(\cdot)$ denotes powerset. Write $P^{-1} : Pwr(\Sigma_o^*) \to Pwr(\Sigma^*)$ for the *inverse-image function* of P.

Throughout the paper, let M denote the marked behavior of the plant to be controlled, and $C \subseteq M$ an imposed specification language. Let $K \subseteq C$. We say that K is *relatively* *observable* (with respect to M, C, and P), or simply C-observable, if the following two conditions hold:

(i) $(\forall s, s' \in \Sigma^*, \forall \sigma \in \Sigma)\ s\sigma \in \overline{K}, s' \in \overline{C}, s'\sigma \in \overline{M}, P(s) = P(s') \Rightarrow s'\sigma \in \overline{K}$

(ii) $(\forall s, s' \in \Sigma^*)\ s \in K, s' \in \overline{C} \cap M, P(s) = P(s') \Rightarrow s' \in K$.

In words, relative observability of K requires for every lookalike pair (s, s') in $\overline{C}$ that (i) s and s' have identical one-step continuations, if allowed in $\overline{M}$, with respect to membership in $\overline{K}$; and (ii) if each string is in M and one actually belongs to K, then so does the other. Note that the tests for relative observability of K are not limited to the strings in $\overline{K}$ (as with standard observability (Lin and Wonham 1988; Cieslak et al. 1988)), but apply to all strings in $\overline{C}$; for this reason, one may think of C as the *ambient* language, relative to which the conditions (i) and (ii) are tested.

We have proved in Cai et al. (2015) that in general, relative observability is stronger than observability, weaker than normality, and closed under arbitrary set unions. This implies that the relatively observable controlled behavior is generally more permissive than the normal one; in particular, one may disable any controllable events that are unobservable. Since a relatively observable language is also observable, one may always implement the language by a feasible and nonblocking supervisor (Wonham 2016).

Write

$$\mathcal{O}(C) = \{K \subseteq C \mid K \text{ is } C\text{-observable}\} \tag{2}$$

for the family of all C-observable sublanguages of C. Then $\mathcal{O}(C)$ is nonempty (the empty language $\emptyset$ belongs) and contains a unique supremal element

$$\sup \mathcal{O}(C) := \bigcup \{K \mid K \in \mathcal{O}(C)\} \tag{3}$$

i.e. the supremal relatively observable sublanguage of C.

2.2 Characterization of relative observability

For $N \subseteq \Sigma^*$, write $[N]$ for $P^{-1}P(N)$, namely the set of all lookalike strings to strings in N. A language N is *normal* with respect to M if $[N] \cap M = N$. For $K \subseteq \Sigma^*$ write

$$\mathcal{N}(K, M) = \{K' \subseteq K \mid [K'] \cap M = K'\}. \tag{4}$$

Since normality is closed under union, $\mathcal{N}(K, M)$ has a unique supremal element $\sup \mathcal{N}(K, M)$ which may be effectively computed (Cho and Marcus 1989; Brandt et al. 1990; Kumar et al. 1993).

Write

$$\overline{C}.\sigma := \{s\sigma \mid s \in \overline{C}\}, \quad \sigma \in \Sigma. \tag{5}$$

Let $K \subseteq C$ and define

$$D(\overline{K}) := \bigcup \left\{ [\overline{K} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma \right\}. \tag{6}$$

Thus $D(\overline{K})$ is the collection of strings of the form $t\sigma$ ($t \in \overline{C}, \sigma \in \Sigma$), that are lookalike to the strings in $\overline{K}$ ending with the same event σ. Note that if $K = \emptyset$ then $D(\overline{K}) = \emptyset$. For a string $s \in \overline{K}$, write $\bar{s}$ for $\overline{\{s\}}$, the set of prefixes of s; it is easily verified that for $t \in \Sigma^*$, if $t \in D(\bar{s})$ then $t \in D(\overline{K})$. This language $D(\overline{K})$ turns out to be key to the following characterization of relative observability.

Proposition 1 *Let $K \subseteq C \subseteq M$. Then K is C-observable if and only if*

$$(i')\ D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$$
$$(ii')\ [K] \cap \left(\overline{C} \cap M\right) = K.$$

Note that condition (i′) is in a form similar to controllability of K (Wonham and Ramadge 1987) (i.e. $\overline{K}\Sigma_u \cap \overline{M} \subseteq \overline{K}$, where Σ_u is the uncontrollable event set), although the expression $D(\overline{K})$ appearing here is more complicated owing to the presence of the normality operator $[\cdot]$. Condition (ii′) is simply normality of K with respect to $\overline{C} \cap M$.

Proof of Proposition 1. We first show that (i′) $\Leftrightarrow$ (i), and then (ii′) $\Leftrightarrow$ (ii).

1. (i′) $\Rightarrow$ (i). Let $s, s' \in \Sigma^*$, $\sigma \in \Sigma$, and assume that $s\sigma \in \overline{K}$, $s' \in \overline{C}$, $s'\sigma \in \overline{M}$, and $P(s) = P(s')$. It will be shown that $s'\sigma \in \overline{K}$. Since $K \subseteq C$, we have $\overline{K} \subseteq \overline{C}$ and

$$\begin{aligned} s\sigma \in \overline{K} &\Rightarrow s\sigma \in \overline{K} \cap \overline{C}.\sigma \\ &\Rightarrow s'\sigma \in [\overline{K} \cap \overline{C}.\sigma] \\ &\Rightarrow s'\sigma \in [\overline{K} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \\ &\Rightarrow s'\sigma \in D(\overline{K}) \\ &\Rightarrow s'\sigma \in D(\overline{K}) \cap \overline{M} \\ &\Rightarrow s'\sigma \in \overline{K} \quad (\text{by}(i')). \end{aligned}$$

2. (i′) $\Leftarrow$ (i). Let $s \in D(\overline{K}) \cap \overline{M}$. According to Eq. 6 $\epsilon \notin D(\overline{K})$; thus $s \neq \epsilon$. Let $s = t\sigma$ for some $t \in \Sigma^*$ and $\sigma \in \Sigma$. Then

$$\begin{aligned} s \in D(\overline{K}) \cap \overline{M} &\Rightarrow t\sigma \in [\overline{K} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \cap \overline{M} \\ &\Rightarrow t \in \overline{C}, t\sigma \in \overline{M}, \\ &\quad\ (\exists t' \in \Sigma^*)(P(t) = P(t'), t'\sigma \in \overline{K} \cap \overline{C}.\sigma) \\ &\Rightarrow t\sigma \in \overline{K}, \quad (\text{by (i)}) \\ &\Rightarrow s \in \overline{K}. \end{aligned}$$

3. (ii′) $\Rightarrow$ (ii). Let $s, s' \in \Sigma^*$ and assume that $s \in K$, $s' \in \overline{C} \cap M$, and $P(s) = P(s')$. Then

$$\begin{aligned} s \in \overline{K} &\Rightarrow s' \in [\overline{K}] \\ &\Rightarrow s' \in [\overline{K}] \cap \overline{C} \cap M \\ &\Rightarrow s'\sigma \in K \quad (\text{by(ii}')). \end{aligned}$$

4. (ii) $\Rightarrow$ (ii′). ($\supseteq$) holds because $K \subseteq [K]$ and $K \subseteq \overline{C} \cap M$. To show ($\subseteq$), let $s \in [K]$ and $s \in \overline{C} \cap M$. Then there exists $s' \in K$ such that $P(s) = P(s')$. Therefore by (ii) we derive $s \in K$. □

Thanks to the characterization of relative observability in Proposition 1, we rewrite $\mathcal{O}(C)$ in Eq. 2 as follows:

$$\mathcal{O}(C) = \{K \subseteq C \mid D(\overline{K}) \cap \overline{M} \subseteq \overline{K}\ \&\ [K] \cap \left(\overline{C} \cap M\right) = K\}. \tag{7}$$

In the next subsection, we will characterize the supremal element $\sup \mathcal{O}(C)$ as the largest fixpoint of a language operator.

2.3 Fixpoint characterization of sup $\mathcal{O}(C)$

Given a language $K \subseteq \Sigma^*$, let

$$F(K) := \{s \in \overline{K} \mid D(\overline{s}) \cap \overline{M} \subseteq \overline{K}\}. \tag{8}$$

Lemma 2 *$F(K)$ is closed, i.e. $\overline{F(K)} = F(K)$. Moreover, $F(K) = \overline{K}$ if and only if $D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$.*

Proof First, let $s \in \overline{F(K)}$; then there exists $w \in \Sigma^*$ such that $sw \in F(K)$, i.e. $sw \in \overline{K}$ and $D(\overline{sw}) \cap \overline{M} \subseteq \overline{K}$. It follows that $s \in \overline{K}$ and $D(\overline{s}) \cap \overline{M} \subseteq \overline{K}$, namely $s \in F(K)$. This shows that $\overline{F(K)} \subseteq F(K)$; the other direction $\overline{F(K)} \supseteq F(K)$ is automatic.

Next, we show that $F(K) = \overline{K}$ if and only if $D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$. (If) Suppose that $D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$. Let $s \in \overline{K}$; it will be shown that $D(\overline{s}) \cap \overline{M} \subseteq \overline{K}$. Taking an arbitrary string $t \in D(\overline{s}) \cap \overline{M}$, by $s \in \overline{K}$ we derive $t \in D(\overline{K}) \cap \overline{M}$. This shows that $s \in F(K)$ by Eq. 8, and hence $\overline{K} \subseteq F(K)$. The other direction $F(K) \subseteq \overline{K}$ is automatic.

(Only if) Suppose that $F(K) = \overline{K}$. In what follows it will be shown that $D(F(K)) \cap \overline{M} \subseteq F(K)$, which is equivalent to $D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$. Let $s \in D(F(K)) \cap \overline{M}$. As in the proof of Proposition 1 (item 2), we know that $s \neq \epsilon$. So let $s = t\sigma$ for some $t \in \Sigma^*$ and $\sigma \in \Sigma$. Then

$$\begin{aligned} s \in D(F(K)) \cap \overline{M} &\Rightarrow t\sigma \in [F(K) \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \cap \overline{M} \\ &\Rightarrow (\exists t' \in \overline{C}) P(t) = P(t'), t'\sigma \in F(K) \\ &\Rightarrow D(\overline{t'\sigma}) \cap \overline{M} \subseteq \overline{K} \quad \text{(by definition of } F(K)). \end{aligned}$$

Then by Eq. 6

$$\bigcup \left\{ [\overline{t'\sigma} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma \right\} \cap \overline{M} \subseteq \overline{K}.$$

Since $t\sigma$ belongs to the left-hand-side of the above inequality, we have $t\sigma \in \overline{K} = F(K)$. Therefore $D(F(K)) \cap \overline{M} \subseteq F(K)$; equivalently $D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$. □

Now define an operator $\Omega : Pwr(\Sigma^*) \to Pwr(\Sigma^*)$ according to

$$\Omega(K) := \sup \mathcal{N}\left(K \cap F(K),\ \overline{C} \cap M\right), \quad K \in Pwr(\Sigma^*). \tag{9}$$

A language K such that $K = \Omega(K)$ is called a *fixpoint* of the operator Ω. The following proposition characterizes sup $\mathcal{O}(C)$ as the *largest* fixpoint of Ω.

Proposition 3 $\sup \mathcal{O}(C) = \Omega(\sup \mathcal{O}(C))$, *and* $\sup \mathcal{O}(C) \supseteq K$ *for every* K *such that* $K = \Omega(K)$.

To prove Proposition 3, it is useful to note the following. Let $\Omega_1, \Omega_2 : Pwr(\Sigma^*) \to Pwr(\Sigma^*)$ be defined as

$$\begin{aligned} \Omega_1(K) &:= K \cap F(K) \\ \Omega_2(K) &:= \sup \mathcal{N}\left(K,\ \overline{C} \cap M\right). \end{aligned}$$

Then $\Omega(K) = \Omega_2(\Omega_1(K)) = \Omega_2 \circ \Omega_1(K)$, i.e. Ω in Eq. 9 is the composition of Ω_1 and Ω_2. By Lemma 2 (the second statement), it is easily checked that

$$K = \Omega_1(K) \Leftrightarrow D(\overline{K}) \cap \overline{M} \subseteq \overline{K};$$

namely K is a fixpoint of Ω_1 if and only if K satisfies the first characterizing condition (i′) of C-observablity. In addition, by definition of Ω_2 we have

$$K = \Omega_2(K) \Leftrightarrow [K] \cap \left(\overline{C} \cap M\right) = K;$$

namely K is a fixpoint of Ω_2 if and only if K satisfies the second characterizing condition (ii′) of C-observablity. The above two statements together imply that

$$\begin{aligned} K = \Omega(K) \;&\Leftrightarrow\; D(\overline{K}) \cap \overline{M} \subseteq \overline{K} \;\&\; [K] \cap \left(\overline{C} \cap M\right) = K \\ &\Leftrightarrow\; K \in \mathcal{O}(C) \end{aligned}$$

which means that K is a fixpoint of Ω if and only if K is C-observable. With this, the proof of Proposition 3 follows immediately.

Proof of Proposition 3. Since $\sup \mathcal{O}(C) \in \mathcal{O}(C)$, it holds that $\sup \mathcal{O}(C) = \Omega(\sup \mathcal{O}(C))$. Moreover, let K be such that $K = \Omega(K)$. Then $K \in \mathcal{O}(C)$, and therefore $K \subseteq \sup \mathcal{O}(C)$. □

In view of Proposition 3, it is natural to attempt to compute $\sup \mathcal{O}(C)$ by iteration of Ω as follows:

$$(\forall j \geq 1)\; K_j = \Omega(K_{j-1}), \quad K_0 = C. \tag{10}$$

It is readily verified that $\Omega(K) \subseteq K$; hence

$$K_0 \supseteq K_1 \supseteq K_2 \supseteq \cdots$$

Namely the sequence $\{K_j\}$ $(j \geq 1)$ is a monotone (descending) sequence of languages. This implies that the (set-theoretic) limit

$$K_\infty := \lim_{j \to \infty} K_j = \bigcap_{j=0}^{\infty} K_j \tag{11}$$

exists. The following result asserts that if K_∞ is reached in a *finite* number of steps, then K_∞ is precisely the supremal relatively observable sublanguage of C, i.e. $\sup \mathcal{O}(C)$.

Proposition 4 *If* K_∞ *in* Eq. 11 *is reached in a finite number of steps, then*

$$K_\infty = \sup \mathcal{O}(C).$$

Proof Since K_∞ is the limit of the monotone sequence $\{K_j\}$ $(j \geq 1)$, for every fixpoint K of Ω, i.e. $K = \Omega(K)$, there holds $K_\infty \supseteq K$. In particular $K_\infty \supseteq \sup \mathcal{O}(C)$, for $\sup \mathcal{O}(C)$ is a fixpoint of Ω.

Now suppose that the limit K_∞ is reached in a finite number of steps. Then $K_\infty = \Omega(K_\infty)$, and hence $K_\infty \in \mathcal{O}(C)$. This shows that $K_\infty \subseteq \sup \mathcal{O}(C)$, and completes the proof. □

In the next section, we shall establish that, when the given languages M and C are *regular*, the limit K_∞ in Eq. 11 is indeed reached in a finite number of steps.

3 Effective computation of $\sup \mathcal{O}(C)$ in the regular case

In this section, we first review the concept of Nerode equivalence relation and a finite convergence result for a sequence of regular languages. Based on these, we prove that the sequence generated by Eq. 10 converges to the supremal relatively observable sublanguage $\sup \mathcal{O}(C)$ in a finite number of steps. Finally, we show that the computation of $\sup \mathcal{O}(C)$ is effective.

3.1 Preliminaries

Let π be an arbitrary *equivalence relation* on Σ^*. Denote by Σ^*/π the set of *equivalence classes* of π, and write $|\pi|$ for the cardinality of Σ^*/π. Define the *canonical projection* $P_\pi : \Sigma^* \to \Sigma^*/\pi$, namely the surjective function mapping any $s \in \Sigma^*$ onto its equivalence class $P_\pi(s) \in \Sigma^*/\pi$.

Let π_1, π_2 be two equivalence relations on Σ^*. The *partial order* $\pi_1 \leq \pi_2$ holds if

$$(\forall s_1, s_2 \in \Sigma^*)\ s_1 \equiv s_2 (\mathrm{mod}\ \pi_1) \Rightarrow s_1 \equiv s_2 (\mathrm{mod}\ \pi_2).$$

The *meet* $\pi_1 \wedge \pi_2$ is defined by

$$(\forall s_1, s_2 \in \Sigma^*)\ s_1 \equiv s_2 (\mathrm{mod}\ \pi_1 \wedge \pi_2) \text{ iff } s_1 \equiv s_2 (\mathrm{mod}\ \pi_1)\ \&\ s_1 \equiv s_2 (\mathrm{mod}\ \pi_2).$$

For a language $L \subseteq \Sigma^*$, write $\mathrm{Ner}(L)$ for the *Nerode equivalence relation* (Hopcroft and Ullman 1979) on Σ^* with respect to L; namely for all $s_1, s_2 \in \Sigma^*$, $s_1 \equiv s_2 (\mathrm{mod}\ \mathrm{Ner}(L))$ provided

$$(\forall w \in \Sigma^*)\ s_1 w \in L \Leftrightarrow s_2 w \in L.$$

Write $||L||$ for the cardinality of the set of equivalence classes of $\mathrm{Ner}(L)$, i.e. $||L|| := |\mathrm{Ner}(L)|$. The language L is said to be *regular* (Hopcroft and Ullman 1979) if $||L|| < \infty$. Henceforth, we assume that the given languages M and C are regular.

An equivalence relation ρ is a *right congruence* on Σ^* if

$$(\forall s_1, s_2, t \in \Sigma^*)\ s_1 \equiv s_2 (\mathrm{mod}\ \rho) \Rightarrow s_1 t \equiv s_2 t (\mathrm{mod}\ \rho).$$

Any Nerode equivalence relation is a right congruence. For a right congruence ρ and languages $L_1, L_2 \subseteq \Sigma^*$, we say that L_1 is *ρ-supported on* L_2 (Wonham 2016, Section 2.8) if $\overline{L_1} \subseteq \overline{L_2}$ and

$$\{\overline{L_1}, \Sigma^* - \overline{L_1}\} \wedge \rho \wedge \mathrm{Ner}(L_2) \leq \mathrm{Ner}(L_1). \tag{12}$$

Here $\{\overline{L_1}, \Sigma^* - \overline{L_1}\}$ is the equivalence relation on Σ^* with two equivalence classes: $\overline{L_1}$ and $\Sigma^* - \overline{L_1}$. The ρ-support relation is *transitive*: namely, if L_1 is ρ-supported on L_2, and L_2 is ρ-supported on L_3, then L_1 is ρ-supported on L_3. The following lemma is central to establish finite convergence of a monotone language sequence.

Lemma 5 (Wonham 2016, Theorem 2.8.11) *Given a monotone sequence of languages $K_0 \supseteq K_1 \supseteq K_2 \supseteq \cdots$ with K_0 regular, and a fixed right congruence ρ on Σ^* with $|\rho| < \infty$, suppose that K_j is ρ-supported on K_{j-1} for all $j \geq 1$. Then each K_j is regular, and the sequence is finitely convergent to a sublanguage K. Furthermore, K is ρ-supported on K_0 and*

$$||K|| \leq |\rho| \cdot ||K_0|| + 1.$$

In view of this lemma, to show finite convergence of the sequence in Eq. 10, it suffices to find a fixed right congruence ρ with $|\rho| < \infty$ such that K_j is ρ-supported on K_{j-1} for all $j \geq 1$. To this end, we need the following notation.

Let $\mu := \mathrm{Ner}(M)$, $\eta := \mathrm{Ner}(C)$ be Nerode equivalence relations and

$$\varphi_j := \{F(K_j), \Sigma^* - F(K_j)\},\ \kappa_j := \{\overline{K_j}, \Sigma^* - \overline{K_j}\} \quad (j \geq 1)$$

also stand for the equivalence relations corresponding to these partitions. Then $|\mu| < \infty$, $|\eta| < \infty$, and $|\varphi_j| = |\kappa_j| = 2$. Let π be an equivalence relation on Σ^*, and define $f_\pi : \Sigma^* \to \mathrm{Pwr}(\Sigma^*/\pi)$ according to

$$(\forall s \in \Sigma^*)\ f_\pi(s) = \{P_\pi(s') \mid s' \in [s] \cap (\overline{C} \cap M)\} \tag{13}$$

where $[s] = P^{-1}P(\{s\})$. Denote by $\ker f_\pi$ the *equivalence kernel* of f_π, i.e.

$$(\forall s, s' \in \Sigma^*)\; s \equiv s' (\mathrm{mod} \ker f_\pi) \text{ iff } f_\pi(s) = f_\pi(s').$$

Write $\wp(\pi) := \ker f_\pi$. Thus $\wp(\pi)$ is an equivalence relation on Σ^* described as follows: two strings are equivalent mod $\wp(\pi)$ precisely when they have the same subsets of equivalence classes of π containing all the respective lookalike strings. For this reason, $\wp(\pi)$ may be viewed as the 'exponential of π with respect to the natural projection P'. The size of $\wp(\pi)$ is $|\wp(\pi)| \leq 2^{|\pi|}$ (Wonham 2016, Ex. 1.4.21). Another property of $\wp(\cdot)$ we shall use later is (Wonham 2016, Ex. 1.4.21):

$$\wp(\pi_1 \wedge \wp(\pi_2)) = \wp(\pi_1 \wedge \pi_2) = \wp(\wp(\pi_1) \wedge \pi_2)$$

where π_1, π_2 are equivalence relations on Σ^*.

3.2 Convergence result

First, we present a key result on the support relation of the sequence $\{K_j\}$ generated by Eq. 10.

Proposition 6 *Consider the sequence* $\{K_j\}$ *generated by* Eq. 10*. For each* $j \geq 1$*, there holds that* K_j *is* ρ*-supported on* K_{j-1}*, where*

$$\rho := \mu \wedge \eta \wedge \wp(\mu \wedge \eta). \tag{14}$$

The equivalence classes of ρ in Eq. 14 are formed by intersecting those of μ, η, and $\wp(\mu \wedge \eta)$. Namely ρ partitions Σ^* into cells of strings that are simultaneously in the same cells of μ, η, and $\wp(\mu \wedge \eta)$. Let us postpone the proof of Proposition 6, and immediately present our main result.

Theorem 7 *Consider the sequence* $\{K_j\}$ *generated by* Eq. 10*, and suppose that the given languages* M *and* C *are regular. Then the sequence* $\{K_j\}$ *is finitely convergent to* $\sup \mathcal{O}(C)$*, and* $\sup \mathcal{O}(C)$ *is a regular language with*

$$||\sup \mathcal{O}(C)|| \leq ||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1.$$

Proof Let $\rho = \mu \wedge \eta \wedge \wp(\mu \wedge \eta)$ as in Eq. 14. Since μ and η are right congruences, so are $\mu \wedge \eta$ and $\wp(\mu \wedge \eta)$ (Wonham 2016, Exercise 6.1.25). Hence ρ is a right congruence, with

$$\begin{aligned} |\rho| &\leq |\mu| \cdot |\eta| \cdot 2^{|\mu| \cdot |\eta|} \\ &= ||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||}. \end{aligned}$$

Since the languages M and C are regular, i.e. $||M||, ||C|| < \infty$, we derive that $|\rho| < \infty$.

It then follows from Lemmas 4, 5 and Proposition 6 that the sequence $\{K_j\}$ is finitely convergent to $\sup \mathcal{O}(C)$, and $\sup \mathcal{O}(C)$ is ρ-supported on K_0, i.e.

$$\begin{aligned} \mathrm{Ner}(\sup \mathcal{O}(C)) &\geq \{\overline{\sup \mathcal{O}(C)}, \Sigma^* - \overline{\sup \mathcal{O}(C)}\} \wedge \rho \wedge \mathrm{Ner}(K_0) \\ &= \{\overline{\sup \mathcal{O}(C)}, \Sigma^* - \overline{\sup \mathcal{O}(C)}\} \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta) \wedge \mathrm{Ner}(K_0) \\ &= \{\overline{\sup \mathcal{O}(C)}, \Sigma^* - \overline{\sup \mathcal{O}(C)}\} \wedge \mu \wedge \wp(\mu \wedge \eta) \wedge \mathrm{Ner}(K_0). \end{aligned}$$

Hence $\sup \mathcal{O}(C)$ is in fact $(\mu \wedge \wp(\mu \wedge \eta))$-supported on K_0, which implies

$$\begin{aligned} ||\sup \mathcal{O}(C)|| &\leq |\mu \wedge \wp(\mu \wedge \eta)| \cdot ||K_0|| + 1 \\ &\leq ||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1 < \infty. \end{aligned}$$

Therefore $\sup \mathcal{O}(C)$ is itself a regular language. □

Theorem 7 establishes the finite convergence of the sequence $\{K_j\}$ in Eq. 10, as well as the fact that an upper bound of $||\sup \mathcal{O}(C)||$ is exponential in the product of $||M||$ and $||C||$.[1]

Remark 1 The limit $\sup \mathcal{O}(C)$ is reached in no more than $||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1$ steps. To see this, let

$$\pi_1 := \{\overline{K_1}, \Sigma^* - \overline{K_1}\} \wedge \rho \wedge \mathrm{Ner}(K_0).$$

Then $|\pi_1| = ||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1$ (similar to the proof of Theorem 7). Since $\mathrm{Ner}(K_1) \geq \pi_1$, it may be verified that π_1 is a right congruence; hence there is an automaton representation of π_1 which is a *recognizer* for K_1 (Wonham 2016). Now let $K_2 = \Omega(K_1) \subseteq K_1$; by transitivity of the ρ-support relation we derive

$$\mathrm{Ner}(K_2) \geq \{\overline{K_2}, \Sigma^* - \overline{K_2}\} \wedge \rho \wedge \mathrm{Ner}(K_0) =: \pi_2.$$

Thus π_2 is a right congruence, and there is an automaton representation of π_2 which is a recognizer for K_2. Moreover, the above inequality implies that π_1, when restricted to $\overline{K_2}$, is finer than $\mathrm{Ner}(K_2)$. Hence in passing from K_1 to K_2, it is only necessary to remove those π_1-cells in $\overline{K_1}$ that are not in $\overline{K_2}$. Therefore $|\pi_2| \leq |\pi_1|$. Inductively one can show that $|\pi_{j+1}| \leq |\pi_j|$ for all $j \geq 1$, where

$$\pi_j := \{\overline{K_j}, \Sigma^* - \overline{K_j}\} \wedge \rho \wedge \mathrm{Ner}(K_0) \; (\leq \mathrm{Ner}(K_j))$$

and there is an automaton representation of π_j which is a recognizer for K_j. With this, and the fact that $|\pi_1| = ||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1$, we conclude that the sequence $\{|\pi_j|\}$, and therefore $\{K_j\}$, converges in at most $||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1$ steps.

In the sequel we prove Proposition 6, for which we need two lemmas.

Lemma 8 *For each $j \geq 1$, the Nerode equivalence relation on Σ^* with respect to $F(K_{j-1})$ satisfies*

$$Ner(F(K_{j-1})) \geq \varphi_j \wedge Ner(K_{j-1}) \wedge \wp(Ner(K_{j-1}) \wedge \mu \wedge \eta).$$

Proof Let $s_1, s_2 \in \Sigma^*$ and assume s_1, s_2 are equivalent mod the equivalence relation on the right-hand-side. Since $s_1 \equiv s_2 (\mathrm{mod}\, \varphi_j)$, either $s_1, s_2 \in \Sigma^* - F(K_{j-1})$ or $s_1, s_2 \in F(K_{j-1})$. First, let $s_1, s_2 \in \Sigma^* - F(K_{j-1})$; then for all $w \in \Sigma^*$ it holds that $s_1 w, s_2 w \in \Sigma^* - F(K_{j-1})$. Thus $s_1 \equiv s_2 (\mathrm{mod}\, \mathrm{Ner}(F(K_{j-1})))$.

Next, let $s_1, s_2 \in F(K_{j-1})$ and assume that

$$s_1 \equiv s_2 (\mathrm{mod}\, \mathrm{Ner}(K_{j-1}) \wedge \wp(\mathrm{Ner}(K_{j-1}) \wedge \mu \wedge \eta)).$$

Also let $w \in \Sigma^*$ be such that $s_1 w \in F(K_{j-1})$. It will be shown that $s_2 w \in F(K_{j-1})$. Note first that $s_2 w \in \overline{K_{j-1}}$, since $s_1 w \in F(K_{j-1}) \subseteq \overline{K_{j-1}}$ and $s_1 \equiv s_2 (\mathrm{mod}\, \mathrm{Ner}(K_{j-1}))$. Hence it is left to show that $D(\overline{s_2 w}) \cap \overline{M} \subseteq \overline{K_{j-1}}$, i.e.

$$\bigcup \left\{ [\overline{s_2 w} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma \right\} \cap \overline{M} \subseteq \overline{K_{j-1}}.$$

[1] For regular languages C, M such that $C \subseteq M$, one may always find finite-state automata, say **C**, **M**, such that **C** is a *subautomaton* of **M** (e.g. Cho and Marcus (1989)). Then the state size of the product of **C** and **M** is simply the state size of **C**. Therefore the automaton representing $\sup \mathcal{O}(C)$ has state size upper bounded by the exponential in the state size of **C**.

It follows from $s_2 \in F(K_{j-1})$ that

$$\bigcup \left\{ [\overline{s_2} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma \right\} \cap \overline{M} \subseteq \overline{K_{j-1}}.$$

Thus let $s_2' \in [s_2]$, $x' \in [\overline{w}]$, and $s_2'x' \in [\overline{s_2 w} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \cap \overline{M}$ for some $\sigma \in \Sigma$. Write $x' := y'\sigma$, $y' \in \Sigma^*$. Since $s_1 \equiv s_2 (\text{mod}\ \wp(\text{Ner}(K_{j-1}) \wedge \mu \wedge \eta))$, there exists $s_1' \in [s_1]$ such that $s_1' \equiv s_2' (\text{mod Ner}(K_{j-1}) \wedge \mu \wedge \eta)$. Hence $s_1'x' \in \overline{M}$ and $s_1'y' \in \overline{C}$, and we derive that $s_1'x' = s_1'y'\sigma \in [\overline{\{s_1 w\}} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \cap \overline{M}$. It then follows from $s_1 w \in F(K_{j-1})$ that $s_1'x' \in \overline{K_{j-1}}$, which in turn implies that $s_2'x' \in \overline{K_{j-1}}$. This completes the proof that $s_2 w \in F(K_{j-1})$, as required. □

Lemma 9 *For K_j ($j \geq 1$) generated by* Eq. 10, *the following statements hold:*

$$K_j = \bigcup \left\{ [s] \cap \left(\overline{C} \cap M\right) \mid s \in \Sigma^* \ \&\ [s] \cap \left(\overline{C} \cap M\right) \subseteq K_{j-1} \cap F(K_{j-1}) \right\};$$
$$Ner(K_j) \geq \mu \wedge \eta \wedge \wp(Ner(K_{j-1}) \wedge Ner(F(K_{j-1})) \wedge \mu \wedge \eta).$$

Proof By Eq. 9 we know that K_j is the supremal normal sublanguage of $K_{j-1} \cap F(K_{j-1})$ with respect to $\overline{C} \cap M$. Thus the conclusions follow immediately from Exercise 6.1.25 of Wonham (2016). □

Now we are ready to prove Proposition 6.

Proof of Proposition 6. To prove that K_j is ρ-supported on K_{j-1} ($j \geq 1$), by definition we must show that

$$\text{Ner}(K_j) \geq \kappa_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta) \wedge \text{Ner}(K_{j-1}).$$

It suffices to show the following:

$$\text{Ner}(K_j) \geq \kappa_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).$$

We prove this statement by induction. First, we show the base case ($j = 1$)

$$\text{Ner}(K_1) \geq \kappa_1 \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).$$

From Lemma 8 and $K_0 = C$ (thus $\text{Ner}(K_0) = \eta$) we have

$$\begin{aligned} \text{Ner}(F(K_0)) &\geq \varphi_1 \wedge \text{Ner}(K_0) \wedge \wp(\text{Ner}(K_0) \wedge \mu \wedge \eta) \\ &= \varphi_1 \wedge \eta \wedge \wp(\mu \wedge \eta). \end{aligned}$$

It then follows from Lemma 9 that

$$\begin{aligned} \text{Ner}(K_1) &\geq \mu \wedge \eta \wedge \wp(\text{Ner}(K_0) \wedge \text{Ner}(F(K_0)) \wedge \mu \wedge \eta) \\ &\geq \mu \wedge \eta \wedge \wp(\eta \wedge \varphi_1 \wedge \eta \wedge \wp(\mu \wedge \eta) \wedge \mu \wedge \eta) \\ &= \mu \wedge \eta \wedge \wp(\varphi_1 \wedge \mu \wedge \eta) \wedge \wp(\mu \wedge \eta) \\ &= \mu \wedge \eta \wedge \wp(\varphi_1 \wedge \mu \wedge \eta). \end{aligned} \tag{15}$$

We claim that

$$\text{Ner}(K_1) \geq \kappa_1 \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).$$

To show this, let $s_1, s_2 \in \Sigma^*$ and assume that $s_1 \equiv s_2 (\text{mod}\ \kappa_1 \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta))$. If $s_1, s_2 \in \Sigma^* - \overline{K_1}$, then for all $w \in \Sigma^*$, $s_1 w, s_2 w \in \Sigma^* - \overline{K_1}$; thus $s_1 \equiv s_2 (\text{mod Ner}(K_1))$. Now let $s_1, s_2 \in \overline{K_1}$. By Lemma 9 we derive that for all $s_1' \in [s_1] \cap \left(\overline{C} \cap M\right)$ and $s_2' \in [s_2] \cap \left(\overline{C} \cap M\right)$, $s_1', s_2' \in \overline{K_1}$. Since $\overline{K_1} \subseteq F(K_0)$, $s_1', s_2' \in F(K_0)$ and hence

$$\{P_{\varphi_1 \wedge \mu \wedge \eta}(s_1') \mid s_1' \in [s_1] \cap \left(\overline{C} \cap M\right)\} = \{P_{\varphi_1 \wedge \mu \wedge \eta}(s_2') \mid s_2' \in [s_2] \cap \left(\overline{C} \cap M\right)\}.$$

Namely $s_1 \equiv s_2 (\mathrm{mod}\ \wp(\varphi_1 \wedge \mu \wedge \eta))$. This implies that $s_1 \equiv s_2 (\mathrm{mod}\ \mathrm{Ner}(K_1))$ by Eq. 15. Hence the above claim is established, and the base case is proved.

For the induction step, suppose that for $j \geq 2$, there holds

$$\mathrm{Ner}(K_{j-1}) \geq \kappa_{j-1} \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).$$

Again by Lemma 8 we have

$$\begin{aligned}\mathrm{Ner}(F(K_{j-1})) &\geq \varphi_{j-1} \wedge \mathrm{Ner}(K_{j-1}) \wedge \wp(\mathrm{Ner}(K_{j-1}) \wedge \mu \wedge \eta)\\ &\geq \varphi_{j-1} \wedge \kappa_{j-1} \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta) \wedge \wp(\kappa_{j-1} \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta) \wedge \mu \wedge \eta)\\ &= \varphi_{j-1} \wedge \kappa_{j-1} \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta) \wedge \wp(\kappa_{j-1} \wedge \mu \wedge \eta)\\ &= \varphi_{j-1} \wedge \kappa_{j-1} \wedge \mu \wedge \eta \wedge \wp(\kappa_{j-1} \wedge \mu \wedge \eta)\end{aligned}$$

Then by Lemma 9,

$$\begin{aligned}\mathrm{Ner}(K_j) &\geq \mu \wedge \eta \wedge \wp(\mathrm{Ner}(K_{j-1}) \wedge \mathrm{Ner}(F(K_{j-1})) \wedge \mu \wedge \eta)\\ &\geq \mu \wedge \eta \wedge \wp(\varphi_{j-1} \wedge \kappa_{j-1} \wedge \mu \wedge \eta \wedge \wp(\kappa_{j-1} \wedge \mu \wedge \eta))\\ &= \mu \wedge \eta \wedge \wp(\varphi_{j-1} \wedge \kappa_{j-1} \wedge \mu \wedge \eta).\end{aligned} \tag{16}$$

We claim that

$$\mathrm{Ner}(K_j) \geq \kappa_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).$$

To show this, let $s_1, s_2 \in \Sigma^*$ and assume that $s_1 \equiv s_2 (\mathrm{mod}\ \kappa_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta))$. If $s_1, s_2 \in \Sigma^* - \overline{K_j}$, then for all $w \in \Sigma^*$, $s_1 w, s_2 w \in \Sigma^* - \overline{K_j}$; hence $s_1 \equiv s_2 (\mathrm{mod}\ \mathrm{Ner}(K_j))$. Now let $s_1, s_2 \in \overline{K_j}$. By Lemma 9 we derive that for all $s_1' \in [s_1] \cap \left(\overline{C} \cap M\right)$ and $s_2' \in [s_2] \cap \left(\overline{C} \cap M\right)$, $s_1', s_2' \in \overline{K_j}$. Since $\overline{K_j} \subseteq F(K_{j-1}) \subseteq \overline{K_{j-1}}$,

$$\begin{aligned}&\{P_{\varphi_{j-1} \wedge \kappa_{j-1} \wedge \mu \wedge \eta}(s_1') \mid s_1' \in [s_1] \cap \left(\overline{C} \cap M\right)\}\\ &= \{P_{\varphi_{j-1} \wedge \kappa_{j-1} \wedge \mu \wedge \eta}(s_2') \mid s_2' \in [s_2] \cap \left(\overline{C} \cap M\right)\}.\end{aligned}$$

Namely $s_1 \equiv s_2 (\mathrm{mod}\ \wp(\varphi_{j-1} \wedge \kappa_{j-1} \wedge \mu \wedge \eta)$. This implies that $s_1 \equiv s_2 (\mathrm{mod}\ \mathrm{Ner}(K_j))$ by Eq. 16. Therefore the above claim is established, and the induction step is completed. □

3.3 Effective computability of Ω

We conclude this section by showing that the iteration scheme in Eq. 10 yields an effective procedure for the computation of $\sup \mathcal{O}(C)$, when the given languages M and C are regular. For this, owing to Theorem 7, it suffices to prove that the operator Ω in Eq. 9 is effectively computable.

Recall that a language $L \subseteq \Sigma^*$ is regular if and only if there exists a finite-state automaton $\mathbf{G} = (Q, \Sigma, \delta, q_0, Q_m)$ such that

$$L_m(\mathbf{G}) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\} = L.$$

Let $\mathcal{O} : (Pwr(\Sigma^*))^k \to (Pwr(\Sigma^*))$ be an operator that preserves regularity; namely $L_1, ..., L_k$ regular implies $\mathcal{O}(L_1, ..., L_k)$ regular. We say that $\mathcal{O}$ is *effectively computable* if from each k-tuple $(L_1, ..., L_k)$ of regular languages, one can construct a finite-state automaton $\mathbf{G}$ with $L_m(\mathbf{G}) = \mathcal{O}(L_1, ..., L_k)$.

The standard operators of language closure, complement,[2] union, intersection, and catenation all preserve regularity and are effectively computable (Eilenberg 1974). Moreover, both the operator $\sup \mathcal{N} : Pwr(\Sigma^*) \to Pwr(\Sigma^*)$ given by

$$\sup \mathcal{N}(L) := \bigcup \{L' \subseteq L \mid [L'] \cap H = L'\}, \quad \text{with respect to some fixed } H \subseteq \Sigma^*$$

and the operator $\sup \mathcal{F} : Pwr(\Sigma^*) \to Pwr(\Sigma^*)$ given by

$$\sup \mathcal{F}(L) := \bigcup \{L' \subseteq L \mid \overline{L'} = L'\}$$

preserve regularity and are effectively computable (see Cho and Marcus (1989) and Wonham and Ramadge (1987), respectively).

The main result of this subsection is the following theorem.

Theorem 10 *Consider the operator* Ω *in* Eq. 9 *and let* $K \subseteq \Sigma^*$. *Then*

$$\Omega(K) = \sup \mathcal{N} \left(K \cap \sup \mathcal{F} \left(\bigcap \{\sup \mathcal{N}(\overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c) \cup (\overline{C}.\sigma)^c \mid \sigma \in \Sigma\} \right) \right)$$

where $\sup \mathcal{N}$ *is with respect to* $\overline{C} \cap M$.

Theorem 10 implies that the operator Ω in Eq. 9 preserves regularity and is effectively computable, inasmuch as the language operations involved – closure, complement, union, intersection, catenation $\overline{C}.\sigma$, $\sup \mathcal{N}$, and $\sup \mathcal{F}$ – all preserve regularity and are effectively computable. Moreover, there exist off-the-shelf algorithms for these language operations, which allows straightforward implementation of Ω.

Remark 2 We have shown in Remark 1 that the sequence $\{K_j\}$ in Eq. 10 converges in at most $||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1$ steps. We have also considered that K_j is recognized by an automaton, say $\mathbf{G}_j$, that represents the following right congruence

$$\pi_j = \{\overline{K_j}, \Sigma^* - \overline{K_j}\} \wedge \rho \wedge \text{Ner}(K_0) \; (\leq \text{Ner}(K_j)), \quad \text{where } \rho = \mu \wedge \eta \wedge \wp(\mu \wedge \eta).$$

Owing to the exponential property of ρ, the state structure of $\mathbf{G}_j$ is already exponential in $||M|| \cdot ||C||$ and thus applying $\sup \mathcal{N}$ on $\mathbf{G}_j$ causes no further exponentiation. It is then verified that by the formula given in Theorem 10, the complexity of computing $\Omega(K_j)$, namely the complexity of each step in generating the sequence $\{K_j\}$, is $O(|\Sigma| \cdot ||M||^5 \cdot ||C||^6 \cdot 2^{2||M|| \cdot ||C||})$. In total therefore, the complexity of the language iteration scheme (10) is $O(|\Sigma| \cdot ||M||^6 \cdot ||C||^7 \cdot 2^{3||M|| \cdot ||C||})$, which improves on the algorithm of double-exponential complexity in Cai et al. (2015). The order estimate has the same exponential term as that of the automaton-based algorithm in Alves et al. (2016), but includes a higher-degree polynomial multiplier. Nevertheless, the advantage of our algorithm is that it is decomposable into a set of well-known language computations and thus admits easy implementation by off-the-shelf algorithms. We leave for future research the design of more efficient language-based algorithms.

Remark 3 In the automaton-based algorithm in Cai et al. (2015), the operations at each iteration are: (1) search for transitions and marker states (of the automaton representing the language $M \cap C \cap P(M \cap C)$) that violate the conditions of relative observability, and (2) remove such transitions, unmark such marker states, trim the resulting automaton and proceed to the next iteration. These operations provide little linguistic insight and were

[2] For a language $L \subseteq \Sigma^*$, its complement, written L^c, is $\Sigma^* - L$.

difficult to implement. By contrast, the computation of Ω at each iteration of the language-based algorithm is decomposable into several well-known language operations (Theorem 10), which are not only distinct from those of the algorithm in (Cai et al. 2015), but are also easily implementable by off-the-shelf algorithms.

Proof of Theorem 10. First, we claim that for each $K \subseteq \Sigma^*$,

$$F(K) = \overline{K} \cap \sup \mathcal{F}\left(\bigcap\{\sup \mathcal{N}(\overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c) \cup (\overline{C}.\sigma)^c \mid \sigma \in \Sigma\}\right).$$

To see this, we derive by Eqs. 8 and 6 that

$$F(K) = \{s \in \overline{K} \mid \bigcup\left\{[\overline{s} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma\right\} \cap \overline{M} \subseteq \overline{K}\}.$$

Hence

$$\begin{aligned}
s \in F(K) &\Leftrightarrow s \in \overline{K} \text{ and } \bigcup\left\{[\overline{s} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma\right\} \cap \overline{M} \subseteq \overline{K} \\
&\Leftrightarrow s \in \overline{K} \text{ and } \bigcup\left\{[\overline{s} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma\right\} \subseteq \overline{K} \cup (\overline{M})^c \\
&\Leftrightarrow s \in \overline{K} \text{ and } (\forall \sigma \in \Sigma)\ [\overline{s} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \subseteq \overline{K} \cup (\overline{M})^c \\
&\Leftrightarrow s \in \overline{K} \text{ and } (\forall \sigma \in \Sigma)\ [\overline{s} \cap \overline{C}.\sigma] \subseteq \overline{K} \cup (\overline{M})^c \cup (\overline{C}.\sigma)^c \\
&\Leftrightarrow s \in \overline{K} \text{ and } (\forall \sigma \in \Sigma)\ [\overline{s} \cap \overline{C}.\sigma] \subseteq \overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c \\
&\Leftrightarrow s \in \overline{K} \text{ and } (\forall \sigma \in \Sigma)\ \overline{s} \cap \overline{C}.\sigma \subseteq \sup \mathcal{N}(\overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c) \\
&\Leftrightarrow s \in \overline{K} \text{ and } (\forall \sigma \in \Sigma)\ \overline{s} \subseteq \sup \mathcal{N}(\overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c) \cup (\overline{C}.\sigma)^c \\
&\Leftrightarrow s \in \overline{K} \text{ and } \overline{s} \subseteq \bigcap\{\sup \mathcal{N}(\overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c) \cup (\overline{C}.\sigma)^c \mid \sigma \in \Sigma\} \\
&\Leftrightarrow s \in \overline{K} \text{ and } s \in \sup \mathcal{F}\left(\bigcap\{\sup \mathcal{N}(\overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c) \cup (\overline{C}.\sigma)^c \mid \sigma \in \Sigma\}\right) \\
&\Leftrightarrow s \in \overline{K} \cap \sup \mathcal{F}\left(\bigcap\{\sup \mathcal{N}(\overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c) \cup (\overline{C}.\sigma)^c \mid \sigma \in \Sigma\}\right).
\end{aligned}$$

This proves the claim. It follows immediately from the definition of Ω that

$$\Omega(K) = \sup \mathcal{N}\left(K \cap \sup \mathcal{F}\left(\bigcap\{\sup \mathcal{N}(\overline{K} \cup (\overline{M} \cap \overline{C}.\sigma)^c) \cup (\overline{C}.\sigma)^c \mid \sigma \in \Sigma\}\right)\right)$$

as required. □

4 Relative observability and controllability

For the purpose of supervisory control under partial observation, we combine relative observability with *controllability* and provide a fixpoint characterization of the supremal relatively observable and controllable sublanguage.

Let the alphabet Σ be partitioned into Σ_c, the subset of controllable events, and Σ_u, the subset of uncontrollable events. For the given M and C, we say that C is *controllable* with respect to M if

$$\overline{C}\Sigma_u \cap \overline{M} \subseteq \overline{C}.$$

Whether or not C is controllable, write $\mathcal{C}(C)$ for the family of all controllable sublanguages of C. Then the supremal element $\sup \mathcal{C}(C)$ exists and is effectively computable (Wonham and Ramadge 1987; Kumar et al. 1993).

Now write $\mathcal{CO}(C)$ for the family of controllable and C-observable sublanguages of C. Note that the family $\mathcal{CO}(C)$ is nonempty inasmuch as the empty language is a member.

Thanks to the closure under union of both controllability and C-observability, the supremal controllable and C-observable sublanguage $\sup\mathcal{CO}(C)$ exists and is given by

$$\sup\mathcal{CO}(C) := \bigcup\{K \mid K \in \mathcal{CO}(C)\}. \tag{17}$$

Define the operator $\Gamma : Pwr(\Sigma^*) \to Pwr(\Sigma^*)$ by[3]

$$\Gamma(K) := \sup\mathcal{O}(\sup\mathcal{C}(K)). \tag{18}$$

The proposition below characterizes $\sup\mathcal{CO}(C)$ as the largest fixpoint of Γ.

Proposition 11 $\sup\mathcal{CO}(C) = \Gamma(\sup\mathcal{CO}(C))$*, and* $\sup\mathcal{CO}(C) \supseteq K$ *for every* K *such that* $K = \Gamma(K)$.

Proof Since $\sup\mathcal{CO}(C) \in \mathcal{CO}(C)$, i.e. both controllable and C-observable,

$$\begin{aligned}\Gamma(\sup\mathcal{CO}(C)) &= \sup\mathcal{O}(\sup\mathcal{C}(\sup\mathcal{CO}(C)))\\ &= \sup\mathcal{O}(\sup\mathcal{CO}(C))\\ &= \sup\mathcal{CO}(C).\end{aligned}$$

Next let K be such that $K = \Gamma(K)$. To show that $K \subseteq \sup\mathcal{CO}(C)$, it suffices to show that $K \in \mathcal{CO}(C)$. Let $H := \sup\mathcal{C}(K)$; thus $H \subseteq K$. On the other hand, from $K = \Gamma(K) = \sup\mathcal{O}(H)$ we have $K \subseteq H$. Hence $K = H$. It follows that $K = \sup\mathcal{C}(K)$ and $K = \sup\mathcal{O}(K)$, which means that K is both controllable and C-observable. Therefore we conclude that $K \in \mathcal{CO}(C)$. □

In view of Proposition 11, we compute $\sup\mathcal{CO}(C)$ by iteration of Γ as follows:

$$(\forall j \geq 1)\ K_j = \Gamma(K_{j-1}), \quad K_0 = C. \tag{19}$$

It is readily verified that $\Gamma(K) \subseteq K$, and thus

$$K_0 \supseteq K_1 \supseteq K_2 \supseteq \cdots$$

Namely the sequence $\{K_j\}$ $(j \geq 1)$ is a monotone (descending) sequence of languages. Recalling the notation from Section 3.1, we have the following key result.

Proposition 12 *Consider the sequence* $\{K_j\}$ *generated by* Eq. 19 *and let* $\rho = \mu \wedge \eta \wedge \wp(\mu \wedge \eta)$. *Then for each* $j \geq 1$, K_j *is* ρ*-supported on* K_{j-1}.

Proof Write $H_j := \sup\mathcal{C}(K_{j-1})$ and $\psi_j := \{\overline{H_j}, \Sigma^* - \overline{H_j}\}$ for $j \geq 1$. Then by Wonham and Ramadge (1987, p. 642) there holds

$$\text{Ner}(H_j) \geq \psi_j \wedge \mu \wedge \text{Ner}(K_{j-1}).$$

We claim that for $j \geq 1$,

$$\text{Ner}(K_j) \geq \kappa_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).$$

We prove this claim by induction. For the base case ($j = 1$),

$$\begin{aligned}\text{Ner}(H_1) &\geq \psi_1 \wedge \mu \wedge \text{Ner}(K_0)\\ &= \psi_1 \wedge \mu \wedge \eta\end{aligned}$$

[3] As defined Γ is the composition of an operator characterizing the set of controllable sublanguages of C and the operator Ω in Eq. 9 characterizing the set of C-observable sublanguages of C.

Since $K_1 = \sup \mathcal{O}(H_1)$, we adopt the following sequence to compute K_1:

$$(\forall i \geq 1)\ T_i = \Omega(T_{i-1}), \quad T_0 = H_1.$$

Following the derivations in the proof of Proposition 6, it is readily shown that each T_i is ρ-supported on H_1; in particular,

$$\begin{aligned}\text{Ner}(K_1) &\geq \kappa_1 \wedge \rho \wedge \text{Ner}(H_1)\\ &\geq \kappa_1 \wedge \psi_1 \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta)\\ &= \kappa_1 \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).\end{aligned}$$

This confirms the base case.

For the induction step, suppose that for $j \geq 2$, there holds

$$\text{Ner}(K_{j-1}) \geq \kappa_{j-1} \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).$$

Thus

$$\begin{aligned}\text{Ner}(H_j) &\geq \psi_j \wedge \mu \wedge \text{Ner}(K_{j-1})\\ &\geq \psi_j \wedge \kappa_{j-1} \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta)\\ &= \psi_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).\end{aligned}$$

Again adopt a sequence to compute K_j as follows:

$$(\forall i \geq 1)\ T_i = \Omega(T_{i-1}), \quad T_0 = H_j.$$

We derive by similar calculations as in Proposition 6 that each T_i is ρ-supported on H_j; in particular,

$$\begin{aligned}\text{Ner}(K_j) &\geq \kappa_j \wedge \rho \wedge \text{Ner}(H_j)\\ &\geq \kappa_j \wedge \psi_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta)\\ &= \kappa_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta).\end{aligned}$$

Therefore the induction step is completed, and the above claim is established. There follows immediately

$$\begin{aligned}\text{Ner}(K_j) &\geq \kappa_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta) \wedge \text{Ner}(K_{j-1})\\ &= \kappa_j \wedge \rho \wedge \text{Ner}(K_{j-1}).\end{aligned}$$

Namely, K_j is ρ-supported on K_{j-1}, as required. □

The following theorem, the main result of this section, follows directly from Proposition 12 and Lemma 5.

Theorem 13 *Consider the sequence* $\{K_j\}$ *in* Eq. 19*, and suppose that the given languages* M *and* C *are regular. Then the sequence* $\{K_j\}$ *is finitely convergent to* $\sup \mathcal{CO}(C)$*, and* $\sup \mathcal{CO}(C)$ *is a regular language with*

$$||\sup \mathcal{CO}(C)|| \leq ||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1.$$

We conclude that $\sup \mathcal{CO}(C)$ is effectively computable, inasmuch as the operators $\sup \mathcal{C}(\cdot)$ and $\sup \mathcal{O}(\cdot)$ are (see Wonham and Ramadge (1987) and Theorem 10, respectively). In particular, the operator Γ in Eq. 18 is effectively computable. Finally, the limit

$\sup \mathcal{CO}(C)$ is reached in no more than $||M|| \cdot ||C|| \cdot 2^{||M|| \cdot ||C||} + 1$ steps, and the complexity of this iteration scheme (19) is $O(|\Sigma| \cdot ||M||^7 \cdot ||C||^7 \cdot 2^{3||M|| \cdot ||C||})$ (cf. Remarks 1 and 2).

5 Conclusions

We have presented a new characterization of relative observability, and an operator on languages whose largest fixpoint is the supremal relatively observable sublanguage. In the case of regular languages and based on the support relation, we have proved that the sequence of languages generated by the operator converges finitely (albeit with exponential complexity) to the supremal relatively observable sublanguage, and the operator is effectively computable. Moreover, for the purpose of supervisory control under partial observation, we have presented a second operator that in the regular case effectively computes the supremal relatively observable and controllable sublanguage.

Both operators have been implemented and tested on a number of examples. We have confirmed that the computational results agree with those by the algorithm in Cai et al. (2015). Thus the new operators provide a useful alternative to ensure presumed correctness based on consistency.

References

Alves M, Carvalho L, Basilio J (2016) New algorithms for verification of relative observability and computation of supremal relatively observable sublanguage. In: Proceedings of the multi-conference on systems and control. Buenos Aires, Argentina, pp 527–531

Brandt RD, Garg VK, Kumar R, Lin F, Marcus SI, Wonham WM (1990) Formulas for calculating supremal controllable and normal sublanguages. Syst Control Lett 15:111–117

Cai K, Wonham WM (2016) A new algorithm for computing the supremal relatively observable sublanguage. In: Proceedings of the workshop on discrete-event systems. Xi'an, China, pp 8–13

Cai K, Zhang R, Wonham WM (2015) Relative observability of discrete-event systems and its supremal sublanguages. IEEE Trans Autom Control 60:659–670

Cai K, Zhang R, Wonham WM (2016) Relative observability and coobservability of timed discrete-event systems. IEEE Trans Autom Control 61:3382–3395

Cho H, Marcus SI (1989) On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation. Math Control Signals Syst 2:47–69

Cieslak R, Desclaux C, Fawaz AS, Varaiya P (1988) Supervisory control of discrete-event processes with partial observations. IEEE Trans Autom Control 33:249–260

Eilenberg S (1974) Automata, languages and machines, volume A. Academic Press

Hopcroft JE, Ullman JD (1979) Introduction to automata theory. Addison-Wesley, Languages and Computation

Kumar R, Garg VK, Marcus SI (1993) Predicates and predicate transformers for supervisory control of discrete event dynamical systems. IEEE Trans Autom Control 38:232–247

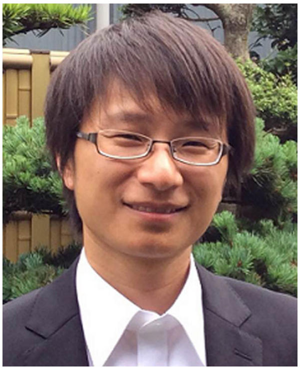
Lin F, Wonham WM (1988) On observability of discrete-event systems. Inform Sci 44:173–198

Moor T, Baier C, Yoo T, Lin F, Lafortune S (2012) On the computation of supremal sublanguages relevant to supervisory control. In: Proceedings of the international workshop discrete event systems (WODES12). Guadalajara, Mexico, pp 175–180

Wonham WM (2016) Supervisory Control of Discrete-Event Systems. Systems Control Group, Dept. of Electrical and Computer Engineering, University of Toronto, updated annually 1998-2016. Available online at http://www.control.toronto.edu/DES

Wonham WM, Ramadge PJ (1987) On the supremal controllable sublanguage of a given language. SIAM J Control Opt 25:637–659

Yin X, Lafortune S (2016) A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. IEEE Trans Autom Control 61:2140–2154

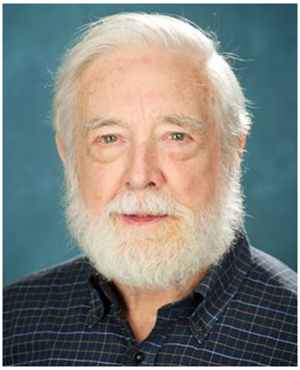


Kai Cai received the B. Eng. degree in Electrical Engineering from Zhejiang University, Hangzhou, China, in 2006; the M.A.Sc. degree in Electrical and Computer Engineering from the University of Toronto, Toronto, ON, Canada, in 2008; and the Ph.D. degree in Systems Science from the Tokyo Institute of Technology, Tokyo, Japan, in 2011. He is currently an Associate Professor at Osaka City University. Previously, he was an Assistant Professor at the University of Tokyo (2013-2014), and a postdoctoral Fellow at the University of Toronto (2011-2013).

Dr. Cai's research interests include distributed control of discrete-event systems and cooperative control of networked multi-agent systems. He is the co-author (with W.M. Wonham) of Supervisor Localization (Springer 2015). He received the Best Paper Award of SCIE in 2013, the Best Student Paper Award of the IEEE Multi-Conference on Systems and Control and the Young Author's Award of SCIE in 2010.

Renyuan Zhang received the B. Eng. degree and the Ph. D degree in Electrical Engineering from Xi'an Jiaotong University, Xi'an, China, in 2007 and 2013 respectively.

He is currently an associate professor at School of Automation, Northwestern Polytechnical University, Xi'an, China. He was an international visiting student in the University of Toronto 2011-2012. His research interests include distributed control of discrete-event systems and hybrid control of urban traffic systems.



W. M. Wonham received the B. Eng. degree in engineering physics from McGill University in 1956, and the Ph.D. in control engineering from the University of Cambridge (U.K.) in 1961.

From 1961 to 1969, he was associated with several U.S. research groups in control. Since 1970, he has been a faculty member in Systems Control, with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. His research interests have included stochastic control and filtering, geometric multivariable control, and discrete-event systems.

He is the author of Linear Multivariable Control: A Geometric Approach (3rd edition) (New York: Springer-Verlag, 1985) and co-author (with C. Ma) of Nonblocking Supervisory Control of State Tree Structures (New York: Springer-Verlag, 2005), and co-author (with K. Cai) of Supervisor Localization (New York: Springer-Verlag, 2015).

Wonham is a Fellow of the Royal Society of Canada and a Foreign Member of the (U.S.) National Academy of Engineering. In 1987, he received the IEEE Control Systems Science and Engineering Award and in 1990 was Brouwer Medallist of the Netherlands Mathematical Society. In 1996 he was appointed University Professor in the University of Toronto, and in 2000 University Professor Emeritus.