

Complexity of Local Observation Consistency in Discrete-Event Systems

Tomáš Masopust

Faculty of Science, Palacký University Olomouc, Czechia

Abstract

Hierarchical and multi-agent supervisory control under partial observation relies on *local* consistency conditions between a plant and its abstraction: local observation consistency (LOC) for the projection abstractions of hierarchical control, and local relabeling observation consistency (LROC) for the relabeling abstractions of multi-agent systems. Both are companions to a global condition and both have been used without their verification complexity being settled. We show that both are PSPACE-complete for nondeterministic plants and decidable in polynomial time for deterministic plants.

Key words: Discrete-event systems, Supervisory control, Partial observation, Hierarchical control, Multi-agent systems

1 Introduction

Hierarchical supervisory control of discrete-event systems (DES) controls a plant through an abstraction of it, and the central question is which conditions on the abstraction make synthesis on the abstract level correct and maximally permissive. Under complete observation, the answer is well known and easy to verify [1, 2, 3]. Under partial observation, it is not. Boutin *et al.* [4] introduced two conditions for the projection abstraction: *observation consistency* (OC), a global condition relating the observations of the two levels, and *local observation consistency* (LOC), a local companion that plays for the abstraction the role the observer property plays under complete observation.

The same pattern appears in multi-agent DES [5, 6], where groups of isomorphic agents are collapsed onto a common template by a *relabeling*. There the global condition is relabeling observation consistency (ROC) and the local companion is *local relabeling observation consistency* (LROC) [6].

Neither condition has its verification complexity settled. For LOC, we announced PSPACE-completeness [7]; however, the claim was given without proof. The proof here is significantly improved and simplified compared with the one presented in the arxiv version of [7]. For LROC, Liu *et al.* [6] observe that it is “similar to a variant of

observability and thus can be checked in a similar way”, which is correct for deterministic plants but leaves the case of nondeterministic plants open.

We show that both LOC verification and LROC verification are PSPACE-complete for nondeterministic plants and decidable in polynomial time for deterministic plants. In particular, for deterministic plants, LOC is verifiable in polynomial time, which corrects an unproved claim in [7].

2 Preliminaries

We assume that the reader is familiar with the basics of supervisory control [8] and complexity theory [9]; P and PSPACE denote the classes of problems decidable in polynomial time and in polynomial space.

An *alphabet* is a finite nonempty set of events. For an alphabet Σ , the set of finite strings over Σ is denoted by Σ^* , and the empty string is denoted by ε . We write $A \dot{\cup} B$ for the union of *disjoint* sets A and B , and we identify a singleton set $\{q\}$ with the element q . A language L over Σ is a subset of Σ^* . The *prefix closure* of L is the set $\bar{L} = \{u \in \Sigma^* : uv \in L \text{ for some } v \in \Sigma^*\}$, and L is *prefix-closed* if $L = \bar{L}$.

A *nondeterministic finite automaton* (NFA) is a tuple $G = (Q_G, \Sigma, \delta, I, F)$, where Q_G is a finite set of states, $I \subseteq Q_G$ is a set of initial states, $F \subseteq Q_G$ is the set of final states, and $\delta: Q_G \times \Sigma \rightarrow 2^{Q_G}$ is the transition

Email address: tomas.masopust@upol.cz (Tomáš Masopust).

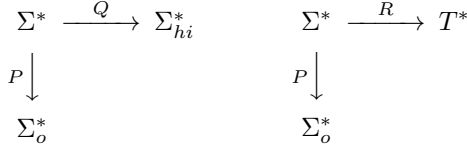


Fig. 1. The abstractions used in the paper.

function that can be extended to $2^{Q_G} \times \Sigma^*$ in the usual way. Since every function is a relation, we also write $\delta \subseteq Q_G \times \Sigma \times Q_G$. The NFA is *deterministic* (DFA) if $|\delta(q, a)| \leq 1$ for every $q \in Q_G$ and $a \in \Sigma$, and $I = \{q_0\}$ for a single *initial state* q_0 . The language of G is the set $L(G) = \{w \in \Sigma^* : \delta(I, w) \cap F \neq \emptyset\}$. Every plant here is prefix-closed, and therefore we take $F = Q_G$, writing *NFA with all states final*; in this case, we omit F from the notation. The languages of such automata are the prefix-closed regular languages [10]. Automata with $F \neq Q_G$ arise only inside proofs.

For alphabets $\Sigma_1 \subseteq \Sigma$, the (*natural*) *projection* $p: \Sigma^* \rightarrow \Sigma_1^*$ is the morphism with $p(a) = a$ for $a \in \Sigma_1$ and $p(a) = \varepsilon$ otherwise; we write $s|_{\Sigma_1}$ for $p(s)$. For a morphism f and a language K , the *inverse image* of f is $f^{-1}(K) = \{s : f(s) \in K\}$. Under partial observation, the alphabet is partitioned as $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$ of observable and unobservable events, respectively, and $P: \Sigma^* \rightarrow \Sigma_o^*$ is the *observation*.

Two abstractions are considered. In the hierarchical setting, a *high-level alphabet* $\Sigma_{hi} \subseteq \Sigma$ is fixed and the abstraction is the projection $Q: \Sigma^* \rightarrow \Sigma_{hi}^*$. In the multi-agent setting, a *relabeling* $R: \Sigma^* \rightarrow T^*$ onto a template alphabet T disjoint from Σ is fixed; it is a letter-to-letter morphism, that is, $R(s_1 s_2) = R(s_1) R(s_2)$, $|R(s)| = |s|$, and it is in general not injective. In the sequel, the letters Q and R are reserved for these two abstractions, see Fig. 1. For both, we assume that they *preserve the observability status of events*; in particular, for $T = T_o \dot{\cup} T_{uo}$, we have $R(\Sigma_o) \subseteq T_o$ and $R(\Sigma_{uo}) \subseteq T_{uo}$.

Definition 1 (LOC [4]). A prefix-closed language $L \subseteq \Sigma^*$ is *locally observation consistent* (LOC) with respect to Q , P , and a set $\Sigma_c \subseteq \Sigma$ if for all $s, s' \in L$ with $P(s) = P(s')$ and every $e \in \Sigma_c \cap \Sigma_{hi}$ such that $Q(s)e \in Q(L)$ and $Q(s')e \in Q(L)$, there exist fillers $y, y' \in (\Sigma \setminus \Sigma_{hi})^*$ such that $P(y) = P(y')$, $sy e \in L$, and $s'y'e \in L$.

Definition 2 (LROC [6]). A relabeling R is *locally relabeling observation consistent* (LROC) with respect to a prefix-closed language $L \subseteq \Sigma^*$ and the observation P if for all $s, s' \in L$ with $P(s) = P(s')$ and all unobservable events $b, b' \in \Sigma_{uo}$ with $R(b) = R(b')$,

$$sb \in L \wedge s'b' \in L \implies s'b \in L. \quad (1)$$

These conditions play the same role in their respective frameworks, and both are companions to a global condition that guarantees maximal permissiveness. However, their premises are of different kinds—LOC asks that the event be enabled in the *abstraction*, LROC that it be enabled in the *plant* after one of the two strings—and we treat them separately.

3 The Local Condition of Hierarchical Control

Let $G = (Q_G, \Sigma, \delta, I)$ be an NFA with all states final, $\Sigma_c \subseteq \Sigma$, and $e \in \Sigma_c \cap \Sigma_{hi}$. For a set $X \subseteq Q_G$, write

$$\text{Fill}(X, e) = \{P(y) : y \in (\Sigma \setminus \Sigma_{hi})^*, \delta(X, ye) \neq \emptyset\}$$

for the set of observations of the fillers that lead from X to an occurrence of e . It is the image under P of a language recognized by G restricted to $(\Sigma \setminus \Sigma_{hi})$ -transitions, started in X , with the states carrying an outgoing e -transition made final. Thus, $\text{Fill}(X, e)$ is regular and an NFA for it of size polynomial in the size of G is immediate from G . Write $L = L(G)$. Since $sy e \in L$ if and only if $\delta(\delta(I, s), ye) \neq \emptyset$, the witnesses that LOC asks for are available if two such sets meet:

$$\begin{aligned}
&\exists y, y': P(y) = P(y') \wedge sy e \in L \wedge s'y'e \in L \\
&\iff \text{Fill}(\delta(I, s), e) \cap \text{Fill}(\delta(I, s'), e) \neq \emptyset.
\end{aligned} \quad (2)$$

We call a triple (s, s', e) with $s, s' \in L$ and $e \in \Sigma_c \cap \Sigma_{hi}$ *admissible* if $P(s) = P(s')$ and $Q(s)e, Q(s')e \in Q(L)$, that is, if it satisfies the premise of LOC. Simplifying the notation by writing $\text{Fill}(s, e) = \text{Fill}(\delta(I, s), e)$, (2) states that the language L is LOC if and only if $\text{Fill}(s, e) \cap \text{Fill}(s', e) \neq \emptyset$ for every admissible triple (s, s', e) .

Lemma 3. A prefix-closed language $L = L(G)$ is LOC with respect to Q , P , and Σ_c if and only if both of the following conditions hold for every $e \in \Sigma_c \cap \Sigma_{hi}$:

- (C1) for all $s, z \in L$ with $Q(s) = Q(z)$, if $\text{Fill}(z, e)$ is nonempty, then so is $\text{Fill}(s, e)$;
- (C2) for all $s, s' \in L$ with $P(s) = P(s')$, if $\text{Fill}(s, e)$ and $\text{Fill}(s', e)$ are both nonempty, then they intersect.

Proof. The proof is based on the following reformulation of the premise of LOC: for $s \in L$ and $e \in \Sigma_{hi}$,

$$\begin{aligned}
Q(s)e \in Q(L) &\iff \text{Fill}(z, e) \neq \emptyset \text{ for some } z \in L \\
&\text{with } Q(z) = Q(s). \quad (3)
\end{aligned}$$

To see it, take $u \in L$ with $Q(u) = Q(s)e$. As Q erases letters outside Σ_{hi} and e is the last letter of $Q(u)$, we can write $u = zye y'$ with $y, y' \in (\Sigma \setminus \Sigma_{hi})^*$ and $Q(z) = Q(s)$. Since L is prefix-closed, $zye \in L$, and hence $P(y) \in \text{Fill}(z, e)$. For the converse, $P(y) \in \text{Fill}(z, e)$ means that $zye \in L$ and $Q(s)e = Q(z)e \in Q(L)$.

We now prove the lemma. To this end, assume that L is LOC, that is, $\text{Fill}(s, e) \cap \text{Fill}(s', e) \neq \emptyset$ for every admissible triple (s, s', e) . To show (C1), let $Q(s) = Q(z)$ with $\text{Fill}(z, e) \neq \emptyset$. By (3), $Q(s)e \in Q(L)$. Since the triple (s, s, e) is admissible, we have $\text{Fill}(s, e) \cap \text{Fill}(s, e) \neq \emptyset$, and hence $\text{Fill}(s, e) \neq \emptyset$ as required. To show (C2), let $P(s) = P(s')$ with both $\text{Fill}(s, e)$ and $\text{Fill}(s', e)$ nonempty. Taking $z = s$ in (3) gives $Q(s)e \in Q(L)$, and taking $z = s'$ gives $Q(s')e \in Q(L)$, that is, (s, s', e) is admissible and LOC yields the intersection.

Conversely, assume (C1) and (C2), and let $s, s' \in L$ with $P(s) = P(s')$ and $Q(s)e, Q(s')e \in Q(L)$ be given. By (3) applied to s , there is $z \in L$ with $Q(z) = Q(s)$ and $\text{Fill}(z, e) \neq \emptyset$; thus, by (C1), $\text{Fill}(s, e) \neq \emptyset$. The same argument applied to s' gives $\text{Fill}(s', e) \neq \emptyset$. Then, (C2) yields $\text{Fill}(s, e) \cap \text{Fill}(s', e) \neq \emptyset$, which by (2) is the conclusion required by LOC. Hence L is LOC. $\square$

Both conditions quantify over pairs of strings that agree on a sub-alphabet, and such pairs are tracked by the following product of G with itself. For $\Xi \subseteq \Sigma$, let $G \otimes_{\Xi} G$ be the directed graph whose vertices are the pairs (X, Y) of subsets of Q_G and whose edges are

$$\begin{aligned} (X, Y) &\xrightarrow{a} (\delta(X, a), \delta(Y, a)) && \text{for } a \in \Xi, \\ (X, Y) &\xrightarrow{a} (\delta(X, a), Y) && \text{for } a \notin \Xi, \\ (X, Y) &\xrightarrow{a} (X, \delta(Y, a)) && \text{for } a \notin \Xi. \end{aligned}$$

A letter of Ξ advances both components, and a letter outside Ξ advances one of them. Let Π_{Ξ} denote the set of vertices reachable from (I, I) of sets of initial states whose two components are both nonempty.

Lemma 4. *For every $\Xi \subseteq \Sigma$ and all $X, Y \subseteq Q_G$, the pair (X, Y) belongs to Π_{Ξ} if and only if there are $s, s' \in L$ with $s|_{\Xi} = s'|_{\Xi}$, $X = \delta(I, s)$, and $Y = \delta(I, s')$.*

Proof. Let a path from (I, I) to (X, Y) be given. Each of its edges is labelled by a letter and advances the first component, the second, or both. Reading the letters of the edges that advance the first component gives a string s , and reading those that advance the second gives a string s' ; that is, $(X, Y) = (\delta(I, s), \delta(I, s'))$. The edges labelled by Ξ advance both components and contribute that letter to s and s' at the same position; therefore, $s|_{\Xi} = s'|_{\Xi}$. Finally, all states of G are final, and hence $X \neq \emptyset$ is equivalent to $s \in L$ and $Y \neq \emptyset$ to $s' \in L$.

Conversely, let $s, s' \in L$ satisfy $s|_{\Xi} = s'|_{\Xi} = a_1 \cdots a_k$, and write $s = y_0 a_1 y_1 \cdots a_k y_k$ and $s' = y'_0 a_1 y'_1 \cdots a_k y'_k$ with $y_j, y'_j \in (\Sigma \setminus \Xi)^*$. Starting at (I, I) , take, for $j = 0, \dots, k$, first the edges that read y_j in the first component only, then those that read y'_j in the second component only, and then, for $j < k$, the edge labelled a_{j+1} that advances both. The vertex reached is $(\delta(I, s), \delta(I, s'))$, and its components are nonempty because $s, s' \in L$. $\square$

For $\Xi = \Sigma_{hi}$ and $\Xi = \Sigma_o$, Lemma 3 reads as follows:

$$\begin{aligned} \forall (X, Y) \in \Pi_{\Sigma_{hi}} : \text{Fill}(Y, e) \neq \emptyset &\Rightarrow \text{Fill}(X, e) \neq \emptyset, && (4) \\ \forall (X, Y) \in \Pi_{\Sigma_o} : \text{Fill}(X, e) \neq \emptyset, \text{Fill}(Y, e) \neq \emptyset &\Rightarrow \text{Fill}(X, e) \cap \text{Fill}(Y, e) \neq \emptyset. && (5) \end{aligned}$$

The LOC verification problem asks, given an NFA G over Σ and alphabets $\Sigma_o, \Sigma_{hi}, \Sigma_c \subseteq \Sigma$, whether $L(G)$ is LOC with respect to Q, P , and Σ_c .

Proposition 5. *LOC verification is in PSPACE.*

Proof. By Lemmata 3 and 4, $L(G)$ is not LOC if and only if there are $e \in \Sigma_c \cap \Sigma_{hi}$ and a pair (X, Y) that violates (4) or (5). We describe a nondeterministic procedure that searches for such a witness in polynomial space. The procedure first guesses e and which of the conditions is violated. It then guesses a path of $G \otimes_{\Xi} G$, with $\Xi = \Sigma_{hi}$ or $\Xi = \Sigma_o$ accordingly, one step at a time, storing only the current pair (X, Y) and a step counter; it stops at a nondeterministically chosen moment and verifies the violation at the pair reached. A pair of subsets of Q_G requires $2|Q_G|$ space and the counter, bounded by the number $2^{2|Q_G|}$ of pairs, occupies $2|Q_G|$ bits; hence the space is linear in the number of states of G . A path longer than the bound goes through a cycle and may be truncated, that is, the counter terminates the search without affecting the set of reachable pairs.

It remains to verify a violation of the reached pair (X, Y) . For a set $X \subseteq Q_G$, an NFA for $\text{Fill}(X, e)$ is obtained from G by keeping only the transitions under $\Sigma \setminus \Sigma_{hi}$, taking X as the set of initial states, declaring final those states that have an outgoing e -transition, and relabeling each transition under c by c if $c \in \Sigma_o$ and by ε otherwise; it has $|Q_G|$ states and is constructible from X on the fly. Analogously for Y . Emptiness of the language of an NFA and emptiness of the intersection of the languages of two NFAs are decidable in nondeterministic logarithmic space in their sizes, by guessing an accepting path, respectively a pair of accepting paths of equal label. Thus, given (X, Y) , both tests run in space logarithmic in the size of G .

The procedure runs in nondeterministic space $O(|Q_G|)$. The result therefore follows from Savitch's theorem, according to which nondeterministic polynomial space is equal to deterministic polynomial space; symbolically, $\text{NPSpace} = \text{PSPACE}$. $\square$

The following lower bound is stated in [7, Thm. 6] without proof. We provide a proof (significantly simplified compared with the proof in the appendix of the arxiv version of [7]) and further improve the result.

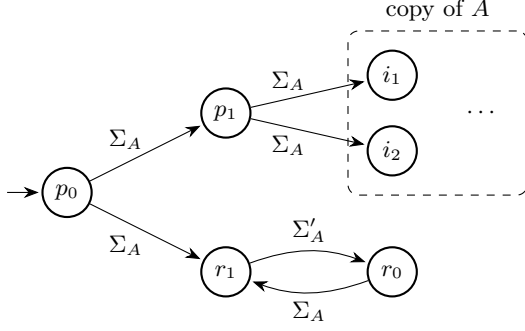


Fig. 2. The automaton B of Proposition 6. The upper part is the A -branch, the lower part is the alternating branch.

Proposition 6. *LOC verification is PSPACE-hard for NFAs. It is PSPACE-hard even if $\Sigma_c = \Sigma_o = \Sigma_{hi}$.*

Proof. We reduce from the universality of NFAs with all states final, which is PSPACE-complete [10]. Let A be an NFA over Σ_A with all states final. Then $L(A)$ is prefix-closed, and we may assume that $\varepsilon \in L(A)$; otherwise, $L(A) = \emptyset$ and the instance is trivial. Let $\Sigma'_A = \{a' : a \in \Sigma_A\}$ be a disjoint copy of Σ_A , and set

$$\Sigma = \Sigma_A \cup \Sigma'_A \quad \text{and} \quad \Sigma_c = \Sigma_o = \Sigma_{hi} = \Sigma_A,$$

that is, $Q = P$ erases the primed letters and the alphabet $\Sigma \setminus \Sigma_{hi} = \Sigma'_A$ is entirely unobservable. From A , we construct an NFA B over Σ , with all states final, as follows: B has a fresh initial state p_0 , a fresh state p_1 , a copy of A , and two fresh states r_0, r_1 ; it has the transitions (p_0, a, p_1) and (p_1, a, i) for every $a \in \Sigma_A$ and every initial state i of A , together with all transitions of the copy of A , and the transitions (p_0, a, r_1) , (r_1, a', r_0) , and (r_0, a, r_1) for every $a \in \Sigma_A$. Thus

$$L = L(B) = \overline{\Sigma_A \Sigma_A L(A)} \cup \overline{(\Sigma_A \Sigma'_A)^*},$$

the first branch goes through states p_0, p_1 and the copy of A , called the *A-branch*, and the second goes through states r_0, r_1 , called the *alternating branch*; see Fig. 2. The automaton B has $|A| + 4$ states and is constructed from A in time polynomial in the size of A .

Since the alphabet Σ'_A is unobservable, $P(u) = \varepsilon$ for every $u \in \Sigma'^*_A$, and LOC is equivalent to the condition

$$\forall s \in L \forall e \in \Sigma_c \cap \Sigma_{hi}: \\ Q(s)e \in Q(L) \implies \exists u \in (\Sigma \setminus \Sigma_{hi})^*: sue \in L. \quad (6)$$

Indeed, LOC instantiated at the triple (s, s, e) gives (6). Conversely, suppose (6) and let $s, s' \in L$ and $e \in \Sigma_c \cap \Sigma_{hi}$ be such that $P(s) = P(s')$ and $Q(s)e, Q(s')e \in Q(L)$. Applying (6) to s and s' separately yields fillers u and u' with $sue \in L$ and $s'u'e \in L$. Since the filler alphabet $\Sigma \setminus \Sigma_{hi}$ is unobservable, the two fillers meet the requirement $P(u) = \varepsilon = P(u')$ of LOC.

It remains to show that (6) holds iff $L(A) = \Sigma_A^*$.

Assume that $L(A) = \Sigma_A^*$, and let $s \in L$ and $e \in \Sigma_A$. If s is generated along the alternating branch, then B is in state p_0, r_0 , or r_1 after s . In the first two cases, $se \in L$, and $u = \varepsilon$ is the required filler; in the third case, $sa'e \in L$ for every $a' \in \Sigma'_A$, and hence $u = a'$ is the required filler. If s is generated along the A -branch, then $s = abt$ with $a, b \in \Sigma_A$ and $t \in L(A)$. Since $L(A) = \Sigma_A^*$, we have $te \in L(A)$, and hence $se = abte \in L$ for $u = \varepsilon$. In all cases (6) holds.

Conversely, assume that $L(A) \neq \Sigma_A^*$. Since $\varepsilon \in L(A)$, there are $t \in L(A)$ and $e \in \Sigma_A$ with $te \notin L(A)$. For $a, b \in \Sigma_A$, the string $s = abt \in L$ is generated along the A -branch. However, no filler u satisfies $sue \in L$. Indeed, since s contains two consecutive unprimed letters, every string of L extending s is generated along the A -branch. After s , the automaton B is in a state of the copy of A , from which no primed letter is available, and therefore $u \in \{\varepsilon\}$. However, $se = abte \in L$ would imply $te \in L(A)$, which is a contradiction. Since $Q(L) = \Sigma_A^*$, we have $Q(s)e \in Q(L)$ and (6) does not hold. $\square$

Combining Propositions 5 and 6, we have the following.

Theorem 7. *LOC verification for NFAs is PSPACE-complete.* $\square$

For DFAs, the situation differs.

Theorem 8. *LOC verification for DFAs is decidable in polynomial time.*

Proof. Let $G = (Q_G, \Sigma, \delta, q_0)$ be a DFA with n states. Since G is deterministic, $\delta(q_0, s)$ is a single state for every $s \in L = L(G)$, and every vertex of $G \otimes_{\Xi} G$ reachable from (q_0, q_0) is a pair of states. Thus, $\Pi_{\Sigma_{hi}}$ and Π_{Σ_o} have at most n^2 elements each; both are computed by reachability in a graph with n^2 vertices. By Lemma 4, $(q, q'') \in \Pi_{\Sigma_{hi}}$ if and only if $q = \delta(q_0, s)$ and $q'' = \delta(q_0, z)$ for some $s, z \in L$ with $Q(s) = Q(z)$, and $(q, q') \in \Pi_{\Sigma_o}$ if and only if $q = \delta(q_0, s)$ and $q' = \delta(q_0, s')$ for some $s, s' \in L$ with $P(s) = P(s')$.

As in the proof of Proposition 5, an NFA for $\text{Fill}(q, e)$ is immediate from G , and emptiness of $\text{Fill}(q, e)$ and of $\text{Fill}(q, e) \cap \text{Fill}(q', e)$ are decidable in nondeterministic logarithmic space (polynomial time); we precompute both for all $O(n^2)$ pairs of states and all $e \in \Sigma_c \cap \Sigma_{hi}$. Conditions (C1) and (C2) of Lemma 3 then read

$$\begin{aligned} \forall (q, q'') \in \Pi_{\Sigma_{hi}}: \text{Fill}(q'', e) \neq \emptyset &\implies \text{Fill}(q, e) \neq \emptyset, \\ \forall (q, q') \in \Pi_{\Sigma_o}: \text{Fill}(q, e) \neq \emptyset \wedge \text{Fill}(q', e) \neq \emptyset &\implies \text{Fill}(q, e) \cap \text{Fill}(q', e) \neq \emptyset, \end{aligned}$$

and both are checked by scanning the two sets of pairs. The whole procedure is polynomial in n and $|\Sigma|$. $\square$

Theorem 8 corrects a claim made in [7], where it is asserted without proof that LOC verification is not easier for deterministic than for nondeterministic plants. Unless $P = PSPACE$, that assertion is false.

4 The Local Condition of Multi-Agent Control

LROC is a different property, which we discuss next.

Proposition 9. *LROC verification is in PSPACE, and decidable in polynomial time for DFAs.*

Proof. Let $G = (Q_G, \Sigma, \delta, I)$ be an NFA with all states final. By (1), LROC fails if and only if there are $s, s' \in L = L(G)$ with $P(s) = P(s')$ and $b, b' \in \Sigma_{uo}$ with $R(b) = R(b')$ such that $sb \in L$, $s'b' \in L$, and $s'b \notin L$. Since all states of G are final, these three memberships are determined by the state sets $X = \delta(I, s)$ and $X' = \delta(I, s')$: they say $\delta(X, b) \neq \emptyset$, $\delta(X', b') \neq \emptyset$, and $\delta(X', b) = \emptyset$. We call a pair (X, X') of subsets of Q_G *bad* if some $b, b' \in \Sigma_{uo}$ with $R(b) = R(b')$ satisfy these three conditions. Whether a given pair is bad is decided by inspecting at most $|\Sigma_{uo}|^2$ choices of (b, b') and applying δ , in time polynomial in the size of G .

By Lemma 4 applied with $\Xi = \Sigma_o$, the pairs (X, X') that arise from strings $s, s' \in L$ with $P(s) = P(s')$ are exactly the elements of Π_{Σ_o} . Hence LROC fails if and only if Π_{Σ_o} contains a bad pair. A nondeterministic procedure guesses a path in $G \otimes_{\Sigma_o} G$ from (I, I) one edge at a time, storing only the current pair and a step counter bounded by the number $2^{2|Q_G|}$ of pairs, and tests the pair reached for badness. A pair takes space $2|Q_G|$ and the counter $2|Q_G|$, and the procedure runs in nondeterministic space $O(|Q_G|)$; thus, LROC verification is in $NPSpace = PSPACE$ by Savitch's theorem.

If G is a DFA, then $I = \{q_0\}$ and $\delta(q_0, s)$ is a single state for every $s \in L$, and every element of Π_{Σ_o} is a pair of states; Π_{Σ_o} has at most $|Q_G|^2$ elements and is computed by reachability in a graph with $|Q_G|^2$ vertices. Testing each of its elements for badness is polynomial, and hence the whole procedure is polynomial. $\square$

The polynomial bound makes precise the observation of [6] that LROC “can be checked in a similar way as observability”. It does not survive nondeterminism.

Proposition 10. *LROC verification is PSPACE-hard for NFAs, even for a relabeling that is injective on the observable events and merges one pair of unobservable events.*

Proof. We reduce universality of NFAs with all states final over $\Delta = \{a, b\}$, which is PSPACE-complete [10]. Let A be such an NFA, then $L(A)$ is prefix-closed. Let $\Sigma = \Delta \dot{\cup} \{c, d_1, d_2\}$ with $\Sigma_o = \Delta$ and $\Sigma_{uo} = \{c, d_1, d_2\}$, and let R be injective on Δ , with $R(c) = \hat{c}$ and $R(d_1) = R(d_2) = \hat{d}$. Consider the language

$$L = \overline{c\Delta^*d_1} \cup \overline{c\Delta^*d_2} \cup \overline{\Delta^*d_2} \cup \overline{L(A)d_1},$$

which is prefix-closed and can obviously be generated by an NFA with all states final of size polynomial in the size of A .

Since R is injective on Σ except for the pair d_1, d_2 , the only instances of (1) with $b \neq b'$ are $(b, b') = (d_1, d_2)$ and $(b, b') = (d_2, d_1)$. Observations erase c, d_1, d_2 , and $P(w) = P(cw) = w$ for $w \in \Delta^*$.

For $(b, b') = (d_2, d_1)$, $s'd_1 \in L$ implies $s' \in c\Delta^* \cup L(A)$, and hence $s'd_2 \in L$ for every $s' \in L$ whose observation is some $w \in \Delta^*$, since both wd_2 and cwd_2 belong to L , and therefore it never violates (1). For $(b, b') = (d_1, d_2)$, $s'd_2 \in L$ restricts the s' to be of the form w or cw for some $w \in \Delta^*$. If $s' = cw$, then $s'd_1 = cwd_1 \in L$ and there is no violation. If $s' = w$, then $s'd_2 = wd_2 \in L$, the string $s = cw$ satisfies $P(s) = P(s')$ and $sd_1 = cwd_1 \in L$, and $s'd_1 = wd_1 \in L$ if and only if $w \in L(A)$. A violation therefore exists if and only if there is $w \in \Delta^* \setminus L(A)$, that is, if and only if $L(A) \neq \Delta^*$. $\square$

Theorem 11. *LROC verification is PSPACE-complete for NFAs and decidable in polynomial time for DFAs.*

Proof. Membership and the bound are shown in Proposition 9, and hardness is shown in Proposition 10. $\square$

5 Algorithms

Theorem 7 rules out a polynomial-time algorithm for LOC on nondeterministic plants, but the search of Proposition 5 is a reachability search over pairs of state sets and can be organized so that its exponential part is entered rarely and traversed lazily. This section describes two ingredients of such a procedure, in the order in which they should be applied: a polynomial test that is often conclusive (Section 5.1), and a precomputation that removes the language operations from the inner loop (Section 5.2).

5.1 A Polynomial-Time Sufficient Condition

Proposition 5 decides LOC by intersecting the languages $\text{Fill}(X, e)$ over pairs of subsets X reachable with a common observation, and it is the subsets that cost exponential space. They can be dispensed with, at the price of a stronger requirement, because Fill distributes over unions in its first argument.

Proposition 12. *Let $G = (Q_G, \Sigma, \delta, I)$ be an NFA with all states final. If*

$$\text{Fill}(q, e) \cap \text{Fill}(q', e) \neq \emptyset$$

for every $e \in \Sigma_c \cap \Sigma_{hi}$ and every pair of states $(q, q') \in \bigcup_{(X,Y) \in \Pi_{\Sigma_o}} X \times Y$, then $L(G)$ is LOC with respect to Q , P , and Σ_c .

Proof. $\text{Fill}(X, e) = \bigcup_{q \in X} \text{Fill}(q, e)$, since $\delta(X, ye) \neq \emptyset$ holds if and only if $\delta(q, ye) \neq \emptyset$ for some $q \in X$. Let $s, s' \in L$ with $P(s) = P(s')$ and let e satisfy $Q(s)e, Q(s')e \in Q(L)$. The sets $\delta(I, s)$ and $\delta(I, s')$ are nonempty, and we may pick $q \in \delta(I, s)$ and $q' \in \delta(I, s')$; the pair (q, q') is reachable in the synchronized product, since $P(s) = P(s')$. The hypothesis then gives $\text{Fill}(q, e) \cap \text{Fill}(q', e) \neq \emptyset$, and therefore $\text{Fill}(s, e) \cap \text{Fill}(s', e) \neq \emptyset$, and $L(G)$ is LOC by (2). $\square$

The pairs of states in Π_{Σ_o} are explored by reachability in a graph with $|Q_G|^2$ vertices; each $\text{Fill}(q, e)$ is recognized by an NFA of size polynomial in the size of G ; and emptiness of the intersection of two NFAs is decidable in nondeterministic logarithmic space. The whole test is polynomial in the size of G . $\square$

5.2 Compiling the Compatibility Relation

Proposition 5 searches over pairs of state sets and tests, at every node, whether two languages intersect. The test should not be repeated: it depends on the states involved, and can be compiled away. For $e \in \Sigma_c \cap \Sigma_{hi}$ define

$$\text{Comp}_e := \{(q, q') \in Q_G^2 : \text{Fill}(q, e) \cap \text{Fill}(q', e) \neq \emptyset\},$$

computable in polynomial time, being $|Q_G|^2$ emptiness tests for intersections of two NFAs of size $O(|G|)$. Since $\text{Fill}(X, e) = \bigcup_{q \in X} \text{Fill}(q, e)$, the condition $\text{Fill}(X, e) \cap \text{Fill}(Y, e) \neq \emptyset$ says that $X \times Y$ meets Comp_e , and at search time the test is a bitmask intersection and no automaton is built. Proposition 12 is the special case in which Comp_e contains all reachable pairs of states, and the polynomial test costs nothing beyond the compilation and should be run first.

6 Conclusion

The two local conditions of hierarchical and multi-agent supervisory control are PSPACE-complete to verify for nondeterministic plants, and both become polynomial in time as soon as the plant is deterministic. On the algorithmic side, the exponential search that decides the hierarchical condition should be preceded by a polynomial test that is often conclusive, and its inner language-intersection test should be compiled away rather than repeated.

References

- [1] H. Zhong, W. M. Wonham, On the consistency of hierarchical supervision in discrete-event systems, *IEEE Transactions on Automatic Control* 35 (10) (1990) 1125–1134.
- [2] K. C. Wong, W. M. Wonham, Hierarchical control of discrete-event systems, *Discrete Event Dynamic Systems* 6 (3) (1996) 241–273.
- [3] K. Schmidt, C. Breindl, Maximally permissive hierarchical control of decentralized discrete event systems, *IEEE Transactions on Automatic Control* 56 (4) (2011) 723–737.
- [4] O. Boutin, J. Komenda, T. Masopust, K. Schmidt, J. H. van Schuppen, Hierarchical control with partial observations: Sufficient conditions, in: *Proceedings of the IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, 2011, pp. 1817–1822.
- [5] Y. Liu, K. Cai, Z. Li, On scalable supervisory control of multi-agent discrete-event systems, *Automatica* 108 (2019) 108460.
- [6] Y. Liu, J. Komenda, Z. Li, Supervisory control of multiagent discrete-event systems with partial observation, *IEEE Control Systems Letters* 6 (2022) 1867–1872.
- [7] J. Komenda, T. Masopust, Conditions for hierarchical supervisory control under partial observation, in: *Workshop on Discrete Event Systems (WODES)*, Vol. 53 of *IFAC-PapersOnLine*, 2020, pp. 303–308, errata in the arxiv version, arXiv:2203.01444.
- [8] C. G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems*, 2nd Edition, Springer, 2008.
- [9] S. Arora, B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009.
- [10] J.-Y. Kao, N. Rampersad, J. Shallit, On NFAs where all states are final, initial, or both, *Theoretical Computer Science* 410 (47–49) (2009) 5010–5021.