Conditions for Hierarchical Supervisory Control under Partial Observation ★

Jan Komenda * **Tomáš Masopust** **,*

* *Institute of Mathematics of the Czech Academy of Sciences, Brno, Czechia*
** *Faculty of Science, Palacky University, Olomouc, Czechia*
(e-mails: komenda@ipm.cz, masopust@math.cas.cz)

Abstract: The fundamental problem in hierarchical supervisory control under partial observation is to find conditions preserving observability between the original (low-level) and the abstracted (high-level) plants. Two conditions for observable specifications were identified in the literature – *observation consistency* (OC) and *local observation consistency* (LOC). However, the decidability of OC and LOC were left open. We show that both OC and LOC are decidable for regular systems. We further show that these conditions do not guarantee that supremal (normal or relatively observable) sublanguages computed on the low level and on the high level always coincide. To solve the issue, we suggest a new condition – *modified observation consistency* – and show that under this condition, the supremal normal sublanguages are preserved between the levels, while the supremal relatively observable high-level sublanguage is at least as good as the supremal relatively observable low-level sublanguage, i.e., the high-level solution may be even better than the low-level solution.

Keywords: Discrete-event system, Hierarchical supervisory control, Normality, Relative observability

★ Partially supported by RVO 67985840 and by the GAČR grant GC19-06175J.

ERRATA

Several results of this paper are not correct as stated. This section contains errata based on the revised and extended version of this conference paper.[1] No changes are made in the following sections.

(1) The statement of Theorem 5, claiming that the verification of (modified) observation consistency is PSpace-complete for NFAs, is unproved. In particular, the membership in PSpace is not shown. The problem is PSpace-hard, but it is open whether the verification of (M)OC is decidable; in particular, it is open whether the problem is in PSpace.

(2) The statement of Theorem 11 is not precise. The correct statement is:

Theorem 11. For a nonblocking DFA G, let $L = L(G)$ and $L_m = L_m(G)$. If L is MOC with respect to Q, P, and P_{hi}, then for every high-level specification $K \subseteq Q(L_m)$,

$$\mathrm{supN}\big(K \| L_m, L, P\big) = \mathrm{supN}(K, Q(L), P_{hi}) \parallel L_m$$

whenever $\mathrm{supN}(K, Q(L), P_{hi})$ and L_m are nonconflicting.

(3) As stated, Theorem 15 is incorrect.

Theorem 15. Assume that each shared event is high level and observable, i.e., $\Sigma_s \subseteq \Sigma_{hi} \cap \Sigma_o$. If, for $i = 1, \dots, n$, L_i is MOC wrt Q_i, P^i_{loc}, and $P^i_{loc|hi}$, then $\|_{i=1}^{n} L_i$ is MOC wrt Q, P, and P_{hi}.

A counterexample: Let $L_1 = \overline{\{h_1 o_1 x\}}$ and $L_2 = \overline{\{o_2 h_2 x\}}$ be two languages with observable events $\Sigma_o = \{o_1, o_2, x\}$ and high-level events $\Sigma_{hi} = \{h_1, h_2, x\}$; that is, x is the only shared event, which is both observable and high-level. Both languages satisfy MOC. However, for $s = h_1 o_1 o_2 h_2 x$ and $t = h_2 h_1 x$ satisfying $P_{hi}(Q(s)) = P_{hi}(h_2 h_1 x) = x = P_{hi}(h_1 h_2 x) = P_{hi}(t)$, there is no $s' \in L_1 \| L_2 = h_1 o_1 x \| o_2 h_2 x$ such that $P(s') = o_1 o_2 x$ and $Q(s') = h_2 h_1 x$, because the only string containing x with the order of observable events $o_1 o_2$ is $h_1 o_1 o_2 h_2 x$, but it does not have the required order of the local high-level events h_1 and h_2.

[1] Komenda, J., Masopust, T., Hierarchical Supervisory Control under Partial Observation: Normality, 2023, https://doi.org/10.48550/arXiv.2203.01444

1. INTRODUCTION

Organizing systems into hierarchical structures is a common engineering practice used in manufacturing, robotics, or artificial intelligence to overcome the combinatorial state explosion problem. Hierarchical supervisory control of discrete-event systems (DES) was introduced by Zhong and Wonham (1990b) as a two-level vertical decomposition of the system. The low-level plant modeling the system behavior is restricted by a high-level specification, and the aim is to synthesize a nonblocking and optimal supervisor based on the high-level abstraction of the plant in such a way that it can be used for a low-level implementation. They identified a sufficient condition to achieve the goal. Zhong and Wonham (1990a) extended the framework to hierarchical coordination control and developed an abstract hierarchical supervisory control theory. Wong and Wonham (1996b) applied the theory to the Brandin-Wonham framework of timed DES. Schmidt et al. (2008) extended hierarchical supervisory control to decentralized systems, and Schmidt and Breindl (2011) found weaker sufficient conditions for maximal permissiveness of high-level supervisors with complete observations. Recently, Baier and Moor (2015) generalized hierarchical supervisory control to the Büchi framework, where the plant and the specification are represented by ω-languages.

Motivated by abstractions of hybrid systems to DES, Hubbard and Caines (2002) developed a hierarchical control theory for DES based on state aggregation, and Torrico and Cury (2002) investigated a hierarchical control approach where the low level is in the Ramadge-Wonham framework and the high level is obtained by state aggregation. Here, the high-level events are subsets of low-level events, and advanced control structures are used to synthesize a controller. Furthermore, da Cunha and Cury (2007) proposed hierarchical supervisory control for DES where the low level is in the Ramadge-Wonham framework and the high level is represented by systems with flexible marking, in order to simplify the modeling of the high level. Ngo and Seow (2014, 2018) investigated hierarchical control for Moore automata and for timed DES, and Sakakibara and Ushio (2018) considered concurrent DES modeled by Mealy automata.

Fekri and Hashtrudi-Zad (2009) first considered hierarchical supervisory control of partially observed DES. They used Moore automata models and defined controllable and observable events based on vocalization. Hence, they need a specific definition of the low-level supervisor. Furthermore, their approach is monolithic, while ours allows distributed synthesis using the standard synchronous composition of the plant with the supervisor.

In this paper, we adapt the classical hierarchical supervisory control of DES in the Ramadge-Wonham framework, where the systems are modeled as DFAs and the abstraction is modeled as a natural projection, i.e., the behavior of the high-level plant is the projection of the behavior of the low-level plant to the high-level alphabet. The problem is then as follows. Given a low-level plant G over an alphabet Σ modeling the system behavior and a high-level specification language K over a high-level alphabet $\Sigma_{hi} \subseteq \Sigma$. The low-level plant G is abstracted to the high-level plant G_{hi} describing the high-level behavior. The aim is to synthesize a nonblocking and optimal supervisor S_{hi} on the high level in such a way that it can be used for a construction of a low-level supervisor S that is nonblocking and optimal wrt the specification $K \| L_m(G)$.

To achieve the goal for fully observed DES, important concepts have been developed in the literature, including the *observer property* of Wong and Wonham (1996a), *output control consistency* (OCC) of Zhong and Wonham (1990b), and *local control consistency* (LCC) of Schmidt and Breindl (2011). These concepts are sufficient for the high-level synthesis of a nonblocking and optimal supervisor to have a low-level implementation.

However, the conditions are not sufficient for partially observed DES. The sufficient condition of Komenda and Masopust (2010) requires that all observable events must be high-level events, which is a very restrictive assumption. Therefore, Boutin et al. (2011) investigated weaker and less restrictive conditions, and introduced two concepts – *local observation consistency* (LOC) and *observation consistency* (OC). The latter ensures a certain consistency between observations on the high level and the low level, and the former is an extension of the observer property to partial observation. The paper shows that, for observable specifications, projections that satisfy OC, LOC, LCC, and that are observers are suitable for the nonblocking least restrictive hierarchical supervisory control under partial observation. The fundamental question whether the properties of OC and LOC are decidable is left open.

In this paper, we first show that checking OC and LOC properties is decidable for systems with regular behaviors and that the problems are actually PSPACE-complete (Theorems 5 and 6).

Then we show that OC and LOC are not sufficient to preserve optimality for non-observable specifications. These are specifications, for which a suitable supremal sublanguage (normal or relatively observable) needs to be computed. We show that OC and LOC do not guarantee that the supremal normal (relatively observable) low-level sublanguage coincides with the composition of the plant and the supremal normal (relatively observable) high-level sublanguage (Example 8).

For normality, we suggest a condition of *modified observation consistency* (MOC) and show that it preserves optimality, i.e., the supremal normal sublanguages are preserved between the levels (Definition 9 and Theorem 11). Then we discuss two special cases often considered in the literature: (i) the case where all observable events are also high-level events, and (ii) the case where all high-level events are also observable. Our new results generalize the previously known results.

For relative observability, we show that MOC ensures that the high-level solution is at least as good as the low-level solution (Theorem 13). In particular, the low-level implementation of the high-level solution may be better than what we can obtain directly on the low level (Example 12). This observation makes relative observability an interesting and suitable notion for hierarchical supervisory control.

Finally, the newly suggested condition of MOC is stronger than OC of Boutin et al. (2011) as shown in Lemma 10. Moreover, similarly as OC, the MOC condition is structural only wrt the plant. We discuss the complexity of MOC in Theorem 14, and show that it is compositional in Theorem 15.

All the missing proofs can be found in the appendix.

2. PRELIMINARIES AND DEFINITIONS

We assume that the reader is familiar with the basics of supervisory control, see Cassandras and Lafortune (2008). For a set A, $|A|$ denotes the cardinality of A. For an alphabet (finite nonempty set) Σ, Σ^* denotes the set of all finite strings over Σ; the empty string is denoted by ε. The alphabet Σ is partitioned into *controllable events* Σ_c and *uncontrollable events* $\Sigma_u = \Sigma \setminus \Sigma_c$ as well as into *observable events* Σ_o and *unobservable events* $\Sigma_{uo} = \Sigma \setminus \Sigma_o$. A language is a subset of Σ^*. For a language $L \subseteq \Sigma^*$, the prefix closure $\overline{L} = \{w \in \Sigma^* \mid wv \in L\}$; L is prefix-closed if $L = \overline{L}$.

A *(natural) projection* $R\colon \Sigma^* \to \Gamma^*$, where $\Gamma \subseteq \Sigma$ are alphabets, is a homomorphism for concatenation defined so that $R(a) = \varepsilon$ for $a \in \Sigma \setminus \Gamma$, and $R(a) = a$ for $a \in \Gamma$. The action of R on $w \in \Sigma^*$ is to remove all events from w that are not in Γ. The inverse image of $w \in \Gamma^*$ under R is the set $R^{-1}(w) = \{s \in \Sigma^* \mid R(s) = w\}$. These definitions can naturally be extended to languages.

A *nondeterministic finite automaton* (NFA) is a quintuple $G = (Q, \Sigma, \delta, I, F)$, where Q is a finite set of states, Σ is an input alphabet, $I \subseteq Q$ is a set of initial states, $F \subseteq Q$ is a set of marked states, and $\delta\colon Q \times \Sigma \to 2^Q$ is the transition function that can be extended to the domain $2^Q \times \Sigma^*$ in the usual way. The automaton G is *deterministic* (DFA) if $|I| = 1$, and $|\delta(q, a)| = 1$ for every state $q \in Q$ and every event $a \in \Sigma$. The language *generated* by G is the set $L(G) = \{w \in \Sigma^* \mid \delta(q_0, w) \in Q\}$, and the language *marked* by G is the set $L_m(G) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$. By definition, $L_m(G) \subseteq L(G)$, and $L(G)$ is prefix-closed. If $\overline{L_m(G)} = L(G)$, then G is *nonblocking*.

Let $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$ be languages. The *parallel composition of* L_1 *and* L_2 is the language $L_1 \| L_2 = P_1^{-1}(L_1) \cap P_2^{-1}(L_2)$, where $P_i\colon (\Sigma_1 \cup \Sigma_2)^* \to \Sigma_i^*$ is a projection, for $i = 1, 2$; see Cassandras and Lafortune (2008) for a definition for automata. For two DFAs G_1 and G_2, $L(G_1 \| G_2) = L(G_1) \| L(G_2)$. Languages L_1 and L_2 are *synchronously nonconflicting* if $\overline{L_1 \| L_2} = \overline{L_1} \| \overline{L_2}$.

Let G be a DFA over an alphabet Σ. A language $K \subseteq L_m(G)$ is *controllable* wrt $L(G)$ and the set of uncontrollable events Σ_u if $\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K}$; K is *observable* wrt $L(G)$, the set of observable events Σ_o with $P\colon \Sigma^* \to \Sigma_o^*$ being the corresponding projection, and the set of controllable events Σ_c if for all $s, s' \in L(G)$ with $P(s) = P(s')$ and for every $e \in \Sigma_c$, if $se \in \overline{K}$, $s'e \in L(G)$, and $s' \in \overline{K}$, then $s'e \in \overline{K}$. Algorithms to verify controllability and observability can be found in Cassandras and Lafortune (2008).

It is known that there is no supremal observable sublanguage. Therefore, stronger properties, such as normality of Lin and Wonham (1988) or relative observability of Cai et al. (2015), are used for specifications that are not observable. Language $K \subseteq L_m(G)$ is *normal* wrt $L(G)$ and the projection $P\colon \Sigma^* \to \Sigma_o^*$ if $\overline{K} = P^{-1}[P(\overline{K})] \cap L(G)$. Relative observability has recently been introduced by Cai et al. (2015) and further studied by Alves et al. (2017) as a condition weaker than normality and stronger than observability. Let $K \subseteq C \subseteq L_m(G)$ be languages. Language K is *relatively observable* wrt C, G, and $P\colon \Sigma^* \to \Sigma_o^*$ (or simply *C-observable*) if for all strings $s, s' \in \Sigma^*$ with $P(s) = P(s')$ and for every $e \in \Sigma$, whenever $se \in \overline{K}$, $s'e \in L(G)$, and $s' \in \overline{C}$, then $s'e \in \overline{K}$. For $C = K$, the definition coincides with observability.

A *decision problem* is a yes-no question. A decision problem is *decidable* if there exists an algorithm that solves the problem. Complexity theory classifies decidable problems to classes based on the time or space an algorithm needs to solve the problem. The complexity class we consider in this paper is PSpace, denoting all problems solvable by a deterministic polynomial-space algorithm. A decision problem is PSpace-complete if the problem belongs to PSpace (*membership*) and every problem from PSpace can be reduced to the problem by a polynomial-time algorithm (*hardness*). It is unknown whether PSpace-complete problems can be solved in polynomial time.

3. PRINCIPLES OF HIERARCHICAL CONTROL

In the sequel, we use the following notation for projections and abstractions, see the commutative diagram in Fig. 1. Let Σ be the low-level alphabet, $\Sigma_{hi} \subseteq \Sigma$ the high-level alphabet, and $\Sigma_o \subseteq \Sigma$ the set of observable events. Let $P\colon \Sigma^* \to \Sigma_o^*$ be the projection corresponding to system's partial observation, $Q\colon \Sigma^* \to \Sigma_{hi}^*$ the projection corresponding to the high-level abstraction, and $P_{hi}\colon \Sigma_{hi}^* \to (\Sigma_{hi} \cap \Sigma_o)^*$ and $Q_o\colon \Sigma_o^* \to (\Sigma_{hi} \cap \Sigma_o)^*$ the corresponding observations and abstractions.

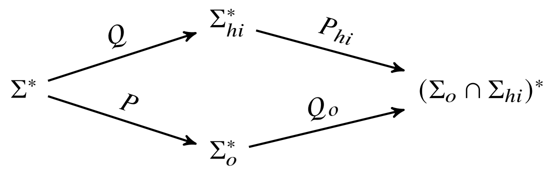

Fig. 1. Commutative diagram of abstractions and projections.

We now state the hierarchical supervisory control problem for partially observed DES.

Problem 1. Let G be a low-level plant over an alphabet Σ, and let K be a high-level specification over an alphabet $\Sigma_{hi} \subseteq \Sigma$. The abstracted high-level plant G_{hi} is defined over the alphabet Σ_{hi} so that $L(G_{hi}) = Q(L(G))$ and $L_m(G_{hi}) = Q(L_m(G))$. The aim of hierarchical supervisory control is to determine, based on the high-level plant G_{hi} and the specification K, without using the low-level plant G, a nonblocking low-level supervisor S such that $L_m(S/G) = K \| L_m(G)$. ◇

Boutin et al. (2011) identified sufficient conditions (observation consistency and local observation consistency) on the low-level plant G for which observability of $K \| L_m(G)$ wrt G is equivalent to observability of K wrt the high-level plant G_{hi}.

A prefix-closed language $L \subseteq \Sigma^*$ is *observation consistent* (OC) wrt projections Q, P, and P_{hi} if for all strings $t, t' \in Q(L)$ such that $P_{hi}(t) = P_{hi}(t')$, there are $s, s' \in L$ such that $Q(s) = t$, $Q(s') = t'$, and $P(s) = P(s')$. Intuitively, any two strings of the high-level plant with the same observation have corresponding strings with the same observation in the low-level plant.

A prefix-closed language $L \subseteq \Sigma^*$ is *locally observation consistent* (LOC) wrt projections Q and P and the set of controllable events Σ_c if for all strings $s, s' \in L$ and all events $e \in \Sigma_c \cap \Sigma_{hi}$ such that $Q(s)e, Q(s')e \in Q(L)$ and $P(s) = P(s')$, there exist low-level strings $u, u' \in (\Sigma \setminus \Sigma_{hi})^*$ such that $P(u) = P(u')$ and $sue, s'u'e \in L$. Intuitively, continuing two observationally equivalent high-level strings by the same controllable event, the corresponding low-level observationally equivalent strings can be continued by this same event in the original plant in the future (after possible empty low-level strings with the same observations). LOC can be seen as a specialization of the observer property and LCC for partially observed DES.

Besides observability, Problem 1 further requires the preservation of controllability between the levels. It has been previously achieved by the conditions of $L_m(G)$*-observer* of Wong and

Wonham (1996a) and *output control consistency* of Zhong and Wonham (1990b), or its weaker variant, *local control consistency* of Schmidt and Breindl (2011). Formally, projection $Q\colon \Sigma^* \to \Sigma_{hi}^*$ is an $L_m(G)$*-observer* for a nonblocking plant G over Σ if for all strings $t \in Q(L_m(G))$ and $s \in \overline{L(G)}$, if $Q(s)$ is a prefix of t, then there exists $u \in \Sigma^*$ such that $su \in L_m(G)$ and $Q(su) = t$. We say that Q is *locally control consistent* (LCC) for a string $s \in L(G)$ if for all $e \in \Sigma_{hi} \cap \Sigma_u$ such that $Q(s)e \in L(G_{hi})$, either there is no $u \in (\Sigma \setminus \Sigma_{hi})^*$ such that $sue \in L(G)$ or there is $u \in (\Sigma_u \setminus \Sigma_{hi})^*$ such that $sue \in L(G)$. We call Q LCC for a language $M \subseteq L(G)$ if Q is LCC for every $s \in M$.

Notice that the conditions are structural and hold for any specification once the plant is fixed. The following result formulates a solution to Problem 1.

Theorem 2. (Boutin et al. (2011)). Let G be a nonblocking DFA over Σ, and let $K \subseteq Q(L_m(G))$ be a (high-level) specification. Let Q be LCC for $L(G)$ and Σ_u, and an $L_m(G)$-observer. Let $L(G)$ be OC wrt Q, P, and P_{hi}, and LOC wrt Q, P, and Σ_c. Then K is controllable wrt $Q(L(G))$ and $\Sigma_u \cap \Sigma_{hi}$, and observable wrt $Q(L(G))$, $\Sigma_o \cap \Sigma_{hi}$, and $\Sigma_c \cap \Sigma_{hi}$ if and only if $K \| L_m(G)$ is controllable wrt $L(G)$ and Σ_u, and observable wrt $L(G)$, Σ_o, and Σ_c. □

Theorem 2 allows to verify the existence of a supervisor realizing a high-level specification K for a given system G, under the aforementioned properties, based on the abstraction G_{hi}. Namely, if there is a nonblocking supervisor S_{hi} such that $L_m(S_{hi}/G_{hi}) = K$, then there is a nonblocking supervisor S such that $L_m(S/G) = K \| L_m(G)$. In particular, a DFA realization G_K of K such that $L_m(G_K) = K$ can be used to implement the supervisor in the form $G_K \| G$.

Considering only observability, the following results hold.

Theorem 3. (Boutin et al. (2011)). Let G be a nonblocking DFA over Σ, and let $K \subseteq Q(L_m(G))$ be a specification. Assume that $L(G)$ is OC wrt Q, P, and P_{hi}, that K and $L_m(G)$ are synchronously nonconflicting, and that $L(G)$ is LOC wrt Q, P, and Σ_c. Then K is observable wrt $Q(L(G))$, $\Sigma_{hi} \cap \Sigma_o$, and $\Sigma_{hi} \cap \Sigma_c$ if and only if $K \| L_m(G)$ is observable wrt $L(G)$, Σ_o, and Σ_c. □

If all controllable events are observable, observability is equivalent to normality, and OC is sufficient to preserve observability.

Corollary 4. (Boutin et al. (2011)). Let G be a nonblocking DFA, and let $K \subseteq Q(L_m(G))$ be a specification. If $L(G)$ is OC wrt Q, P, and P_{hi}, and K and $L_m(G)$ are synchronously nonconflicting, then K is normal wrt $Q(L(G))$ and P_{hi} if and only if $K \| L_m(G)$ is normal wrt $L(G)$ and P. □

We now show that a result similar to Theorem 3 does not hold for relative observability without additional assumptions; namely, if K is C-observable, then $K \| L_m(G)$ is not necessarily $C \| L(G)$-observable. Let $K = \{\varepsilon, a\}$, $C = \{\varepsilon, a, au\}$ over $\Sigma_{hi} = \{a, u\}$, and $L(G) = \{\varepsilon, a, ae, au, aue\}$ over $\Sigma = \{a, u, e\}$ be prefix-closed languages, and hence synchronously nonconflicting. Let $\Sigma_o = \{a, e\}$. It can be verified that $L(G)$ is OC and LOC, and that K is C-observable wrt $Q(L(G)) = C$, and hence observable. However, $K \| L(G)$ is not $C \| L(G)$-observable, since $ae \in K \| L(G)$, $au \in C \| L(G)$, and $aue \in L(G)$, but $aue \notin K \| L(G)$ (but $K \| L(G)$ is observable by Theorem 3).

4. VERIFICATION OF OBSERVATION CONSISTENCY

In this section, we show that the verification of OC is PSpace-complete, and hence decidable, for systems modeled by finite automata. The same problem for LOC is treated in the next section.

Theorem 5. Verifying OC for systems modeled by NFAs is PSpace-complete.

Proof. To prove membership in PSpace, we generalize the parallel composition to a set of synchronizing events. Let Σ be an alphabet, and let $L_1, L_2 \subseteq \Sigma^*$ be languages of NFAs $G_1 = (Q_1, \Sigma, \delta_1, I_1, F_1)$ and $G_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$, respectively. Let $\Sigma' \subseteq \Sigma$ be a set of synchronizing events. The parallel composition of L_1 and L_2 synchronized on the events of Σ' is denoted by $L_1 \|_{\Sigma'} L_2$ and defined as the language of the NFA

$$G_1 \|_{\Sigma'} G_2 = (Q_1 \times Q_2, (\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\}), \delta, I_1 \times I_2, F_1 \times F_2)\,,$$

where the alphabet is a set of pairs based on the synchronization of events in Σ'. There are two categories of pairs to construct, corresponding to (a) events in Σ', and (b) events in $\Sigma \setminus \Sigma'$. For every $a \in \Sigma'$, we have the pair (a, a), and for every $a \in \Sigma \setminus \Sigma'$, we have two pairs (a, ε) and (ε, a). The transition function $\delta\colon (Q_1 \times Q_2) \times ((\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\})) \to Q_1 \times Q_2$ is defined on these event pairs as follows:

- for $a \in \Sigma'$, $\delta((p, q), (a, a)) = \delta_1(p, a) \times \delta_2(q, a)$;
- for $a \in \Sigma \setminus \Sigma'$, $\delta((p, q), (a, \varepsilon)) = \delta_1(p, a) \times \{q\}$ and $\delta((p, q), (\varepsilon, a)) = \{p\} \times \delta_2(q, a)$;
- undefined otherwise.

For simplicity, a sequence of event pairs, $(a_1, \varepsilon)(a_2, a_2)(\varepsilon, a_3)$, is written as a pair of the concatenated components $(a_1 a_2, a_2 a_3)$. Then we can say that the language consists of pairs of strings of the form (w, w'), where w and w' coincide on the letters of Σ', that is, $P'(w) = P'(w')$ for the projection $P'\colon \Sigma^* \to \Sigma'^*$.

Let $L \subseteq \Sigma^*$ be a prefix-closed language, and let Σ_o and Σ_{hi} be the respective observation and high-level alphabets. We show that L is OC wrt Q, P, and P_{hi} if and only if

$$Q(L) \;\|_{\Sigma_{hi} \cap \Sigma_o}\; Q(L) \subseteq Q\left(L \;\|_{\Sigma_o}\; L\right)\,,$$

where, for an event (a, b), $Q(a, b) = (Q(a), Q(b))$. Membership in PSpace then follows, since we can express $Q(L)$, as well as $Q(L \|_{\Sigma_o} L)$, as NFAs, and the inclusion of two NFAs can be verified in PSpace, see Clemente and Mayr (2019).

The intuition behind the equivalence is to couple all strings $t, t' \in Q(L)$ with the same high-level observations, which are exactly the pairs $(t, t') \in Q(L) \|_{\Sigma_{hi} \cap \Sigma_o} Q(L)$, and to verify that for every such pair there are strings $s, s' \in L$ with the same observations, which are exactly the pairs $(s, s') \in L \|_{\Sigma_o} L$, that are abstracted to the pair (t, t'), that is, they satisfy $(Q(s), Q(s')) = (t, t')$.

The rest of the proof can be found in the appendix. □

By a slight modification of the proof, it can be shown that the problem is not easier for DFAs, that is, it remains PSpace-hard even for DFA models. We leave this proof for the full version.

5. VERIFICATION OF LOCAL OBSERVATION CONSISTENCY

In this section, we study decidability and complexity of LOC. As in the case of OC, the problem is not easier for DFA models.

The proof is again left for the full version. A proof sketch of the following theorem can be found in the appendix.

Theorem 6. Verification of LOC for systems modeled by NFAs is PSPACE-complete.

6. PRESERVATION OF SUPREMALITY

Problem 1 requires that the specification language K is achievable by the supervisor, i.e., K is observable. However, this is not always the case. If K is not observable, a common approach is to find a suitable sublanguage of K that is observable. Since there is no supremal observable sublanguage, the supremal normal sublanguage or the supremal relatively observable sublanguage is computed instead. The problem is now formulated as follows.

Problem 7. Given a low-level plant G over Σ and a high-level specification K over $\Sigma_{hi} \subseteq \Sigma$. The abstracted high-level plant G_{hi} over Σ_{hi} is defined so that $L(G_{hi}) = Q(L(G))$ and $L_m(G_{hi}) = Q(L_m(G))$. The aim is to determine a maximally permissive nonblocking supervisor S such that $L_m(S/G) \subseteq K \| L_m(G)$ using the abstraction G_{hi}. That is, if a maximally permissive nonblocking supervisor S_{hi} exists for the abstracted plant such that $L_m(S_{hi}/G_{hi}) \subseteq K$, then a maximally permissive nonblocking supervisor S exists such that $L_m(S/G) \subseteq K \| L_m(G)$.

Compared to Corollary 4 saying that under the OC condition the specification K is normal if and only if $K \| L_m(G)$ is normal, the following example shows that OC is not sufficient to preserver normality (relative observability) if the supremal normal (relatively observable) sublanguage of the specification K is a strict sublanguage of K. The problem is that it is not true that every supremal normal (relatively observable) sublanguage of $K \| L_m(G)$ is of the form $X \| L_m(G)$ for some convenient language $X \subseteq K$, and hence there may be no X that would be the supremal normal sublanguage of K.

Before stating the example, we introduce the following notation. For a prefix-closed language L and a specification $K \subseteq L$, we write $\mathrm{supN}(K, L)$ (resp. $\mathrm{supRO}(K, L)$) to denote the supremal normal (resp. the supremal relatively observable) sublanguage of K wrt L and the corresponding set of observable events.

Example 8. Let $\Sigma = \{a, b, c\}$ with $\Sigma_o = \{a, c\}$ and $\Sigma_{hi} = \{b, c\}$, and let $L = \{\varepsilon, a, b, c, ba, ac, bac\}$ and $K = \{\varepsilon, b, c\} \subseteq Q(L) = \{\varepsilon, b, c, bc\}$. To show that L is OC, notice that $P_{hi}(\varepsilon) = \varepsilon = P_{hi}(b)$ and $P_{hi}(c) = c = P_{hi}(bc)$, and hence we have two cases: (i) $t = \varepsilon$ and $t' = b$, and (ii) $t = c$ and $t' = bc$. Case (i) is trivial because we can choose $s = t = \varepsilon$ and $s' = t' = b$, which clearly satisfies OC. For case (ii), we choose $s = ac$ and $s' = bac$. Then, $Q(s) = c = t$, $Q(s') = bc = t'$, and $P(s) = ac = P(s')$. Thus, L is OC.

To compute the supremal normal sublanguages, we use the formula of Brandt et al. (1990) stating that $\mathrm{supN}(B, M) = B - P^{-1}P(M - B)\Sigma^*$, for prefix-closed languages $B \subseteq M \subseteq \Sigma^*$, and we obtain the following: $K \| L = a^*ba^* \cup a^*ca^* \cup a^* \cap L = \{\varepsilon, a, b, c, ba, ac\}$, $L - K\|L = \{bac\}$, and $P^{-1}P(bac) = P^{-1}(ac) = b^*ab^*cb^*$. This gives that $c \in \mathrm{supN}(K\|L, L) = K\|L - P^{-1}P(L - K\|L)\Sigma^* = \{\varepsilon, a, b, c, ba\}$. On the other hand, $Q(L) - K = \{\varepsilon, b, c, bc\} - \{\varepsilon, b, c\} = \{bc\}$, $P_{hi}(bc) = c$, and $P_{hi}^{-1}(c) = b^*cb^*$, which gives that $c \notin \mathrm{supN}(K, Q(L))\|L = Q^{-1}(K - P_{hi}^{-1}P_{hi}(Q(L) - K)\Sigma_{hi}^*) \cap L = Q^{-1}(\{\varepsilon, b\}) \cap L = \{\varepsilon, a, b, ba\}$ showing that OC is not a sufficient condition to preserve supremal normal sublanguages.

Inspecting further the example, the reader may verify that the computed supremal normal sublanguages coincide with the supremal relatively observable sublanguages for the choice of $C = K$. Therefore, the example also illustrates that OC is neither a sufficient condition to preserve supremal relatively observable sublanguages. ◇

To preserver the properties for supremal sublanguages, we modify the condition of OC by fixing one of the components.

Definition 9. A prefix-closed language $L \subseteq \Sigma^*$ is *modified observation consistent* (MOC) wrt projections Q, P, and P_{hi} if for every $s \in L$ and every $t' \in Q(L)$ such that $P_{hi}(Q(s)) = P_{hi}(t')$, there exists $s' \in L$ such that $P(s) = P(s')$ and $Q(s') = t'$.

MOC is a stronger property than OC. Indeed, if L is MOC, then for any $t, t' \in Q(L)$ with $P_{hi}(t) = P_{hi}(t')$, we have that $t = Q(s)$ for some $s \in L$, and hence there exists $s' \in L$ such that $P(s) = P(s')$ and $Q(s') = t'$, which shows that L is OC. This proves the following observation.

Lemma 10. MOC implies OC. □

6.1 Normality

We now show that MOC guarantees the preservation of normality for supremal sublanguages.

Theorem 11. Let G be a nonblocking DFA, and let $K \subseteq Q(L_m(G))$ be a specification. If $L(G)$ is MOC wrt Q, P, and P_{hi}, and K and $L_m(G)$ are synchronously nonconflicting, then

$$\mathrm{supN}\big(K\|L_m(G), L(G)\big) = \mathrm{supN}(K, Q(L(G))) \parallel L_m(G)\,.$$

Proof. ($\supseteq$): Since $\mathrm{supN}(K, Q(L(G)))$ is normal wrt $Q(L(G))$ and P_{hi}, Corollary 4 implies that $\mathrm{supN}(K, Q(L(G))) \parallel L_m(G)$ is normal wrt $L(G)$ and P. The implication that normality of K implies normality of $K\|L_m(G)$ in Corollary 4 holds without any assumptions. Therefore, $\mathrm{supN}\big(K, Q(L(G))\big) \parallel L_m(G) \subseteq \mathrm{supN}\big(K\|L_m(G), L(G)\big)$.

($\subseteq$): Let $S \subseteq K\|L_m(G)$ be normal wrt $L(G)$ and P, that is, $\overline{S} = P^{-1}P(\overline{S}) \cap L(G)$. Then, $Q(S) \subseteq K \cap Q(L_m(G)) = K$. We show that $Q(S)$ is normal wrt $Q(L(G))$ and P_{hi}, i.e., that $Q(\overline{S}) = P_{hi}^{-1}P_{hi}(Q(\overline{S})) \cap Q(L(G))$. To do this, let $s \in \overline{S}$ and $t' \in Q(L(G))$ be such that $P_{hi}(Q(s)) = P_{hi}(t')$, that is, $t' \in P_{hi}^{-1}P_{hi}(Q(\overline{S})) \cap Q(L(G))$. We show that $t' \in Q(\overline{S})$. By MOC, there exists $s' \in L(G)$ such that $Q(s') = t'$ and $P(s) = P(s')$, i.e., $s' \in P^{-1}P(s) \cap L(G) \subseteq P^{-1}P(\overline{S}) \cap L(G) = \overline{S}$, and hence $t' = Q(s') \in Q(\overline{S})$, which shows normality of $Q(S)$. □

Two special cases are often considered in the literature: (i) $\Sigma_o \subseteq \Sigma_{hi}$, and (ii) $\Sigma_{hi} \subseteq \Sigma_o$. We show that both imply MOC, and hence OC. Consequently, Theorem 11 strengthens the result of Komenda and Masopust (2010) showing that for any prefix-closed languages $L \subseteq \Sigma^*$ and $K \subseteq Q(L)$, if $\Sigma_o \subseteq \Sigma_{hi}$, then $\mathrm{supN}(K, Q(L)) \parallel L = \mathrm{supN}(K\|L, L)$.

First, assume that $\Sigma_o \subseteq \Sigma_{hi}$. Then $P = P_{hi}Q$, since Q_o is an identity. Let $s \in L$ and $t' \in Q(L)$ be such that $P_{hi}(Q(s)) = P_{hi}(t')$. Consider any $s' \in L$ with $Q(s') = t'$; such s' exists because $t' \in Q(L)$. Then, $P(s) = P_{hi}(Q(s)) = P_{hi}(t') = P_{hi}(Q(s')) = P(s')$, which was to be shown.

Second, assume that $\Sigma_{hi} \subseteq \Sigma_o$. Then, P_{hi} is an identity, and hence for any $s \in L$ and $t' \in Q(L)$ satisfying $P_{hi}(Q(s)) = P_{hi}(t')$, we have $Q(s) = P_{hi}(Q(s)) = P_{hi}(t') = t'$, i.e., we can chose $s' = s$ in the definition of MOC.

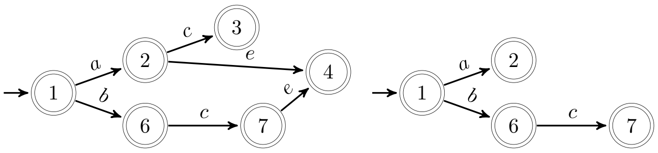

Fig. 2. Plant G and a specification K

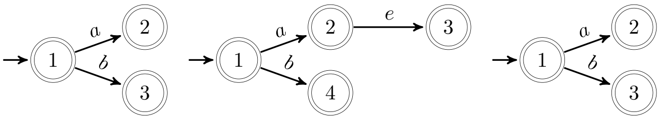

Fig. 3. Languages $\mathrm{supRO}(K, Q(L(G)))$, $\mathrm{supRO}(K, Q(L(G))) \parallel L_m(G)$, and $\mathrm{supRO}(K \| L_m(G), L(G))$, respectively

6.2 Relative Observability

We now show that an analogy of Theorem 11 does not hold for relative observability. In particular, the inclusion

$$\mathrm{supRO}\big(K \| L_m(G), L(G)\big) \supseteq \mathrm{supRO}(K, Q(L(G))) \parallel L_m(G)$$

does *not* hold in general as shown in the following example.

Example 12. Let the low-level plant and the high-level specification be defined by automata in Fig. 2. Let $\Sigma_{hi} = \{a, b, c\}$ and $\Sigma_o = \{e\}$. Then $\mathrm{supRO}(K, Q(L(G)))$ is shown in Fig. 3 as well as $\mathrm{supRO}(K, Q(L(G))) \parallel L_m(G)$. There, the reader can also see the supremal relatively observable sublanguage of $K \| L_m(G)$ wrt $K \| L_m(G)$, $L(G)$, and P, which obviously does not include $\mathrm{supRO}(K, Q(L(G))) \parallel L_m(G)$. $\diamond$

By Theorem 3, $\mathrm{supRO}(K, Q(L(G))) \parallel L_m(G)$ is always observable. It is thus an interesting question under which conditions the opposite inclusion holds. In other words, under which conditions is the low-level implementation of the high-level supervisor at least as good as the low-level supervisor? We now show that MOC is such a condition.

Theorem 13. Let G be a nonblocking DFA over Σ and $K \subseteq Q(L_m(G))$ a specification. If $L(G)$ is MOC wrt Q, P, and Σ_c, and K and $L_m(G)$ are synchronously nonconflicting, then

$$\mathrm{supRO}(K \| L_m(G), L(G)) \subseteq \mathrm{supRO}(K, Q(L(G))) \parallel L_m(G)\,.$$

Proof. Let $S = \mathrm{supRO}(K \| L_m(G), L(G))$. Since $S \subseteq K \| L_m(G)$, $Q(S) \subseteq K \cap Q(L_m(G)) = K$. We now show that $Q(S)$ is relatively observable wrt K, $Q(L(G))$, and P_{hi}. To this end, let $t, t' \in \Sigma_{hi}^*$ be such that $P_{hi}(t) = P_{hi}(t')$, and let $e \in \Sigma_{hi}$ be such that $te \in Q(\overline{S})$, $t' \in \overline{K}$, and $t'e \in Q(L(G))$. We have to show that $t'e \in Q(\overline{S})$. To this aim, let $se \in \overline{S}$ be such that $Q(se) = te$. Since $t'e \in Q(L(G))$ and $P_{hi}(Q(se)) = P_{hi}(t'e)$, MOC implies that there is $w' \in L(G)$ such that $Q(w') = t'e$ and $P(se) = P(w')$. Then $w' = s'e$ for some $s' \in L(G)$. Since $Q(w') = t'e$, we have that $Q(s') = t'$ and $P(s) = P(s')$. From $t' \in \overline{K}$ and the synchronous nonconflictingness of K and $L_m(G)$, we conclude that $s' \in \overline{K} \| L(G) = \overline{K \| L_m(G)}$. Altogether, $P(s) = P(s')$, $se \in \overline{S}$, $s' \in \overline{K \| L_m(G)}$, and $s'e \in L(G)$. Then, relative observability of S wrt $K \| L_m(G)$, $L(G)$, and P implies that $s'e \in \overline{S}$. Hence, $t'e = Q(s'e) \in Q(\overline{S})$. $\square$

Notice that the plant in Example 12 does not satisfy MOC, and hence MOC is not a necessary condition in Theorem 13.

A proof of the following result can be found in the appendix.

Theorem 14. Verifying MOC for NFAs is PSPACE-complete.

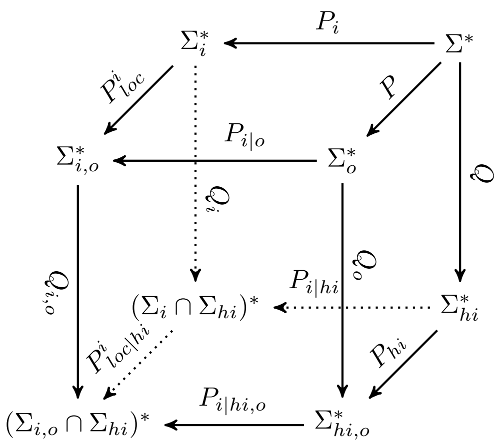

Fig. 4. Our notation for the used projections

Similarly as for OC, the verification of MOC is not easier for DFA models. We provide a proof of PSPACE-hardness for DFAs in the full version.

7. MODULARITY

Let $G = G_1 \| \cdots \| G_n$ be a modular DES. For simplicity, we write L_i to denote $L(G_i)$ and $L = L(G) = L_1 \| \cdots \| L_n$. Similarly for $L_{m,i} = L_m(G_i)$ and $L_m = L_m(G)$.

In addition to the high-level alphabet Σ_{hi} and the set of observable events Σ_o, we have the local alphabets Σ_i, $i = 1, \ldots, n$. The intersection of the alphabets is denoted by adding two corresponding subscripts, e.g., $\Sigma_{i,o} = \Sigma_i \cap \Sigma_o$ denotes the locally observable events of Σ_i, and $\Sigma_{hi,o} = \Sigma_{hi} \cap \Sigma_o$ denotes the high-level observable events. The various projections are denoted as shown in Fig. 4.

We further assume that the high-level alphabet contains all shared events, i.e., $\Sigma_s \subseteq \Sigma_{hi}$, where $\Sigma_s = \cup_{i \neq j}(\Sigma_i \cap \Sigma_j)$ is the set of all events shared by two or more components. In addition, we assume that the modular components agree on the controllability and observability status of the shared events, which is a standard assumption in hierarchical decentralized control.

We now show that if all the local languages satisfy MOC, the their parallel composition also satisfies MOC.

Theorem 15. Assume that each shared event is high level and observable, i.e., $\Sigma_s \subseteq \Sigma_{hi} \cap \Sigma_o$. If, for $i = 1, \ldots, n$, L_i is MOC wrt Q_i, P^i_{loc}, and $P^i_{loc|hi}$, then $\|_{i=1}^n L_i$ is MOC wrt Q, P, and P_{hi}.

8. CONCLUSION

We have completed the missing results in hierarchical supervisory control under partial observation. The regular behavior of the systems is essential for decidability of OC, MOC, and LOC. In the full version, we show that if slightly more expressive one-turn deterministic pushdown systems are used, the properties are undecidable. Deterministic pushdown systes have been discussed in supervisory control in the context of controllability and synthesis as a generalization of system models for which the synthesis is still possible.

REFERENCES

Alves, M.V.S., Carvalho, L.K., and Basilio, J.C. (2017). New algorithms for verification of relative observability and computation of supremal relatively observable sublanguage. *IEEE Trans. Autom. Control*, 62(11), 5902–5908.

Baier, C. and Moor, T. (2015). A hierarchical and modular control architecture for sequential behaviours. *Discrete Event Dyn. Syst.*, 25(1-2), 95–124.

Boutin, O., Komenda, J., Masopust, T., Schmidt, K., and van Schuppen, J.H. (2011). Hierarchical control with partial observations: Sufficient conditions. In *CDC-ECC*, 1817–1822.

Brandt, R.D., Garg, V., Kumar, R., Lin, F., Marcus, S.I., and Wonham, W.M. (1990). Formulas for calculating supremal controllable and normal sublanguages. *Syst. Control Lett.*, 15(2), 111–117.

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

Cassandras, C.G. and Lafortune, S. (2008). *Introduction to discrete event systems*. Springer, 2nd edition.

Clemente, L. and Mayr, R. (2019). Efficient reduction of nondeterministic automata with application to language inclusion testing. *Logical Methods in Computer Science*, 15(1).

da Cunha, A. and Cury, J. (2007). Hierarchical supervisory control based on discrete event systems with flexible marking. *IEEE Trans. Autom. Control*, 52(12), 2242–2253.

Fekri, M.Z. and Hashtrudi-Zad, S. (2009). Hierarchical supervisory control of discrete-event systems under partial observation. In *CDC/CCC*, 181–186.

Hubbard, P. and Caines, P. (2002). Dynamical consistency in hierarchical supervisory control. *IEEE Trans. Autom. Control*, 47(1), 37–52.

Kao, J.Y., Rampersad, N., and Shallit, J. (2009). On NFAs where all states are final, initial, or both. *Theor. Comput. Sci.*, 410(47-49), 5010–5021.

Komenda, J. and Masopust, T. (2010). Supremal normal sublanguages in hierarchical supervisory control. In *WODES*, 121–126.

Lin, F. and Wonham, W.M. (1988). On observability of discrete event systems. *Inf. Sci.*, 44(3), 173–198.

Ngo, Q. and Seow, K. (2014). Command and control of discrete-event systems: Towards online hierarchical control based on feasible system decomposition. *IEEE T. Autom. Sci. Eng.*, 11(4), 1218–1228.

Ngo, Q. and Seow, K. (2018). A hierarchical consistency framework for real-time supervisory control. *Discrete Event Dyn. Syst.*, 28(3), 375–426.

Sakakibara, A. and Ushio, T. (2018). Decentralized supervision and coordination of concurrent discrete event systems under LTL constraints. In *WODES*, volume 51 of *IFAC-PapersOnLine*, 7–12.

Schmidt, K. and Breindl, C. (2011). Maximally permissive hierarchical control of decentralized discrete event systems. *IEEE Trans. Autom. Control*, 56(4), 723–737.

Schmidt, K., Moor, T., and Perk, S. (2008). Nonblocking hierarchical control of decentralized discrete event systems. *IEEE Trans. Autom. Control*, 53(10), 2252–2265.

Torrico, C. and Cury, J. (2002). Hierarchical supervisory control of discrete event systems based on state aggregation. In *IFAC World Congress*, volume 35 of *IFAC Proc. Volumes*, 169–174.

Wong, K. and Wonham, W. (1996a). Hierarchical control of discrete-event systems. *Discrete Event Dyn. Syst.*, 6(3), 241–273.

Wong, K. and Wonham, W. (1996b). Hierarchical control of timed discrete-event systems. *Discrete Event Dyn. Syst.*, 6(3), 275–306.

Wonham, W.M. and Cai, K. (2018). *Supervisory control of discrete-event systems*. Springer.

Zhong, H. and Wonham, W. (1990a). Hierarchical coordination. In *ISIC*, volume 1, 8–14.

Zhong, H. and Wonham, W. (1990b). On the consistency of hierarchical supervision in discrete-event systems. *IEEE Trans. Autom. Control*, 35(10), 1125–1134.

Appendix A. PROOFS

A.1 PSpace-*hardness proof of Theorem 5*

We first show that if L is OC, then the inclusion holds. To this end, assume that $(t,t') \in Q(L) \parallel_{\Sigma_{hi}\cap\Sigma_o} Q(L)$. By the definition of $\parallel_{\Sigma_{hi}\cap\Sigma_o}$, $t,t' \in Q(L)$ and t,t' coincide on the letters of $\Sigma_{hi} \cap \Sigma_o$, i.e., $P_{hi}(t) = P_{hi}(t')$. Since L is OC, there are $s,s' \in L$ such that $Q(s) = t$, $Q(s') = t'$, and $P(s) = P(s')$. However, $P(s) = P(s')$ implies that $(s,s') \in L \parallel_{\Sigma_o} L$, and $Q(s) = t$ and $Q(s') = t'$ imply that $(Q(s),Q(s')) = (t,t')$, which shows the inclusion.

On the other hand, assume that the inclusion holds. We show that L is OC. To this end, assume that $t,t' \in Q(L)$ are such that $P_{hi}(t) = P_{hi}(t')$. By the definition of $\parallel_{\Sigma_{hi}\cap\Sigma_o}$, we obtain that $(t,t') \in Q(L) \parallel_{\Sigma_{hi}\cap\Sigma_o} Q(L)$. Since the inclusion holds, we have $(t,t') \in Q(L \parallel_{\Sigma_o} L)$, which means that there is a pair $(s,s') \in L \parallel_{\Sigma_o} L$ such that $(Q(s),Q(s')) = (t,t')$. Since $(s,s') \in L \parallel_{\Sigma_o} L$, strings s and s' belong to L and coincide on the letters from Σ_o, i.e., $P(s) = P(s')$, which was to be shown.

To show PSpace-hardness, we reduce the problem of deciding universality for NFAs with all states marked, see Kao et al. (2009). Such NFAs recognize exactly prefix-closed languages. The problem asks, given an NFA A over Σ with all states marked, whether the language $L(A) = \Sigma^*$. To A, we construct an NFA B such that $L(B) = @\#L(A) \cup @\Sigma^* \cup \#\Sigma^*$. It is not difficult to construct B from A in polynomial time by adding a new initial state that goes to the initial state of A under the sequence @# and that has a self-loop under every event from Σ after @, and by adding a new state reachable under # having a self-loop under Σ. Let the abstraction Q remove $\{@\}$, and the observation P remove $\{\#\}$, that is, $\Sigma_{hi} = \Sigma \cup \{\#\}$ and $\Sigma_o = \Sigma \cup \{@\}$. Then $Q(L(B)) = \Sigma^* \cup \#\Sigma^*$. We now show that $L(B)$ is OC if and only if A is universal.

If A is universal, then any two different strings $t,t' \in Q(L(B))$ with $P_{hi}(t) = P_{hi}(t')$ are such that $t' = \#t$ (or vice versa). Then, $s = @t$ and $s' = @\#t$ belong to $L(B)$, $Q(s) = t$, $Q(s') = \#t$, and $P(s) = @t = P(s')$. Hence $L(B)$ is OC.

If A is not universal, there is $w \notin L(A)$. Consider the strings $@w$, $\#w \in L(B)$. Then $w, \#w \in Q(L(B))$ and $P_{hi}(w) = P_{hi}(\#w) = w$. We now show that there are no strings $s, s' \in L(B)$ such that $Q(s) = w$, $Q(s') = \#w$, and $P(s) = P(s')$, i.e., that $L(B)$ is not OC. To do this, we observe that $Q^{-1}(w) \cap L(B) = \{@w\}$ and $Q^{-1}(\#w) \cap L(B) = \{\#w\}$; $@\#w$ does not belong to $L(B)$ because $w \notin L(A)$. But then $P(@w) = @w \neq w = P(\#w)$, which completes the proof.

A.2 PSpace-*hardness proof of Theorem 6*

To show membership in PSpace, we use a similar technique as in the previous theorem. Namely, we construct an automaton recognizing the sublanguage of $L \times Q(L) \times L \times Q(L)$, where every $(w_1, w_2, w_3, w_4) \in L \times Q(L) \times L \times Q(L)$ satisfies

$$P(w_1) = P(w_3), Q(w_1) = w_2, \text{ and } Q(w_3) = w_4 .$$

We denote the language by $[L, Q(L), L, Q(L)]$. If L is recognized by an NFA $G = (Q, \Sigma, \gamma, I, F)$, then $[L, Q(L), L, Q(L)]$ is recognized by the automaton

$$H = (Q^4, [(\Sigma \cup \{\varepsilon\}) \times (\Sigma_{hi} \cup \{\varepsilon\})]^2, \delta, I^4, F^4)$$

where the alphabet consists of quadruples and the transition function $\delta\colon Q^4 \times ((\Sigma \cup \{\varepsilon\}) \times (\Sigma_{hi} \cup \{\varepsilon\}))^2 \to Q^4$ is defined on these quadruples as follows:

- for $a \in \Sigma_o \cap \Sigma_{hi}$, $\delta((p,q,r,s),(a,a,a,a)) = \gamma(p,a) \times \gamma(q,a) \times \gamma(r,q) \times \gamma(s,a)$;
- for $a \in \Sigma_o \setminus \Sigma_{hi}$, $\delta((p,q,r,s),(a,\varepsilon,a,\varepsilon)) = \gamma(p,a) \times (\gamma(q,a) \cup \{q\}) \times \gamma(r,a) \times (\gamma(s,a) \cup \{s\})$;
- for $a \in \Sigma_{hi} \setminus \Sigma_o$,
  $$\delta((p,q,r,s),(a,a,\varepsilon,\varepsilon)) = \gamma(p,a) \times \gamma(q,a) \times \{r\} \times \{s\},$$
  $$\delta((p,q,r,s),(\varepsilon,\varepsilon,a,a)) = \{p\} \times \{q\} \times \gamma(r,a) \times \gamma(s,a);$$
- for $a \notin \Sigma_{hi} \cup \Sigma_o$,
  $$\delta((p,q,r,s),(a,\varepsilon,\varepsilon,\varepsilon)) = \gamma(p,a) \times (\gamma(q,a) \cup \{q\}) \times \{r\} \times \{s\},$$
  $$\delta((p,q,r,s),(\varepsilon,\varepsilon,a,\varepsilon)) = \{p\} \times \{q\} \times \gamma(r,a) \times (\gamma(s,a) \cup \{s\}).$$

Thus, any element of the language $[L, Q(L), L, Q(L)]$ is of the form $(s, Q(s), s', Q(s'))$ with $P(s) = P(s')$. On the other hand, for any $s, s' \in L$ with $P(s) = P(s')$, it can be shown that $(s, Q(s), s', Q(s')) \in [L, Q(L), L, Q(L)]$.

Let $e \in \Sigma_c \cap \Sigma_{hi}$. The LOC condition states that for any $s, s' \in L$ with $P(s) = P(s')$ and $Q(s)e, Q(s')e \in Q(L)$ something holds. Therefore, we need to restrict the language $[L, Q(L), L, Q(L)]$ only to the elements for which $Q(s)e, Q(s')e \in Q(L)$. To do this, we concatenate the event $(\varepsilon, e, \varepsilon, e)$ to $[L, Q(L), L, Q(L)]$ and intersect the result with the language $\Sigma^* \times Q(L) \times \Sigma^* \times Q(L)$. This checks that, for any $s, s' \in L$ with $P(s) = P(s')$, we also have $Q(s)e, Q(s')e \in Q(L)$.

For every such $(s, (Q(s), s', Q(s'))$, the LOC condition requires that there are $u, u' \in (\Sigma \setminus \Sigma_{hi})^*$ such that $P(u) = P(u')$ and $sue, s'u'e \in L$. To verify whether this is satisfied, we check whether the language

$$[L, Q(L), L, Q(L)] \cdot (\varepsilon, e, \varepsilon, e) \cap (\Sigma^* \times Q(L) \times \Sigma^* \times Q(L))$$

is a subset of the sublanguage $L \times \Sigma^* \times L \times \Sigma^*$ where, for every (x, y, z, w), there is an extension from the language

$$[(\Sigma \setminus \Sigma_{hi})^*, \varepsilon, (\Sigma \setminus \Sigma_{hi})^*, \varepsilon] \cdot (e, \varepsilon, e, \varepsilon);$$

here, $[(\Sigma \setminus \Sigma_{hi})^*, \varepsilon, (\Sigma \setminus \Sigma_{hi})^*, \varepsilon]$ is the sublanguage of $(\Sigma \setminus \Sigma_{hi})^* \times \{\varepsilon\} \times (\Sigma \setminus \Sigma_{hi})^* \times \{\varepsilon\}$ with the property that, for any $(u, \varepsilon, u', \varepsilon) \in [(\Sigma \setminus \Sigma_{hi})^*, \varepsilon, (\Sigma \setminus \Sigma_{hi})^*, \varepsilon]$, $P(u) = P(u')$. An automaton for this language is constructed in a similar way as the automaton H above.

Checking the existence of such an extension corresponds to the operation of *right quotient* denoted by $/$, i.e., we use the language $L \times \Sigma^* \times L \times \Sigma^* / [(\Sigma \setminus \Sigma_{hi})^*, \varepsilon, (\Sigma \setminus \Sigma_{hi})^*, \varepsilon] \cdot (e, \varepsilon, e, \varepsilon)$.[2]

Altogether, for every event $e \in \Sigma_c \cap \Sigma_{hi}$, we check the inclusion

$$[L, Q(L), L, Q(L)] \cdot (\varepsilon, e, \varepsilon, e) \cap (\Sigma^* \times Q(L) \times \Sigma^* \times Q(L))$$
$$\subseteq (L \times \Sigma^* \times L \times \Sigma^*) / [(\Sigma \setminus \Sigma_{hi})^*, \varepsilon, (\Sigma \setminus \Sigma_{hi})^*, \varepsilon] \cdot (e, \varepsilon, e, \varepsilon)$$

which requires only polynomial space. We leave the proof details for the full version of the paper.

[2] An automaton for this language can be constructed, e.g., as suggested by Jan Hendrik at https://cs.stackexchange.com/questions/102037/constructive-proof-to-show-the-quotient-of-two-regular-languages-is-regular

To show PSPACE-hardness, we reduce the PSPACE-complete universality problem for NFAs with all states marked Kao et al. (2009). Let A be an NFA over $\Sigma_A = \{a_1, a_2, \dots, a_n\}$, $n \geq 2$, such that $L(A) \neq \emptyset$. Then $L(A)$ is prefix-closed, and hence $\varepsilon \in L(A)$. The universality problem asks whether the language $L(A) = \Sigma_A^*$. Let $\Sigma_c = \Sigma_o = \Sigma_{hi} = \Sigma_A$, and let $\Sigma'_A = \{a' \mid a \in \Sigma_A\}$ be a disjoint copy of Σ_A. From A, we construct an NFA B over $\Sigma = \Sigma_A \cup \Sigma'_A$ with all states marked such that $L(B) = \overline{\Sigma_A \cdot \Sigma_A \cdot L(A)} \cup \overline{(\Sigma_A \cdot \Sigma'_A)^*}$; see Fig. A.1 for an illustration. Then, $Q(L(B)) = \Sigma_A^*$. We now show that A is universal if and only if $L(B)$ is LOC.

Assume that A is not universal, and consider a shortest string $w \in \Sigma_A^* \setminus L(A)$. Then $w = te$ for some $t \in L(A)$ and $e \in \Sigma_A$. We show that $L(B)$ is not LOC. Set $s = a_1 a_1 t \in L(B)$ and notice that $e \in \Sigma_A = \Sigma_{hi} \cap \Sigma_c$. Let $t = b_1 \cdots b_m$ and $s' = a_1 a'_1 a_1 a'_1 b_1 b'_1 \cdots b_m b'_m e \in L(B)$; indeed, s' can be generated from state n_3. Then, $Q(s')e = Q(a_1 a'_1 a_1 a'_1 b_1 b'_1 \cdots b_m b'_m)e = Q(s)e \in Q(L(B))$ and $P(s) = s = P(s')$. Since s is generated by B only from state n_1, because of the initial prefix $a_1 a_1$, and there is no transition labeled by an event from $\Sigma \setminus \Sigma_{hi} = \Sigma'_A$ reachable from n_1, there is no $u \in (\Sigma \setminus \Sigma_{hi})^*$ such that $a_1 a_1 tue = sue \in L(B)$; notice also that $se = a_1 a_1 te = a_1 a_1 w \notin L(B)$ because $w \notin L(A)$. Hence, $L(B)$ is not LOC.

On the other hand, assume that A is universal. Let $s, s' \in L(B)$ be such that $P(s) = P(s')$, and let $e \in \Sigma_c \cap \Sigma_{hi} = \Sigma_A$. Clearly, $Q(s)e, Q(s')e \in Q(L(B))$. Let $t \in \{s, s'\}$. If t is generated from state n_3, it can indeed be extended by a string $v \in \Sigma'_A \cup \{\varepsilon\}$ to generate event e; in that case, we have that $P(v) = \varepsilon$. If t is generated from state n_1, it can clearly generate event e from states n_1 and n_2; thus, if $t = a_i a_j t'$ for some $a_i, a_j \in \Sigma_A$ and $t' \in \Sigma_A^*$, the universality of A implies that $t'e \in L(A)$. Altogether, we have shown that $sue, s'u'e \in L(B)$ for some $u, u' \in \Sigma_A'^*$ with $P(u) = P(u') = \varepsilon$.

A.3 Proof of Theorem 14

Since MOC is a modification of OC, the proof is a modification of that of Theorem 5. Let $L \subseteq \Sigma^*$ be a prefix-closed language, and let Σ_o and Σ_{hi} be the respective observation and high-level alphabets. We show that L is MOC wrt Q, P, and P_{hi} iff

$$L \parallel_{\Sigma_{hi} \cap \Sigma_o} Q(L) \subseteq Q_2\left(L \parallel_{\Sigma_o} L\right) ,$$

where, for an event (a, b), $Q_2(a, b) = (a, Q(b))$. Membership in PSPACE then follows, since we can express $Q(L)$, as well as $Q_2(L \parallel_{\Sigma_o} L)$, as NFAs, and the inclusion of two NFAs can be verified in PSPACE.

We first show that if L is MOC, then the inclusion holds. To this end, assume that $(s, t') \in L \parallel_{\Sigma_{hi} \cap \Sigma_o} Q(L)$. By the definition of $\parallel_{\Sigma_{hi} \cap \Sigma_o}$, $s \in L$, $t' \in Q(L)$, and $P_{hi}(Q(s)) = P_{hi}(t')$. Since L is MOC, there is $s' \in L$ such that $Q(s') = t'$ and $P(s) = P(s')$. However, $P(s) = P(s')$ implies $(s, s') \in L \parallel_{\Sigma_o} L$, and $Q(s') = t'$ implies that $(s, Q(s')) = (s, t')$, which shows the inclusion.

We now show that the inclusion implies that L is MOC. To this end, assume that $s \in L$, $t' \in Q(L)$, and $P_{hi}(Q(s)) = P_{hi}(t')$. By the definition of $\parallel_{\Sigma_{hi} \cap \Sigma_o}$, we obtain that $(s, t') \in L \parallel_{\Sigma_{hi} \cap \Sigma_o} Q(L)$. Since the inclusion holds, $(s, t') \in Q_2(L \parallel_{\Sigma_o} L)$, which means that there is a pair $(s, s') \in L \parallel_{\Sigma_o} L$ such that $(s, Q(s')) = (s, t')$ and that the strings s and s' belong to L and coincide on Σ_o, i.e., $P(s) = P(s')$, which was to be shown.

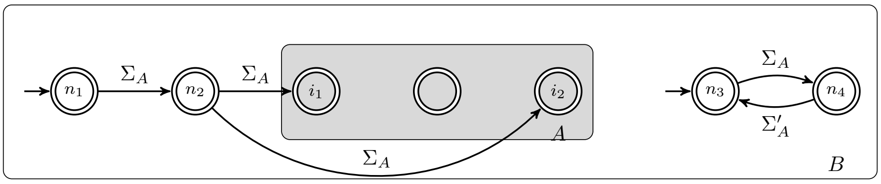

Fig. A.1. Construction of the NFA B from the NFA A; four new states n_1, n_2, n_3, n_4; states n_1 and n_3 are the only initial states of B; a transition from n_1 to n_2, and from n_2 to every initial state of A (denoted by i_1 and i_2) under every event from Σ_A

We show PSPACE-hardness by reduction from the problem of deciding universality for NFAs with all states marked. Let A be an NFA over Σ with all states marked. We construct a DFA B such that $L(B) = @\#L(A) \cup @\Sigma^* \cup \#\Sigma^* \cup L(A)$. It is not difficult to construct B from A in polynomial time. Let $\Sigma_{hi} = \Sigma \cup \{\#\}$ and $\Sigma_o = \Sigma \cup \{@\}$. Then $Q(L(B)) = \Sigma^* \cup \#\Sigma^*$. We now show that $L(B)$ is MOC if and only if A is universal.

Assume that $L(A) = \Sigma^*$. Let $s \in L(B)$ and $Q(s) \neq t' \in Q(L(B))$ with $P_{hi}(Q(s)) = P_{hi}(t')$. We have the following cases:

(1) $Q(s) \in \Sigma^*$ and $t' = \#Q(s)$ for $s \in @\Sigma^* \cup L(A)$:
  (a) If $s = @w \in @\Sigma^*$, let $s' = @\#w$.
  (b) If $s \in L(A)$, let $s' = \#s$.
(2) $t' \in \Sigma^*$ and $Q(s) = \#t'$ for $s \in @\#L(A) \cup \#\Sigma^*$:
  (a) If $s = @\#t' \in @\#L(A)$, let $s' = @t'$.
  (b) If $s = \#t' \in \#\Sigma^*$, let $s' = t'$.

In all cases, it can be verified that $s' \in L(B)$, $Q(s') = t'$, and $P(s) = P(s')$, and hence $L(B)$ is MOC.

If A is not universal, there is $w \notin L(A)$. We consider the strings $s = @w \in L(B)$ and $\#w \in Q(L(B))$, for which $P_{hi}(Q(@w)) = P_{hi}(\#w) = w$, and show that there is no $s' \in L(B)$ such that $Q(s') = \#w$ and $P(s) = P(s')$, i.e., that $L(B)$ is not MOC. To do this, notice that $Q^{-1}(\#w) \cap L(B) = \{\#w\}$, and hence $P(s) = P(@w) = @w \neq w = P(\#w)$, which completes the proof.

A.4 Proof of Theorem 15

The proof makes use of the following well-known result.

Lemma 16. (Wonham and Cai (2018)). Let $\Sigma_s \subseteq \Sigma_{hi}$, and let $L_i \subseteq \Sigma_i^*$ be languages, then $Q(\|_{i=1}^n L_i) = \|_{i=1}^n Q_i(L_i)$.

Let $L = \|_{i=1}^n L_i$, and assume that $s \in L$ and $t' \in Q(L)$ are such that $P_{hi}(Q(s)) = P_{hi}(t')$. We show that there is $s''' \in L$ such that $Q(s''') = t'$ and $P(s) = P(s''')$. Since $\Sigma_s \subseteq \Sigma_{hi}$, Lemma 16 implies $Q(\|_{i=1}^n L_i) = \|_{i=1}^n Q_i(L_i)$. Projecting to local alphabets gives that $P_{i|hi}(Q(s)) \in Q_i(L_i)$ and $P_{i|hi}(t') \in Q_i(L_i)$, $i = 1, \ldots, n$. Moreover, $P_{hi}(Q(s)) = P_{hi}(t')$ implies that $P_{i|hi,o}(P_{hi}(Q(s))) = P_{i|hi,o}(P_{hi}(t'))$. The commutative diagram of Fig. 4 gives that $P^i_{loc|hi} P_{i|hi}(Q(s)) = P^i_{loc|hi} P_{i|hi}(t')$, and that $P_{i|hi}(Q(s)) = Q_i P_i(s)$. Let $s_i = P_i(s)$ and $t'_i = P_{i|hi}(t')$. Then MOC of L_i wrt Q_i, P^i_{loc}, $P^i_{loc|hi}$ implies that there is $s'_i \in L_i$ such that $Q_i(s'_i) = t'_i$ and $P^i_{loc}(s_i) = P^i_{loc}(s'_i)$, $i = 1, \ldots, n$. We first show that $\|_{i=1}^n s'_i$ is nonempty. It suffices to prove that $Q(\|_{i=1}^n s'_i)$ is nonempty. Since $\Sigma_s \subseteq \Sigma_{hi}$, Lemma 16 gives that $Q(\|_{i=1}^n s'_i) = \|_{i=1}^n Q_i(s'_i) = \|_{i=1}^n P_{i|hi}(t')$, which is nonempty, because $t' \in \|_{i=1}^n P_{i|hi}(t')$. Hence, there is $s' \in \|_{i=1}^n s'_i$ such that $Q(s') = t'$. Furthermore, $P^i_{loc}(s_i) = P^i_{loc}(s'_i)$, for $i = 1, \ldots, n$, means that $P(s) \in P(\|_{i=1}^n s_i) = \|_{i=1}^n P^i_{loc}(s_i) = \|_{i=1}^n P^i_{loc}(s'_i) = P(\|_{i=1}^n s'_i) \ni P(s')$ by $\Sigma_s \subseteq \Sigma_o$ and Lemma 16. Hence, there is $s'' \in \|_{i=1}^n s'_i$ such that $P(s) = P(s'')$. Since $P_{hi}(t') = P_{hi}Q(s)$, there is $s''' \in Q(s')\|P(s'')$. But then $Q(s''') = Q(s') = t'$ and $P(s''') = P(s'') = P(s)$, which was to be shown.