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
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 \(\text{PSPACE}\)-complete for NFAs, is unproved. In particular, the membership in \(\text{PSPACE}\) is not shown. The problem is \(\text{PSPACE}\)-hard, but it is open whether the verification of (M)OC is decidable; in particular, it is open whether the problem is in \(\text{PSPACE}\).

(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 \(Q, P, P_{hi}\), then for every high-level specification \(K \subseteq Q(L_m)\).

\[
\sup N(K\|L_m, L, P) = \sup N(K, Q(L), P_{hi}) \| L_m
\]

whenever \(\sup N(K, Q(L), P_{hi})\) 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., \(\Sigma_o \subseteq \Sigma_{hi} \cap \Sigma_h\). If, for \(i = 1, \ldots, n\), \(L_i\) is MOC wrt \(Q_i, P_{loc}^i, P_{loc|hi}^i\); then \(\prod_{i=1}^n L_i\) is MOC wrt \(Q, P, P_{hi}\).

A counterexample: Let \(L_1 = \{h_1 o_1 x\}\) and \(L_2 = \{o_2 h_2 x\}\) 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 \(s = h_1 o_1 o_2 x\) 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 \(h_1 o_1 o_2 x\), 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 \( \omega \)-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 supervisory 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 \( \Sigma \) 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 with respect to the specification \( K \mid L_m(G) \).

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 with 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 $\Sigma, \Sigma'$ denotes the set of all finite strings over $\Sigma$; the empty string is denoted by $\epsilon$. The alphabet $\Sigma$ is partitioned into controllable events $\Sigma_c$ and uncontrollable events $\Sigma_u = \Sigma \setminus \Sigma_c$ as well as into observable events $\Sigma_o$ and unobservable events $\Sigma_{uo} = \Sigma_o \setminus \Sigma_o$. A language is a subset of $\Sigma^*$. For a language $L \subseteq \Sigma^*$, the prefix closure $\overline{L} = \{ w \in \Sigma^* \mid \forall w \in L \}$; $L$ is prefix-closed if $L = \overline{L}$.

A (natural) projection $R : \Sigma^* \to \Gamma^*$, where $\Gamma \subseteq \Sigma$ are alphabets, is a homomorphism for concatenation defined so that $R(a) = a$ for $a \in \Sigma \setminus \Gamma$, and $R(a) = a = a$ for $a \in \Gamma$. The action of $R$ on $w \in \Sigma^*$ is to remove all events from $w$ that are not in $\Gamma$. 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, $\Sigma$ is an input alphabet, $I \subseteq Q$ is a set of initial states, $F \subseteq Q$ is a set of marked states, and $\delta : Q \times \Sigma \to 2^Q$ is the transition function that can be extended to the domain $2^Q \times 2^\Sigma$ 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 letter $a \in \Sigma$. 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 $L_m(G) = L(G)$, 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 $L_1 \parallel L_2 = P^{-1}(L_1) \cap P^{-1}(L_2)$, where $P_1 : (\Sigma_1 \cup \Sigma_2)^* \to \Sigma_1^*$ is a projection, for $i = 1, 2$; see Cassandras and Laafortune (2008) for a definition for automata. For two DFAs $G_1$ and $G_2$, $L(G_1) \parallel L(G_2) = L(G_1)\parallel L(G_2)$, Languages $L_1$ and $L_2$ are synchronously nonconflicting if $[L_1][L_2] = [L_1][L_2]$.

Let $G$ be a DFA over an alphabet $\Sigma$. A language $K \subseteq L_m(G)$ is controllable wrt $L(G)$ and the set of uncontrollable events $\Sigma_u$ if $K \cap \Sigma_u \cap L(G) \subseteq \overline{K}$; $K$ is observable wrt $L(G)$, the set of observable events $\Sigma_o$ with $\Sigma = \Sigma_c \cup \Sigma_o$ being the corresponding projection, and the set of controllable events $\Sigma_c$ 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 $s' \in \overline{K}$, Algorithms to verify controllability and observability can be found in Cassandras and Laafortune (2008).

It is known that there is no suprenal 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 $P : \Sigma^* \to \Sigma_o$ 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 $P : \Sigma^* \to \Sigma_o$ (or simply C-observable) if for all strings $s, s' \in \Sigma^*$ with $P(s) = P(s')$ and for every $e \in \Sigma_c$, 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 $\Sigma$ be the nonempty alphabet, $\Sigma_{hi} \subseteq \Sigma$ the high-level alphabet, and $\Sigma_o \subseteq \Sigma$ the set of observable events. Let $P : \Sigma^* \to \Sigma_o$ be the projection corresponding to system’s partial observation, $Q : \Sigma^* \to \Sigma_{hi}$ the projection corresponding to the high-level abstraction, and $P_{hi} : \Sigma_{hi} \to (\Sigma_{hi} \cap \Sigma_o)^*$ and $Q_{hi} : \Sigma_o \to (\Sigma_{hi} \cap \Sigma_o)^*$ the corresponding observations and abstractions.

![Diagram](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 $\Sigma$, 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 $\Sigma_{hi}$ 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 $L_m(S/G) = K\parallel L_m(G)$.

Boutin et al. (2011) identified sufficient conditions (observation consistency and local observation consistency) on the low-level plant $G$ for which observability of $K\parallel L_m(G)$ 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 $s, t \in Q(L)$ 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 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 $\Sigma_c$ if for all strings $s, s' \in L$ and all events $e \in \Sigma_c \cap \Sigma_{hi}$ such that $P(s)e = P(s'\epsilon) \in Q(L)$ and $P(s) = P(s')$, there exist low-level strings $u, u' \in (\Sigma \setminus \Sigma_{hi})^*$ such that $P(u) = P(u')$ and such $s' \epsilon e \in L$. Intuitively, continuing two observationally equivalent high-level strings by the same controllable event, the corresponding low-level observationally equivalent strings can be continued by the 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 $Q : \Sigma \to L_m(\Sigma)$ is an $L_m(\Gamma)$-observer for a nonblocking plant $G$ over $\Sigma$ if for all strings $t \in Q(L_m(G))$ and $s \in L(\Gamma)$, 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 $s \in L(\Gamma)$ if for all $e \in \Sigma_{hi} \cap \Sigma_{o}$ such that $Q(s)e \in L(\Gamma_{hi})$, 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 $M \subseteq L(\Gamma)$ 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 $\Sigma$, and let $K \subseteq Q(L_m(G))$ be a (high-level) specification. Let $Q$ be LCC for $L(\Gamma)$ and $\Sigma_u$, and an $L_m(\Gamma)$-observer. Let $L(G)$ be OC wrt $\Sigma$, and $P_{hi}$, and LOC wrt $\Sigma$, and $\Sigma_{hi}$. Then $K$ is LCC wrt $Q(L(\Gamma))$ and $\Sigma_{hi} \cap \Sigma_{oi}$, and observable wrt $Q(L(G))$, $\Sigma_{hi} \cap \Sigma_{oi}$, and $\Sigma_{hi} \cap \Sigma_{oi}$ if and only if $K \cap L_m(G)$ is observable wrt $L(G)$, $\Sigma_{oi}$, and $\Sigma_{oi}$.

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 $L_{hi}(S_{hi}/G_{hi}) = K$, then there is a nonblocking supervisor $S$ such that $L_{hi}(S/G) = K \cap L_m(G)$. In particular, a DFA realization of $G_K$ of 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 $\Sigma$, and let $K \subseteq Q(L_m(G))$ be a specification. Assume that $L(G)$ is OC wrt $\Sigma$, and $P_{hi}$, that $K$ and $L_m(G)$ are synchronously nonconflicting, and that $L(G)$ is LOC wrt $\Sigma$, and $\Sigma_{hi}$. Then $K$ is observable wrt $Q(L(G))$, $\Sigma_{hi} \cap \Sigma_{oi}$, and $\Sigma_{hi} \cap \Sigma_{oi}$ if and only if $K \cap L_m(G)$ is observable wrt $L(G)$, $\Sigma_{oi}$, and $\Sigma_{oi}$.

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 $\Sigma$, and $P_{hi}$, and $\Sigma_{hi}$ and $L_m(G)$ are synchronously nonconflicting, then $K$ is normal wrt $Q(L(G))$ and $P_{hi}$ if and only if $K \cap L_m(G)$ is normal wrt $L(G)$ and $P_{hi}$.

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 $K \cap L_m(G)$ is not necessarily $C \cap L(G)$-observable. Let $K = \{e, a\}$, $C = \{e, a, au\}$ over $\Sigma_{hi} = \{a, u\}$, and $L(G) = \{e, a, ae, au, aue\}$ over $\Sigma = \{a, u, e\}$ be prefix-closed languages, and hence synchronously nonconflicting. Let $\Sigma_{oi} = \{a, e\}$. 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, $K \cap L_m(G)$ is not $C \cap L(G)$-observable, since $ae \in K \cap L(G)$, $au \in C \cap L(G)$, and $au \in L(G)$, but $aue \notin K \cap L(G)$ (but $K \cap L(G)$ 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 $\Sigma$ 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 $\Sigma'$ is denoted by $L_1 \parallel_{\Sigma'} L_2$ and defined as the language of the NFA $G_1 \parallel_{\Sigma'} G_2 = (Q_1 \times Q_2, (\Sigma \cup \{e\}) \times (\Sigma \cup \{e\}), \delta, I_1 \times I_2, F_1 \times F_2)$, where the alphabet is a set of pairs based on the synchronization of events in $\Sigma'$.

As in the case of OC, the problem is not easier for DFA models. Therefore, the problem is PSPACE-hard even for DFA models.
6. PRESERVATION OF SUPREMATY

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

Compared to Corollary 4 saying that under the OC condition the specification $K$ is normal if and only if $K \mid L_m(G)$ is normal, the following example shows that OC is not sufficient to preserve normality (relative observability) if the supremal normal (relatively observable) sublanguage of the specification $K$ is a strict sublanguage of $K \mid L_m(G)$ is of the form $X \mid L_m(G)$ 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 $supN(K, L)$ (resp. $supRO(K, L)$) to denote the supremal normal (resp. supremal relatively observable) sublanguage of $K$ wrt $L$ and the corresponding set of observable events.

**Example 8.** Let $\Sigma = \{a, b, c\}$ with $\Sigma_h = \{a, c\}$ and $\Sigma_{hi} = \{b, c\}$, and let $L = \{\varepsilon, a, b, c, ba, bac, bac\}$ and $K = \{\varepsilon, bc\} \subseteq L = \{\varepsilon, a, b, c, ba, bac, bac\} = Q(L)$. To show that $L$ is OC, notice that $P_{hi}(b) = P_{hi}(b)$ 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 $supN(B, M) = P - P^1P(M-B)B^*$, for prefix-closed languages $B \subseteq M \subseteq \Sigma^*$, and we obtain the following: $K \mid L = a'ba' \cup a'ca' \cup a' \cap L = \{\varepsilon, a, b, c, ba, bac\} = L \setminus K \mid L = \{ bac\}$, and $P - P^1P(bac) = P^1(ac) = b'a'bc'$. This gives that $c \in supN(K \mid L, L) = K \setminus L - P^1P(L \setminus K \mid L) \Sigma = \{a, b, c, ba\}$. On the other hand, if $Q(L) = K = \{c\} \subseteq L$. It follows that $K \mid L = L \setminus K \mid L = \{a'b, b'a, c, ba\} = \{bac\}$ such that $c \notin supN(K \mid L, L) = Q^1(K - P_{hi} \mid L = \{a'b, b'a, c, ba\}) \subseteq L = Q^1(K - P_{hi} \mid L = \{a'b, b'a, c, ba\} = \{bac\}$ 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 preserve 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 condition (MOC) if, $P_{hi}$ 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 $Q(s') = t'$.

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

**Proof.** (2): Since $supN(K, Q(L(G)))$ is normal wrt $Q(L(G))$ and $P_{hi}$, Corollary 4 implies that $supN(K, Q(L(G))) \mid L_m(G)$ is normal wrt $L(G)$ and $P$. The implication that $K$ implies normality of $K \mid L_m(G)$ in Corollary 4 holds without any assumptions. Therefore, $supN(K, Q(L(G))) \mid L_m(G) \subseteq supN(K \mid L_m(G), L(G))$.

(⇒): Let $S \subseteq K \mid L_m(G)$ be normal wrt $L(G)$ and $P$, that is, $\overline{S} = P^{-1}P(\overline{S}) \cap L(G)$. Then, $Q(S) \subseteq K \cap Q(L(G)) = K$. We show that $Q(S)$ is normal wrt $Q(L(G))$ and $P_{hi}$, i.e., that $Q(S) = P_{hi}^{-1}P_{hi}(Q(S)) \cap Q(L(G))$. 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, $t' \in P_{hi}^{-1}P_{hi}(Q(S)) \cap Q(L(G))$. We show that $t' \in Q(S)$. 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 $t' = Q(s') \in Q(S)$, which shows normality of $Q(S)$.

Two special cases are often considered in the literature: (i) $S \subseteq \Sigma_h$, and (ii) $\Sigma_{hi} \subseteq \Sigma_h$. 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 $\Sigma_h \subseteq \Sigma_h$, then $supN(K, Q(L)) \mid L = supN(K \mid L, L)$.

First, assume that $S \subseteq \Sigma_h$. Then $P = P_{hi}Q$, since $S_0$ 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 $\Sigma_{hi} \subseteq \Sigma_h$. 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.
We now show that an analogy of Theorem 11 does not hold for relative observability. In particular, the inclusion
\[ \supRO(K \| L_m(G), L(G)) \supseteq \supRO(K, Q(L(G))) \| L_m(G), L(G) \]
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 \( \Sigma_0 = \{a, b, c\} \) and \( \Sigma_q = \{\epsilon\} \). Then \( \supRO(K, Q(L(G))) \) is shown in Fig. 3 as well as \( \supRO(K, Q(L(G))) \| L_m(G), L(G) \). There, the reader can also see the supremal relatively observable sublanguage of \( K \| L_m(G), L(G), P \), which obviously does not include \( \supRO(K, Q(L(G))) \| L_m(G) \).

By Theorem 3, \( \supRO(K, Q(L(G))) \| L_m(G) \) 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 \( \Sigma \) and \( K \subseteq Q(L_m(G)) \) a specification. If \( L(G) \) is MOC wrt \( Q, P, \) and \( \Sigma_q \), and \( K \) and \( L_m(G) \) are synchronously nonconflicting, then \( \supRO(K \| L_m(G), L(G)) \subseteq \supRO(K, Q(L(G))) \| L_m(G) \).

Proof. Let \( S = \supRO(K \| L_m(G), L(G)) \). Since \( S \subseteq K \| L_m(G), Q(S) = K \cap Q(L_m(G)) = K \). 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 \( e \in Q(S) \). Hence, \( e \in Q(S) \). We have to show that \( t'e \in Q(\bar{S}) \). This is true, let \( s \in \bar{S} \) 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 \( Q(s') = t'e \) and \( P(s) = P(s') \). From \( t'e \in \bar{S} \) and the synchronous nonconflictingness of \( K \) and \( L_m(G) \), we conclude that \( s' \in K \| L_m(G) \subseteq K \| L_m(G) \). Altogether, \( P(s) = P(s'), s \in \bar{S}, s' \in K \| L_m(G), \) and \( s'e \in L(G) \). Then, relative observability of \( S \) wrt \( K \| L_m(G), L(G) \), and \( P \) implies that \( s'e \in \bar{S} \). Hence, \( t'e = Q(s'e) \in Q(\bar{S}) \).

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.
computation of suprema of relatively observable sublanguage. *IEEE Trans. Automat. Cont.*, 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 suprema controllable and normal sublanguages. *Systems & Control Letters*, 15(2), 111–117.

Cai, K., Zhang, R., and Wonham, W.M. (2015). Relative controllability and normal sublanguages. *IEEE Trans. Automat. Cont.*, 52(12), 2242–2253.

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

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

Kao, J.Y., Rampersad, N., and Shallit, J. (2009). On NFAs and the letters from $\Sigma_1$. *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 supervisory 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. Automat. Cont.*, 56(4), 723–737.

Schmidt, K., Moor, T., and Perk, S. (2008). Nonblocking hierarchical control of decentralized discrete event systems. *IEEE Trans. Automat. Cont.*, 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 timed discrete-event systems. *Discrete Event Dyn. Syst.*, 6(3), 275–306.

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. Automat. Cont.*, 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 $(t, t') \in Q(L) \subseteq \Sigma_1 \subseteq Q(L)$. By the definition of $\Sigma_1$, $t, t' \in Q(L)$ and $t, t'$ coincide on the letters of $\Sigma_1$. Hence, $P_{t}(t) = P_{t'}(t')$. Since $L$ is OC, we have $s, s' \in L$ such that $P(s) = t, P(s') = t'$, and $P(s') = P(s)$. However, $P(s) = P(s')$ implies that $(s, s') \in L \subseteq \Sigma_1$ 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 $P_{t}(t) = P_{t'}(t')$. By the definition of $\Sigma_1$, we obtain that $(t, t') \in Q(L) \subseteq \Sigma_1 \subseteq Q(L)$. Since the inclusion holds, we have $(t, t') \in Q(L) \subseteq \Sigma_1$, which means that there is a pair $(s, s') \in L \subseteq \Sigma_1$ such that $(Q(s), Q(s')) = (t, t')$. Since $(s, s') \in L \subseteq \Sigma_1$, strings $s$ and $s'$ belong to $L$ and coincide on the letters of $\Sigma_1$, 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 $\Sigma$ with all states marked, whether the language $L(A) = \Sigma^*$. To $A$, we construct an NFA $B$ such that $L(B) = \Sigma^* \cup \Sigma^* \cup \Sigma^*$. 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 $\Sigma$ after $\#$, and by adding a new state reachable under $\#$ having a self-loop under $\Sigma$. Let the abstraction $Q$ remove $\{\#\}$, and the observation $P$ remove $\{\#\}$, that is, $\Sigma' = \Sigma \cup \{\#\}$ and $\Sigma' = \Sigma' \cup \{\#\}$. 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 $P_{h}(t) = P_{h}(t')$ 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 $\#w \in L(B)$. Then $w, \#w \in Q(L(B))$ and $P_{h}(w) = P_{h}(\#w) = w$. 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 $Q^{-1}(w) \cap L(B) = \{w\} \cap L(B) = \{\#w\}$, i.e., $\#w$ 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 $Q$ with $\#w \in L(B)$. Then $w, \#w \in Q(L(B))$ and $P_{h}(w) = P_{h}(\#w) = w$. 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 $Q^{-1}(\#w) \cap L(B) = \{\#w\}$. We now show that $P(\#w) = w \neq w = P(\#w)$, which completes the proof.
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

$$P(w_1) = P(w_3), Q(w_1) = w_2,$$ 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

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

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

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

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 $e \in \Sigma_o \cap \Sigma_{hi}$. The LOC condition states that for any $s, s' \in L$ with $P(s) = P(s')$ and $Q(s)e, Q(s')e \in Q(L)$, it holds that $\delta((p, q, r, s), (e, s, s, e)) = (p \gamma(q, e) \times \gamma r(q, e) \times \gamma s(e, s))$ and $\delta((p, q, r, s), (e, e, e, e)) = (p \gamma(q, e) \times \gamma r(q, e) \times \gamma s(e, e))$. 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 $sue, su'e \in L$. To verify whether this is satisfied, we check whether the language

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

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

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

is the sublanguage of $(\Sigma \setminus \Sigma_{hi})^* \times \{e\} \times (\Sigma \setminus \Sigma_{hi})^* \times \{e\}$ with the property that, for any $(u, u', e) \in [(\Sigma \setminus \Sigma_{hi})^*, e] \cdot (\Sigma \setminus \Sigma_{hi})^* \cdot e$, $P(u) = P(u')$. 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 \(P_{hi}\), i.e., we use the language $L \times \Sigma^* \times L \times \Sigma^* \setminus [(\Sigma \setminus \Sigma_{hi})^*, e] \cdot (\Sigma \setminus \Sigma_{hi})^* \cdot e$.

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

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

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

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 $\Sigma = \{a_1, a_2, \ldots, a_n\}$, $n \geq 2$, such that $L(A) \neq \emptyset$. Then $L(A)$ is prefix-closed, and hence $e \in L(A)$. The universality problem asks whether the language $L(A) = \Sigma^*$. Let $\Sigma_e = \Sigma = \Sigma_A$, and let $\Sigma_A' = \{a' \mid a \in \Sigma_A\}$ be a disjoint copy of $\Sigma_A$. From $A$, we construct an NFA $B$ over $\Sigma = \Sigma_e \cup \Sigma_A'$ with all states marked such that $L(B) = \Sigma^* \cup (\Sigma_e \cup (\Sigma_A \cdot [\Sigma_A']^*)^*)$. See Fig. A.1 for an illustration. Then, $Q(L(B)) = \Sigma^*$. 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 $w \in \Sigma^* \setminus \Sigma_e$. Then $w = te$ for some $t \in L(A)$ and $e \in \Sigma_e$. We show that $L(B)$ is not LOC. Set $s = a_1 a_2 t e$. Notice that $e \in \Sigma_e = \Sigma = \Sigma_A$. Let $t = b_1 \ldots b_m$ and $s' = a_1 a_2 b_1' \ldots b_m' e \in L(B)$; indeed, $s'$ can be generated from state $n_1$. Then, $Q(s') = \Sigma(e \cdot a_1 a_2 b_1' \ldots b_m' e) \in Q(L(B))$, and $P(s) = s = P(s')$. Since $s$ is generated by $B$ only from state $n_1$, because of the initial prefix $a_1 a_2$, and there is no transition labeled by an event from $\Sigma \setminus \Sigma_e = \Sigma_A$ reachable from $(n_1, t)$, there is no $u \in (\Sigma \setminus \Sigma_e)^*$ such that $a_1 a_2 t e = su \in (\Sigma \setminus \Sigma_e)^*$. 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 $e \in \Sigma_e \cap \Sigma_A = \Sigma_A$. Clearly, $Q(s)e, Q(s')e \in Q(L(B))$. Let $t \in (s, s')$. If $t$ is generated from state $n_1$, it can indeed be extended by a string $u \in \Sigma_A \setminus \{e\}$ to generate event $e$; in that case, we have that $P(v) = e$. If $t$ is generated from state $n_1$, it can clearly generate event $e$ from states $n_1$ and $n_2$; thus, if $t = a_1 a_2 t'$ for some $a_1, a_2 \in \Sigma_A$ and $t' \in \Sigma_A$, the universality of $A$ implies that $t' \in e(A)$. Altogether, we have shown that $sue, su'e \in L(B)$ for some $u, u' \in \Sigma^*$ with $P(u) = P(u') = e$.

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 $\Sigma_A$ and $\Sigma_{hi}$ be the respective observation and high-level alphabets. We show that $L$ is MOC wrt. $Q, P$, and $P_{hi}$ iff

$$L \sqsubseteq [\Sigma_{hi} \cap \Sigma_A, Q(L)] \subseteq Q(L) \sqsubseteq \Sigma^* \subseteq [\Sigma_{hi} \cap \Sigma_A, L],$$

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 $Q_2(L) \sqsubseteq \Sigma^*$, 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 $(s, t') \in L \sqsubseteq [\Sigma_{hi} \cap \Sigma_A, Q(L)]$. By the definition of $[\Sigma_{hi} \cap \Sigma_A, s \in L, t' \in Q(L)$, and $P_{hi}(Q(s)) = P_{hi}(t')$. Since $L$ is MOC, there is $t' \in L$ such that $Q(s') = t'$ and $P(s) = P(t')$. However, $P(s) = P(t')$ implies $(s, s') \in L \sqsubseteq \Sigma^*$, 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 $[\Sigma_{hi} \cap \Sigma_A$, we obtain that $(s, t') \in L \sqsubseteq [\Sigma_{hi} \cap \Sigma_A, L]$. Since the inclusion holds, $(s, t') \in Q_2(L) \sqsubseteq \Sigma^*$, which means that there is a pair $(s, s') \in L \sqsubseteq \Sigma^*$ such that $(s, Q(s')) = (s, t')$ and that the strings $s$ and $s'$ belong to $L$ and coincide on $\Sigma_A$, i.e., $P(s) = P(s')$, which was to be shown.
Since commutative diagram of Fig. 4 gives that Lemma 16. The proof makes use of the following well-known result.

We show PSPACE-hardness by reduction from the problem of deciding universality for NFAs with all states marked. Let \( A \) be an NFA over \( \Sigma \) with all states marked. We construct a DFA \( B \) such that \( L(B) = \emptyset \#L(A) \cup \#\Sigma^* \cup L(A) \). It is not difficult to construct \( B \) from \( A \) in polynomial time. Let \( \Sigma_{hi} = \Sigma \cup \{\#\} \) and \( \Sigma_0 = \Sigma \cup \{\@\} \). Then \( L(B) = \Sigma^* \cup \#\Sigma^* \). 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 \( s \in \@H(A) \cup \#\Sigma^* \):
   a. If \( s = \@t' \in \@H(A) \), 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 \( Q^{-1}(\#w) \cap L(B) = \emptyset \#w \), 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 \( \Sigma_0 \subseteq \Sigma_{hi} \), and let \( L_i \subseteq \Sigma_i^* \) be languages, then \( Q(\|\#\|=i, L_i) = \|\#\|=i, Q_i(L_i) \).

Let \( L = \|\#\|=i, L_i \), 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 \( \Sigma_0 \subseteq \Sigma_{hi} \), Lemma 16 implies \( Q(\|\#\|=i, L_i) = \|\#\|=i, Q_i(L_i) \). Projecting to local alphabets gives that \( P_{i|hi}(Q(s)) \subseteq Q_i(L_i) \) and \( P_{i|hi}(t') \subseteq Q_i(L_i) \), \( i = 1, \ldots, n \). 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 \( P_{i|hi,o}(P_{hi}(Q(s))) = P_{i|hi,o}(P_{hi}(t'))) \), and that \( P_{i|hi}(Q(s)) = Q_i(L_i) \) and \( t'_i = P_{i|hi}(t') \). Then MOC of \( L_i \) wrt \( Q_i \). \( P_{i|hi} \). \( P_{i|hi} \) implies that there is \( s'_i \in L_i \) such that \( Q_i(s'_i) = t'_i \) and \( P_{i|hi}(s)_i = t'_i \). We first show that \( \|\#\|=i, s'_i \) is nonempty. It suffices to prove that \( Q(\|\#\|=i, s'_i) \) is nonempty. Since \( \Sigma_0 \subseteq \Sigma_{hi} \), Lemma 16 gives that \( Q(\|\#\|=i, L_i) = \|\#\|=i, Q_i(L_i) \), which is nonempty, because \( t' \in \|\#\|=i, P_{i|hi}(t') \). Hence, there is \( s'_i \in \|\#\|=i, s'_i \) such that \( Q(s'_i) = t'_i \). Furthermore, \( P_{i|hi}(s'_i) = t'_i \) for \( i = 1, \ldots, n \), means that \( P(s) = P(s') \) by \( \|\#\|=i, s_i \).