Characterizations and Effective Computation of Supremal Relatively Observable Sublanguages*

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

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. The computational effectiveness of the operator is demonstrated on a case study.

Keywords

Supervisory control, partial-observation, relative observability, regular language, Nerode equivalence relation, support relation, discrete-event systems, automata

I. Introduction

In [3] 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

---

*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. 7399.

[1]K. Cai is with Urban Research Plaza, Osaka City University, Japan (kai.cai@eng.osaka-cu.ac.jp)

[2]R. Zhang is with School of Automation, Northwestern Polytechnical University, China (ryzhang@nwpu.edu.cn)

[3]W.M. Wonham is with the Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto, Canada (wonham@control.utoronto.ca).

to which the standard observability conditions [8] are tested. Relative observability is proved to be stronger than observability [5], [8], weaker than normality [5], [8], 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.

In this paper and its conference precursor [2], we present a new characterization of relative observability. The original definition of relative observability in [3] 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 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* ([9, Section 2.8]) based on Nerode equivalence relations [7]. 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 [10] 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 [3]. 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. On the other hand, both the language and automaton-based algorithms have (at least) exponential complexity in the worst case, which is the unfortunate nature of supervisor synthesis under partial observation. Our previous experience with the automaton-based algorithm in [3] 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 have implemented this operator and tested its effectiveness on a case study.

The rest of the paper is organized as follows. In Section II 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 III 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 IV we combine relative observability and controllability, and propose an operator that effectively computes the supremal relatively observable and controllable sublanguage. Section V presents illustrative examples, and finally in Section VI we state conclusions.

This paper extends its conference precursor [2] in the following respects. (1) In the main result of Section III, Theorem 1, 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 Subsection III-C. (3) Relative observability is combined with controllability in Section IV, and a new operator is presented that effectively computes the supremal relatively observable and controllable sublanguage. (4) A case study is given in Subsection V-B to demonstrate the effectiveness of the newly proposed computation schemes.

II. Characterizations of Relative Observability and its Supremal Element

In this section, the concept of relative observability proposed in [3] is first reviewed. Then we present a new characterization of relative observability, together with a fixpoint characterization of the supremal relatively observable sublanguage.

A. 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 [5], [8]), 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 [3] that in general, relative observability is stronger than observability, weaker than normality, and closed under arbitrary set unions. 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.

B. 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 [1], [4].

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 in 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 $D(\overline{K}) = \emptyset$. 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*

$$\text{(i}'\text{)}\ D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$$

$$\text{(ii}'\text{)}\ [K] \cap \big(\overline{C} \cap M\big) = K.$$

Note that condition (i′) is in a form similar to controllability of K [10] (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 normality of K with respect to $\overline{C} \cap M$.

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

1. (i′) ⇒ (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

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

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

and $\sigma \in \Sigma$. Then

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

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 $s \in K$. $\square$

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

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

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

C. Fixpoint Characterization of $\sup \mathcal{O}(C)$

For a string $s \in \Sigma^*$, write $\bar{s}$ for $\overline{\{s\}}$, the set of prefixes of s. Given a language $K \subseteq \Sigma^*$, let

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

Lemma 1. *$F(K)$ is closed, i.e. $\overline{F(K)} = F(K)$. Moreover, if $K \in \mathcal{O}(C)$, then $F(K) = \overline{K}$.*

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, suppose that $K \in \mathcal{O}(C)$; by (7) we have $D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$. Let $s \in \overline{K}$; it will be shown that $D(\bar{s}) \cap \overline{M} \subseteq \overline{K}$. Taking an arbitrary string $t \in D(\bar{s}) \cap \overline{M}$, we derive

$$\begin{aligned}
&t \in \bigcup \big\{[\bar{s} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma\big\} \cap \overline{M} \\
\Rightarrow &t \in \bigcup \big\{[\overline{K} \cap \overline{C}.\sigma] \cap \overline{C}.\sigma \mid \sigma \in \Sigma\big\} \cap \overline{M} \\
\Rightarrow &t \in \overline{K}.
\end{aligned}$$

This shows that $s \in F(K)$ by (8), and hence $\overline{K} \subseteq F(K)$. The other direction $F(K) \supseteq \overline{K}$ is automatic. $\square$

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

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

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

Proof. Since $\sup \mathcal{O}(C) \in \mathcal{O}(C)$, we have

$$\begin{aligned}
\Omega(\sup \mathcal{O}(C)) &= \sup \mathcal{N}\big(\sup \mathcal{O}(C) \cap F(\sup \mathcal{O}(C)), \overline{C} \cap M\big) \\
&= \sup \mathcal{N}(\sup \mathcal{O}(C) \cap \overline{\sup \mathcal{O}(C)}, \overline{C} \cap M) \\
&= \sup \mathcal{N}(\sup \mathcal{O}(C), \overline{C} \cap M) \\
&= \sup \mathcal{O}(C).
\end{aligned}$$

Next let K be such that $K = \Omega(K)$. To show that $K \subseteq \sup \mathcal{O}(C)$, it suffices to show that $K \in \mathcal{O}(C)$. From

$$K = \Omega(K) := \sup \mathcal{N}\big(K \cap F(K),\ \overline{C} \cap M\big)$$

we have $K \subseteq K \cap F(K)$. But $K \cap F(K) \subseteq K$. Hence, in fact, $K = K \cap F(K)$. This implies that $K = \sup \mathcal{N}\big(K,\ \overline{C} \cap M\big)$; namely K is normal with respect to $\overline{C} \cap M$.

On the other hand, by $K = K \cap F(K) \subseteq F(K)$, we have $\overline{K} \subseteq \overline{F(K)} = F(K)$. But $F(K) \subseteq \overline{K}$ by definition; therefore $\overline{K} = F(K)$. 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

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

Then by (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 $D(\overline{K}) \cap \overline{M} \subseteq \overline{K}$. This completes the proof of $K \in \mathcal{O}(C)$. $\square$

In view of Proposition 2, 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 3. *If K_∞ in (11) is reached in a finite number of steps, then*

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

Proof. Suppose that the limit K_∞ is reached in a finite number of steps. Then $K_\infty = \Omega(K_\infty)$. As in the proof of Proposition 2, we derive that $K_\infty \in \mathcal{O}(C)$.

It remains to show that K_∞ is the supremal element of $\mathcal{O}(C)$. Let $K' \in \mathcal{O}(C)$; it will be shown that $K' \subseteq K_\infty$ by induction. The base case $K' \subseteq K_0$ holds because $K' \subseteq C$ and $K_0 = C$. Suppose that

$K' \subseteq K_{j-1}$. Let $s \in \overline{K'}$. Then $s \in \overline{K_{j-1}}$ and

$$\begin{aligned} D(\overline{s}) \cap \overline{M} &\subseteq D(\overline{K'}) \cap \overline{M} \\ &\subseteq \overline{K'} \quad \text{(by } K' \in \mathcal{O}(C)\text{)} \\ &\subseteq \overline{K_{j-1}}. \end{aligned}$$

Hence $s \in F(K_{j-1})$. This shows that

$$\begin{aligned} &\overline{K'} \subseteq F(K_{j-1}) \\ &\Rightarrow K' \subseteq F(K_{j-1}) \\ &\Rightarrow K' \subseteq K_{j-1} \cap F(K_{j-1}). \end{aligned}$$

Moreover, since $K' \in \mathcal{O}(C)$, K' is normal with respect to $\overline{C} \cap M$. Thus $K' \subseteq \sup \mathcal{N}\big(K_{j-1} \cap F(K_{j-1}),\ \overline{C} \cap M\big) = K_j$. This completes the proof of the induction step, and therefore confirms that $K' \subseteq K_\infty$. □

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

III. 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 then prove that the sequence generated by (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.

A. 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

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

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

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

For a language $L \subseteq \Sigma^*$, write $\text{Ner}(L)$ for the *Nerode equivalence relation* [7] on Σ^* with respect to L; namely for all $s_1, s_2 \in \Sigma^*$, $s_1 \equiv s_2(\text{mod Ner}(L))$ 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 $\text{Ner}(L)$, i.e. $||L|| := |\text{Ner}(L)|$. The language L is said to be *regular* [7] 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

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

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 [9, Section 2.8] if $\overline{L_1} \subseteq \overline{L_2}$ and

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

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 2. *[9, 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 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 (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 $\mu := \text{Ner}(M)$, $\eta := \text{Ner}(C)$ 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 $f_\pi : \Sigma^* \to \text{Pwr}(\Sigma^*/\pi)$ according to

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

where $[s] = P^{-1}P(\{s\})$. Write $\wp(\pi) := \ker f_\pi$. The size of $\wp(\pi)$ is $|\wp(\pi)| \leq 2^{|\pi|}$ [9, Ex. 1.4.21]. Another property of $\wp(\cdot)$ we shall use later is [9, 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^*.

B. Convergence Result

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

Proposition 4. *Consider the sequence $\{K_j\}$ generated by (10). For each $j \geq 1$, there holds that K_j is ρ-supported on K_{j-1}, where*

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

Let us postpone the proof of Proposition 4, and present immediately our main result.

Theorem 1. *Consider the sequence $\{K_j\}$ generated by (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 (14). Since μ and η are right congruences, so are $\mu \wedge \eta$ and $\wp(\mu \wedge \eta)$ ([9, Example 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 3 and 2 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.

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

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

In the sequel we prove Proposition 4, for which we need two lemmas.

Lemma 3. *For each $j \geq 1$, the Nerode equivalence relation on Σ^* with respect to $F(K_{j-1})$ satisfies*

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

Proof. 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 $s_1 \equiv s_2(\text{mod Ner}(F(K_{j-1})))$.

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

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

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 $s_1 \equiv s_2(\text{mod Ner}(K_{j-1}))$. Hence it is left to show that $D(\overline{s_2 w}) \cap \overline{M} \subseteq \overline{K_{j-1}}$, i.e.

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

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 of $s_2 w \in F(K_{j-1})$, as required. □

Lemma 4. *For K_j ($j \geq 1$) generated by (10), the following statements hold:*

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

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

Proof. By (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 Example 6.1.25 of [9]. $\square$

Now we are ready to prove Proposition 4.

Proof of Proposition 4. 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 3 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 4 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 4 we derive that for all $s_1' \in [s_1] \cap \big(\overline{C} \cap M\big)$ and $s_2' \in [s_2] \cap \big(\overline{C} \cap M\big)$, $s_1', s_2' \in \overline{K_1}$. Since $\overline{K_1} \subseteq F(K_0)$, $s_1', s_2' \in F(K_0)$ and hence

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

Namely $s_1 \equiv s_2(\text{mod } \wp(\varphi_1 \wedge \mu \wedge \eta))$. This implies that $s_1 \equiv s_2(\text{mod Ner}(K_1))$ by (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

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

Again by Lemma 3 we have

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

Then by Lemma 4,

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

We claim that

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

To show this, let $s_1, s_2 \in \Sigma^*$ and assume that $s_1 \equiv s_2(\text{mod } \kappa_j \wedge \mu \wedge \eta \wedge \wp(\mu \wedge \eta))$. 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 $s_1 \equiv s_2(\text{mod Ner}(K_j))$. Now let $s_1, s_2 \in \overline{K_j}$. By Lemma 4 we derive that for all $s_1' \in [s_1] \cap \big(\overline{C} \cap M\big)$ and $s_2' \in [s_2] \cap \big(\overline{C} \cap M\big)$, $s_1', s_2' \in \overline{K_j}$. Since $\overline{K_j} \subseteq F(K_{j-1}) \subseteq \overline{K_{j-1}}$,

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

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

$\square$

C. Effective Computability of Ω

We conclude this section by showing that the iteration scheme in (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 1, it suffices to prove that the operator Ω in (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,[1] union, and intersection all preserve regularity and are effectively computable [6]. Moreover, both the operator $\sup\mathcal{N} : Pwr(\Sigma^*) \to Pwr(\Sigma^*)$ given by

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

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 [4] and [10], respectively).

The main result of this subsection is the following theorem.

Theorem 2. *Suppose that M and C are regular. Then the operator Ω in (9) preserves regularity and is effectively computable.*

The following proposition is a key fact.

Proposition 5. *For each $K \subseteq \Sigma^*$,*

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

Proof. By (8) and (6),

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

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

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}
$$

□

We also need the following lemma.

Lemma 5. *Let $\sigma \in \Sigma$ be fixed. Then the operator $B_\sigma : Pwr(\Sigma^*) \to Pwr(\Sigma^*)$ given by*

$$B_\sigma(L) := \overline{L}.\sigma = \{s\sigma \mid s \in \overline{L}\}$$

preserves regularity and is effectively computable.

Proof. Let $\mathbf{G} = (Q, \Sigma, \delta, q_0, Q_m)$ be a finite-state automaton with $L_m(\mathbf{G}) = L$. We will construct a new finite-state automaton $\mathbf{H}$ such that $L_m(\mathbf{H}) = B_\sigma(L)$. The construction is in two steps. First, let q^* be a new state (i.e. $q^* \notin Q$), and define $\mathbf{G}' = (Q', \Sigma, \delta', q_0, Q'_m)$ where

$$Q' := Q \cup \{q^*\}, \quad \delta' := \delta \cup \{(q, \sigma, q^*) | q \in Q\}, \quad Q'_m := \{q^*\}.$$

Thus $\mathbf{G}'$ is a finite-state automaton with $L_m(\mathbf{G}') = B_\sigma(L)$. However, $\mathbf{G}'$ is *nondeterministic*, inasmuch as $\delta'(q, \sigma) = \{q', q^*\}$ whenever $\delta(q, \sigma)$ is defined and $\delta(q, \sigma) = q'$. The second step is hence to apply the standard subset construction to convert the nondeterministic $\mathbf{G}'$ to a *deterministic* finite-state automaton $\mathbf{H}$ with $L_m(\mathbf{H}) = L_m(\mathbf{G}') = B_\sigma(L)$. This completes the proof. □

Finally we present the proof of Theorem 2.

Proof of Theorem 2. By Proposition 5 and the definition of $\Omega : Pwr(\Sigma^*) \to Pwr(\Sigma^*)$ in (9), for each $K \subseteq \Sigma^*$ we derive

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

Since the language closure, complement, union, intersection, $\sup \mathcal{N}$, $\sup \mathcal{F}$ and $\overline{C}.\sigma$ (by Lemma 5) all preserve regularity and are effectively computable, the same conclusion for the operator Ω follows immediately. □

In the proof, we see that the operator Ω in (9) is decomposed into a sequence of standard or well-known language operations. This allows straightforward implementation of Ω using off-the-shelf algorithms.

IV. 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 [10].

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 closed-under-union property of both controllability and C-observability, the supremal controllable and C-observable sublanguage $\sup \mathcal{CO}(C)$ therefore 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

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

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

Proposition 6. *$\sup \mathcal{CO}(C) = \Gamma(\sup \mathcal{CO}(C))$, 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 $K \in \mathcal{CO}(C)$. $\square$

In view of Proposition 6, 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 III-A, we have the following key result.

Proposition 7. *Consider the sequence $\{K_j\}$ generated by (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 [10, 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 ($j = 1$),

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

Since $K_1 = \sup\mathcal{O}(H_1)$, we set up 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 4, it is readily shown that each T_i is ρ-supported on H_1; in particular,

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

This confirms the base case.

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

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

Thus

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

Again set up 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 4 that each T_i is ρ-supported on H_j; in particular,

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

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

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

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

The following theorem is the main result of this section, which follows directly from Proposition 7 and Lemma 2.

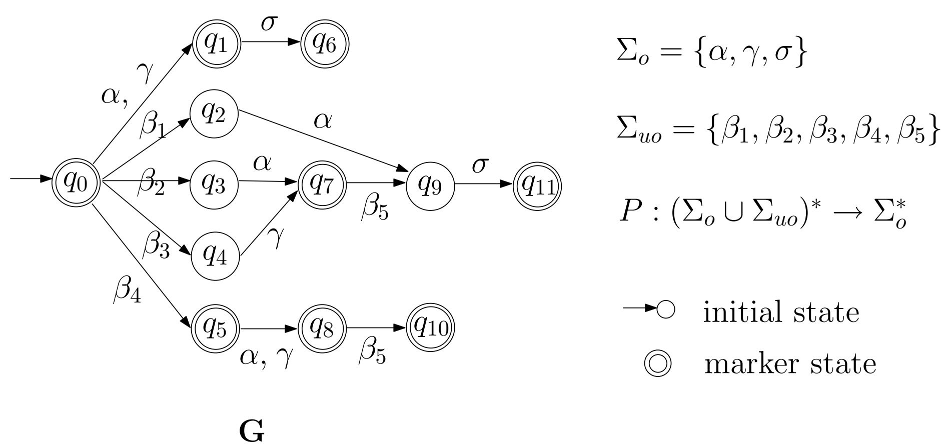

Fig. 1. Example: computation of the supremal C-observable sublanguage $\sup \mathcal{O}(C)$ by iteration of the operator Ω in (9)

Theorem 3. *Consider the sequence $\{K_j\}$ in (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*

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

Finally, $\sup \mathcal{CO}(C)$ is effectively computable, inasmuch as the operators $\sup \mathcal{C}(\cdot)$ and $\sup \mathcal{O}(\cdot)$ are (see [10] and Theorem 2, respectively). In particular, the operator Γ in (18) is effectively computable.

V. Examples

In this section, we first give an example to illustrate the computation of the supremal C-observable sublanguage $\sup \mathcal{O}(C)$ (by iteration of the operator Ω). Then we present an empirical study on the computation of the supremal controllable and C-observable sublanguage $\sup \mathcal{CO}(C)$ (by iteration of the operator Γ, which has been implemented by a computer program).

A. An Example of Computing $\sup \mathcal{O}(C)$

Consider the example displayed in Fig. 1. The observable event set is $\Sigma_o = \{\alpha, \gamma, \sigma\}$ and unobservable $\Sigma_{uo} = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5\}$; thus the natural projection is $P : (\Sigma_o \cup \Sigma_{uo})^* \to \Sigma_o^*$. Let

$$M := L_m(\mathbf{G}) = \{\epsilon, \alpha, \gamma, \alpha\sigma, \gamma\sigma, \beta_1\alpha\sigma, \beta_2\alpha, \beta_2\alpha\beta_5\sigma, \beta_3\gamma,$$
$$\beta_3\gamma\beta_5\sigma, \beta_4, \beta_4\alpha, \beta_4\gamma, \beta_4\alpha\beta_5, \beta_4\gamma\beta_5\}$$

and the specification language

$$C := M - \{\beta_4\alpha\beta_5, \beta_4\gamma\beta_5\}.$$

Both M and C are regular languages.

Now apply the operator Ω in (9). Initialize $K_0 = C$. The first iteration $j = 1$ starts with

$$\begin{aligned} F(K_0) &= \{s \in \overline{K_0} \mid D(\overline{s}) \cap \overline{M} \subseteq \overline{K_0}\} \\ &= \{\epsilon, \alpha, \gamma, \alpha\sigma, \gamma\sigma, \beta_1, \beta_1\alpha, \beta_1\alpha\sigma, \beta_2, \beta_2\alpha, \beta_3, \beta_3\gamma, \beta_4, \beta_4\alpha, \beta_4\gamma\} \\ &= \overline{K_0} - \{\beta_2\alpha\beta_5, \beta_2\alpha\beta_5\sigma, \beta_3\gamma\beta_5, \beta_3\gamma\beta_5\sigma\}. \end{aligned}$$

Note that since $\beta_2\alpha\beta_5\sigma \in K_0$, strings $\beta_2\alpha\beta_5, \beta_2\alpha\beta_5\sigma \in \overline{K_0}$. But $\beta_2\alpha\beta_5, \beta_2\alpha\beta_5\sigma \notin F(K_0)$; this is because the string $\beta_4\alpha\beta_5$ belongs to $D(\overline{\beta_2\alpha\beta_5}) \cap \overline{M}$ and $D(\overline{\beta_2\alpha\beta_5\sigma}) \cap \overline{M}$, but $\beta_4\alpha\beta_5$ does not belong to $\overline{K_0}$. For the same reason, $\beta_3\gamma\beta_5, \beta_3\gamma\beta_5\sigma \in \overline{K_0}$ but $\beta_3\gamma\beta_5, \beta_3\gamma\beta_5\sigma \notin F(K_0)$. Next calculate

$$\begin{aligned} F(K_0) \cap K_0 &= \{\epsilon, \alpha, \gamma, \alpha\sigma, \gamma\sigma, \beta_1\alpha\sigma, \beta_2\alpha, \beta_3\gamma, \beta_4, \beta_4\alpha, \beta_4\gamma\} \\ &= K_0 - \{\beta_2\alpha\beta_5\sigma, \beta_3\gamma\beta_5\sigma\}. \end{aligned}$$

Removing strings $\beta_2\alpha\beta_5\sigma, \beta_3\gamma\beta_5\sigma$ from K_0 makes $F(K_0) \cap K_0$ *not* normal with respect to $\overline{C} \cap M$. Indeed, $\alpha\sigma, \beta_1\alpha\sigma \in [\beta_2\alpha\beta_5\sigma] \cap \overline{C} \cap M$ and $\gamma\sigma \in [\beta_3\gamma\beta_5\sigma] \cap \overline{C} \cap M$ violate the normality condition and therefore must also be removed. Hence,

$$\begin{aligned} K_1 &= \sup \mathcal{N}(F(K_0) \cap K_0, \overline{C} \cap L_m(\mathbf{G})) \\ &= \{\epsilon, \alpha, \gamma, \beta_2\alpha, \beta_3\gamma, \beta_4, \beta_4\alpha, \beta_4\gamma\} \\ &= (F(K_0) \cap K_0) - \{\alpha\sigma, \beta_1\alpha\sigma, \gamma\sigma\}. \end{aligned}$$

This completes the first iteration $j = 1$.

Since $K_1 \subsetneqq K_0$, we proceed to $j = 2$,

$$\begin{aligned} F(K_1) &= \{s \in \overline{K_1} \mid D(\overline{s}) \cap \overline{M} \subseteq \overline{K_1}\} \\ &= \{\epsilon, \gamma, \beta_2, \beta_3, \beta_3\gamma, \beta_4, \beta_4\gamma\} \\ &= \overline{K_1} - \{\alpha, \beta_2\alpha, \beta_4\alpha\}. \end{aligned}$$

We see that $\alpha, \beta_2\alpha, \beta_4\alpha \in \overline{K_1}$ but $\alpha, \beta_2\alpha, \beta_4\alpha \notin F(K_1)$. This is because the string $\beta_1\alpha \in D(\overline{\alpha}) \cap \overline{M}$, $D(\overline{\beta_2\alpha}) \cap \overline{M}$, and $D(\overline{\beta_4\alpha}) \cap \overline{M}$, but $\beta_1\alpha \notin \overline{K_1}$. Note that $\beta_1\alpha$ was in $\overline{K_0}$ since $\beta_1\alpha\sigma \in K_0$, but $\beta_1\alpha\sigma$

was removed so as to ensure normality of K_1; this in turn removed $\beta_1\alpha$, which now causes removal of strings $\alpha, \beta_2\alpha, \beta_4\alpha$ altogether. Continuing,

$$
\begin{aligned}
F(K_1) \cap K_1 &= \{\epsilon, \gamma, \beta_3\gamma, \beta_4, \beta_4\gamma\} \\
&= K_1 - \{\alpha, \beta_2\alpha, \beta_4\alpha\}.
\end{aligned}
$$

Removing strings $\alpha, \beta_2\alpha, \beta_4\alpha$ does not destroy normality of K_1. Indeed $F(K_1) \cap K_1$ is normal with respect to $\overline{C} \cap M$ and we have

$$
\begin{aligned}
K_2 &= \sup \mathcal{N}(F(K_1) \cap K_1, \overline{C} \cap M) \\
&= \{\epsilon, \gamma, \beta_3\gamma, \beta_4, \beta_4\gamma\} \\
&= F(K_1) \cap K_1.
\end{aligned}
$$

This completes the second iteration $j = 2$.

Since $K_2 \subsetneqq K_1$, we proceed to $j = 3$ as follows:

$$
\begin{aligned}
F(K_2) &= \{s \in \overline{K_2} \mid D(\overline{s}) \cap \overline{M} \subseteq \overline{K_2}\} \\
&= \{\epsilon, \gamma, \beta_3, \beta_3\gamma, \beta_4, \beta_4\gamma\} = \overline{K_2}; \\
F(K_2) \cap K_2 &= \overline{K_2} \cap K_2 = K_2; \\
K_3 &= \sup \mathcal{N}(F(K_2) \cap K_2, \overline{C} \cap M) \\
&= \sup \mathcal{N}(K_2, \overline{C} \cap M) = K_2.
\end{aligned}
$$

Since $K_3 = K_2$, the limit of the sequence in (10) is reached. Therefore

$$
K_3 = \{\epsilon, \gamma, \beta_3\gamma, \beta_4, \beta_4\gamma\}
$$

is the supremal C-observable sublanguage of C.

B. A Case Study of Computing $\sup \mathcal{CO}(C)$

Consider the same case study as in [3, Section V-B], namely a manufacturing workcell served by five automated guided vehicles (AGV). Adopting the same settings, we apply the implemented Γ operator to compute the supremal relatively observable and controllable sublanguage $\sup \mathcal{CO}(C)$, as represented by a finite-state automaton, say **SUPO**. That is,

$L_m(\mathbf{SUPO}) = \sup \mathcal{CO}(C)$.

For this case study, the full-observation supervisor (representing the supremal controllable sublanguage) has 4406 states and 11338 transitions. Selecting different subsets of unobservable events, the

TABLE I. **SUPO** COMPUTED FOR DIFFERENT SUBSETS OF UNOBSERVABLE EVENTS IN THE AGV CASE STUDY

$\Sigma_{uo} = \Sigma - \Sigma_o$	State #, transition # of **SUPO**
$\{$13$\}$	(4406,11338)
$\{$21$\}$	(4348,10810)
$\{$31$\}$	(4302,11040)
$\{$43$\}$	(4319,10923)
$\{$51$\}$	(4400,11296)
$\{$12,31$\}$	(1736,4440)
$\{$24,41$\}$	(4122,10311)
$\{$31,43$\}$	(4215,10639)
$\{$32,51$\}$	(2692,6596)
$\{$41,51$\}$	(3795,9355)
$\{$11,31,41$\}$	(163,314)
$\{$12,33,51$\}$	(94,140)
$\{$12,24,33,44,53$\}$	(72,112)
$\{$12,21,32,43,51$\}$	(166,314)
$\{$13,23,31,33, 41,43,51,53$\}$	(563,1244)

computational results for the supremal relatively observable and controllable sublanguages, or **SUPO**, are listed in Table I. We see in all cases but the first ($\Sigma_{uo} = \{13\}$) that the state and transition numbers of **SUPO** are fewer than those of the full-observation supervisor. When $\Sigma_{uo} = \{13\}$, in fact, the supremal controllable sublanguage is already observable, and is therefore itself the supremal relatively observable and controllable sublanguage.

Moreover, we have confirmed that the computation results agree with those by the algorithm in [3]. Thus the new computation scheme provides a useful alternative to ensure presumed correctness based on consistency.

VI. 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 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. Finally we have presented an example and a case study to illustrate the effectiveness of the proposed computation schemes.

REFERENCES

[1] 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 & Control Letters*, 15(2):111–117, 1990.

[2] K. Cai and W. M. Wonham. A new algorithm for computing the supremal relatively observable sublanguage. In *Proc. Workshop on Discrete-Event Systems*, pages 8–13, Xi'an, China, 2016.

[3] K. Cai, R. Zhang, and W. M. Wonham. Relative observability of discrete-event systems and its supremal sublanguages. *IEEE Trans. Autom. Control*, 60(3):659–670, 2015.

[4] H. Cho and S. I. Marcus. On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation. *Math. of Control, Signals, and Systems*, 2(1):47–69, 1989.

[5] R. Cieslak, C. Desclaux, A. S. Fawaz, and P. Varaiya. Supervisory control of discrete-event processes with partial observations. *IEEE Trans. Autom. Control*, 33(3):249–260, 1988.

[6] S. Eilenberg. *Automata, Languages and Machines*. Volume A, Academic Press, 1974.

[7] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[8] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Inform. Sci.*, 44(3):173–198, 1988.

[9] W. M. Wonham. 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, 2016.

[10] W. M. Wonham and P. J. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Control and Optimization*, 25(3):637–659, 1987.