Non-interference in partial order models

Béatrice Bérard
Sorbonne Universités, UPMC Univ. Paris 06
CNRS UMR 7606, LIP6,
75005, Paris,France
Email: Beatrice.Berard@lip6.fr

Loïc Hélouët
INRIA Rennes
Campus de Beaulieu,
35041 Rennes Cedex, France
Email: loic.helouet@inria.fr

John Mullins
Dept. of Computer & Software Eng.,
École Polytechnique de Montréal, P.O. Box 6079,
Montreal (QC) Canada, H3C 3A7
Email: John.Mullins@polymtl.ca

***Abstract*—Non-interference (NI) is a property of systems stating that confidential actions should not cause effects observable by unauthorized users. Several variants of NI have been studied for many types of models, but rarely for true concurrency or unbounded models. This work investigates NI for High-level Message Sequence Charts (HMSC), a scenario language for the description of distributed systems, based on composition of partial orders. We first propose a general definition of security properties in terms of equivalence among observations, and show that these properties, and in particular NI are undecidable for HMSCs. We hence consider weaker local properties, describing situations where a system is attacked by a single agent, and show that *local NI* is decidable. We then refine local NI to a finer notion of *causal NI* that emphasizes causal dependencies between confidential actions and observations, and extend it to causal NI with (selective) declassification of confidential events. Checking whether a system satisfies local and causal NI and their declassified variants are PSPACE-complete problems.**

I. Introduction

Context. *Non-interference* (NI) has been introduced to characterize the absence of harmful information flow in a system. It ensures that confidential actions of a system can not produce any effect visible by a public observer. The original notion of non-interference in [1] was expressed in terms of language equivalence for deterministic Mealy machines with confidential input and public output. Since then, several variants of information flow properties (IFP) have extended NI to non-deterministic models (transition systems, process algebra, Petri nets,...) and finer notions of observation (simple trace observation, deadlock or branching detection,....) to describe the various observational powers of an attacker. For a given system S, NI is usually defined as: $\pi_V(\llbracket S \setminus C \rrbracket) \equiv \pi_V(\llbracket S \rrbracket)$, where $\equiv$ is language equivalence, $\llbracket S \rrbracket$ denotes the semantics of S, π_V is the projection on a subset V of visible actions of the system, and $S \setminus C$ denotes the model S from which all confidential actions from C are pruned. *Intransitive non-interference* (INI) relaxes NI to handle possible *declassification* of confidential actions. It ensures that confidential actions of a system cannot produce any effect visible by a public observer unless they are declassified, causing so a harmless information flow. This issue has been addressed in [2], by comparing observations of visible actions in runs of a system (hence including runs containing non-declassified confidential actions), and observations of visible actions in runs of the same system that only contain confidential actions that are declassified afterwards. Most IFPs have been expressed as combinations of *basic security predicates* (BSPs) [3], [4] or as a behavioral equivalence under observation contexts [5]. A systematic presentation of IFPs can be found, e.g., in [3]–[5].

Despite the fact that IFPs are always informally expressed in term of causality *i.e.*, confidential activity should not cause observable effects on the public behaviour, they are almost always formalized in terms of interleaving semantics [6]–[9] and hence, do not consider true concurrency or causality. This is clearly a lack in the formalization of IFPs for several reasons. First, from an algorithmic point of view, it is usually inefficient to compute a set of interleavings to address a problem that can be solved on an equivalent partial order representation. Second, from a practical point of view, an attacker of a system may gain more information if he knows that some confidential action has occurred recently in its causal past. Indeed, transactions in a distributed system can leave many traces (visited websites, cookies,...) on machines which are not *a priori* committed to protect confidential actions of third parties. Recently, however, [10] proposed a characterization of NI as a syntactic property of its unfolding in the context of true concurrency semantics for Petri nets, but the technique addresses only safe nets.

Very few results address IFPs for unbounded models. BSPs and NI are proved undecidable for pushdown systems, but decidability was obtained for small subclasses of context-free languages [11]. Decidability of a bisimulation-based strengthened version of NI called *non-deducibility on composition* (NDC) for unbounded Petri nets is proved in [8]. A system satisfies NDC if observation of its visible actions remains indistinguishable from the observation of the system interacting with *any* environment. This result was extended in [9] to INI with selective declassification (INISD).

Contribution. This work considers IFPs for an unbounded true concurrency model, namely High-level Message Sequence Charts (HMSCs). This model, standardized by the ITU [12], is well accepted to represent executions of distributed systems, where security problems are of primary concern. We first define a class of IFPs on HMSCs, as an inclusion relation on observations, following [5], [11] and [13]. We prove that observation inclusion (and hence the simple NI property and most of IFPs) is undecidable for HMSCs. We then characterize decidable sub-classes of the problem: inclusion becomes decidable when the observation of the specified system is regular, and in particular when visible events are located on a single process, and even when the considered HMSC is not regular. We then discuss the meaning of NI in a context where causal dependencies among occurrences of events are considered. This leads to a new notion called *causal interference* for HMSCs. Causal interference detects interference as soon as an attacker can observe occurrences of confidential actions from visible events, and furthermore, one of the observed events is a consequence of the confidential one. We finally relax causal interference in the context of declassification. We introduce *intransitive causal non-interference* that considers observable causal dependencies among confidential and visible events as safe, as soon as a declassification occurs in between. We show that all local variants of these problems are PSPACE-complete.

Outline. Model definitions are given in Section II. In Section III, we first introduce observations of HMSCs, then we formally define inclusion problems and non-interference and show their undecidability for HMSCs. In Section IV, we characterize information flow in a single finite scenario, and rephrase it in terms of coloring, and show in Section V that NI in HMSCs can be solved by reusing this coloring, and is PSPACE-complete for cases where observation is limited to a single process. We extend this framework to declassification and we prove the decidability of a local version of interference with (selective) declassification in Section VI. We compare our approach with related works, and conclude in Section VII.

II. Preliminaries

This section introduces the models that will be used throughout the paper, namely automata and High-level Message Sequence Charts (HMSCs), with their associated languages. Message Sequence Charts (MSCs) are formal representations of distributed executions, *i.e.*, chronograms, that are frequently used to depict the behavior of a set of asynchronous communicating processes. This simple graphical representation emphasizes on messages and localization of actions, with partial order semantics. The model of HMSCs, standardized by the ITU [12], was proposed to describe the more elaborate behaviors of distributed systems, for instance those of communication protocols, by combining MSCs. Illustrations of MSCs and HMSCs are given in Figures 1 and 2 of Sections IV and V. HMSCs are used to describe typical scenarios for the use of distributed systems, and then serve as requirements. They can also be used as input to generate code skeletons for distributed systems. Hence, an information leak that appears in these early requirement is likely to be a feature of the final system. It is then interesting to find these leaks at early design stages. Another interesting point with HMSCs is their expressive power: they define behaviors of systems with asynchronous communications, which are not necessarily finite state systems. They are uncomparable with Petri nets, for instance. Answering interference questions for HMSCs provides security techniques for a whole class of infinite systems that can not be modeled with other formalisms.

Let Σ be a finite alphabet. A word over Σ is a sequence $w = a_1 a_2 \dots a_n$ of letters from Σ, and Σ^* denotes the set of finite words over Σ, with ε the empty word. A *language* is a subset L of Σ^*. A *finite automaton* over Σ is a tuple $\mathcal{A} = (S, \delta, s_0, F)$, where S is a finite set of states, $s_0 \in S$ is the initial state, $F \subseteq S$ is a set of accepting states, and $\delta \subseteq S \times \Sigma \times S$ is a transition relation. A word $w = a_1 \dots a_n \in \Sigma^*$, is accepted by $\mathcal{A}$ if there exists a sequence of transitions labeled by a_i's, starting from s_0 and ending in an accepting state, *i.e.*, $\exists s_1, \dots s_n, \forall i \in 1..n, (s_{i-1}, a_i, s_i) \in \delta$ with $s_n \in F$. It is well known that automata accept *regular languages*.

A *Labeled Partial Order* (LPO) over alphabet Σ is a triple $(E, \leq, \alpha)$ where $(E, \leq)$ is a partially ordered set (poset) and $\alpha : E \to \Sigma$ is a labeling of E by letters of Σ. The set of all LPOs over alphabet Σ is denoted by $LPO(\Sigma)$. Given a relation $R \subseteq X \times X$ on some set X, we denote by R^* the transitive and reflexive closure of R.

Definition 1 (MSC): A *Message Sequence Chart* over finite sets $\mathbb{P}$ of processes, $\mathbb{M}$ of messages and finite alphabet A, is a tuple $M = (E, (\leq_p)_{p \in \mathbb{P}}, \alpha, \mu, \phi)$, where

- E is a finite set of *events*, partitioned as $E = E_S \uplus E_R \uplus E_I$, according to the type of event considered: message sending, reception, or internal action (also called *atomic action*);
- $\phi : E \to \mathbb{P}$ is a mapping associating with each event the process that executes it. Hence, the sets $E_p = \phi^{-1}(\{p\})$ for $p \in \mathbb{P}$, also form a partition of E;
- For every process $p \in \mathbb{P}$, the relation $\leq_p \subseteq E_p \times E_p$ is a total ordering on events located on process p;
- $\mu \subseteq E_S \times E_R$ is a relation symbolizing message exchanges, such that if $(e, f) \in \mu$ with $e \in E_p$ and $f \in E_q$, then $p \neq q$. Furthermore, it induces a bijection from E_S onto E_R, so with a slight abuse of notation, $(e, f) \in \mu$ is also written as $f = \mu(e)$. With each pair $(e, f) \in \mu$ is associated a message in $\mathbb{M}$;
- α is a mapping from E to $\Sigma \subseteq (\mathbb{P} \times \{!, ?\} \times \mathbb{P} \times \mathbb{M}) \cup (\mathbb{P} \times A)$, associating a label with each event. The labeling is consistent with μ: if $f = \mu(e)$, with associated message m, sent by process p to process q, then $\alpha(e)$ is written as $p!q(m)$ and $\alpha(f)$ as $q?p(m)$. If e is an internal action a located on process p, then $\alpha(e)$ is of the form $p(a)$. The labeling is extended by morphism over E^*.

We write $\leq_M$ for the relation $\leq_M = (\bigcup_{p \in \mathcal{P}} \leq_p \cup \mu)^*$, and require that for any MSC M, $\leq_M$ is a partial order, that is a reflexive, transitive, and acyclic relation. When clear from the context, we will simply write $\leq$ instead of $\leq_M$.

We denote by $\mathcal{M}sc(\mathbb{P}, \mathbb{M}, A)$ the set of all MSCs over the sets $\mathbb{P}$ of processes, $\mathbb{M}$ of messages, and alphabet A. The usual terminology and definitions of partially ordered sets apply to MSCs. For an event $e \in E$ of M, the set of *predecessors* of e is M is $\uparrow(e) = \{f \in E \mid f \leq e\}$ and the set of *successors* of e in M is $\downarrow(e) = \{f \in E \mid e \leq f\}$. Given a subset E' of E, the *restriction* of M to E', denoted by $M_{|E'}$, is the LPO $\big(E', \leq_M \cap (E' \times E'), \alpha_{|E'}\big)$ and we denote by $M \setminus E'$ the restriction of M to $E \setminus E'$.

Definition 2: A *linear extension* of an MSC with n events $M = (E, (\leq_p)_{p \in \mathbb{P}}, \alpha, \mu, \phi)$ is a sequence $r = e_1 e_2 \dots e_n$ of all events of M such that for every $j > k$ $e_j \not\leq e_k$. A *linearization* of M is a word $w \in \Sigma^*$ such that there exists a linear extension

r of M with $w = \alpha(r)$. The *language* of M, written $\mathcal{L}(M)$, is the set of all linearizations of M.

The language of a MSC is hence defined over alphabet $\Sigma = \{p!q(m) \mid p, q \in \mathbb{P} \wedge m \in \mathbb{M}\} \cup \{p?q(m) \mid p, q \in \mathbb{P},\ m \in \mathbb{M}\} \cup \{p(a) \mid p \in \mathbb{P},\ a \in A\}$. To design more elaborate behaviors, including choices and iterations, MSCs are composed. A key ingredient is sequential composition, that assembles MSCs processwise to form larger MSCs.

Definition 3: Let $M_1 = (E_1, (\leq_{1,p})_{p\in\mathbb{P}}, \alpha_1, \mu_1, \phi_1)$ and $M_2 = (E_2, (\leq_{2,p})_{p\in\mathbb{P}}, \alpha_2, \mu_2, \phi_2)$ be two MSCs defined over disjoint sets of events. The *sequential composition* of M_1 and M_2, denoted by $M_1 \circ M_2$ is the MSC
$M_1 \circ M_2 = (E_1 \cup E_2, (\leq_{1\circ 2,p})_{p\in\mathbb{P}}, \alpha_1 \cup \alpha_2, \mu_1 \cup \mu_2, \phi_1 \cup \phi_2)$, where $\leq_{1\circ 2,p} = \left(\leq_1 \cup \leq_2 \cup \phi_1^{-1}(\{p\}) \times \phi_2^{-1}(\{p\})\right)^*$ and $f_1 \cup f_2$ denotes a function defined over $Dom(f_1) \cup Dom(f_2)$, that associates $f_1(x)$ with every $x \in Dom(f_1)$ and $f_2(x)$ with every $x \in Dom(f_2)$.

This (associative) operation, also called concatenation, can be extended to n MSCs and is used to give a semantics to higher level constructs, namely HMSCs. Roughly speaking, an HMSC is a finite automaton where transitions are labeled by MSCs. It describes a **set of MSCs** obtained by assembling (using sequential composition) MSCs that appear along paths.

Definition 4 (HMSC): A *High-level MSC* (HMSC) is a tuple $H = (N, \rightarrow, \mathcal{M}, n_0, F)$, where N is a set of nodes, $\mathcal{M}$ is a finite set of MSCs, $\rightarrow \subseteq N \times \mathcal{M} \times N$ is a transition relation, $n_0 \in N$ is the initial node, and F is a set of accepting nodes.

As for any kind of automaton, paths and languages can be defined for HMSCs. A *path* ρ of H is a sequence of transitions $t_1 t_2 \ldots t_k$ such that for each $i \in \{1, \ldots, k\}$, $t_i = (n_i, M_i, n'_i)$ belongs to $\rightarrow$, with $n'_i = n_{i+1}$ for each $i \leq k-1$. A path ρ is *accepting* if it starts from node n_0 (*i.e.*, $t_1 = (n_0, M_1, n_1)$), and it terminates in a node of F (*i.e.*, $t_k = (n_k, M_k, n'_k)$ for some $n'_k \in F$).

Definition 5: Let $\rho = t_1 t_2 \ldots t_k$ be a path of a HMSC H. The MSC associated with ρ is $M_\rho = h_1(M_1) \circ h_2(M_2) \cdots \circ h_k(M_k)$ where each h_i is an isomorphism that guarantees $\forall j \neq i, h_i(E_i) \cap h_j(E_j) = \emptyset$.

More intuitively, the MSC associated with a path is obtained by concatenating MSCs encountered along this path after renaming the events to obtain disjoint sets of events. To simplify notation, we often drop the isomorphisms used to rename events, writing simply $M_\rho = M_1 \circ M_2 \circ \cdots \circ M_k$.

With this automaton structure and the sequential composition of MSCs, an HMSC H defines a set of *accepting paths*, denoted by $\mathcal{P}_H$, a set of MSCs

$$\mathcal{F}_H = \{M_\rho \mid \rho \in \mathcal{P}_H\},$$

and a linearization language $\mathcal{L}(H) = \bigcup_{M\in\mathcal{F}_H} \mathcal{L}(M)$. Finally note that a single MSC M can be seen as a particular HMSC H_M with a single transition (n_0, M, n_1) between two nodes, hence a single path from initial node n_0 to final node n_1 and language $\mathcal{L}(H_M) = \mathcal{L}(M)$.

It is well known that the linearization language of a HMSC is not necessarily regular, but rather a closure of a regular language under partial commutation, which yields many undecidability results (see for instance [14], [15]). This does not immediately mean that all IFPs are undecidable for HMSCs: indeed, we later define non-trivial and meaningful subclasses of HMSCs and observations for which these problems become decidable.

III. Observation and non-interference for HMSCs

The power of an external observer can be described by an observation function, mapping every behavior of a system to some observables. In [3], [4], [11], observation functions are seen as some particular kind of language theoretic operations (projection, morphism, insertion, deletion of letters,...), and in [13], they are defined as combinations of rational operations (transductions, intersections, unions of rational languages).

In a distributed context, visible events can originate from sensors that belong to different processes, and occurrences of such events can easily be recorded. If the system is equipped with vectorial clocks, one can also record causal dependencies among observed events. However, even if those visible events happen to be observed in some particular linear order, this does not mean that this order corresponds to the actual execution, because the processes are not synchronized. Hence, while weaker than a linearization, the natural and realistic notion of observation for distributed computations is a labeled partial order, where events that are not surely causally dependant are considered concurrent.

Definition 6: An observation function is a mapping from $\mathcal{M}sc(\mathbb{P}, \mathbb{M}, A)$ to $LPO(B)$ for some alphabet B.

As proposed in [4] with the notion of *views*, the alphabet labeling events that occur during an execution of a system can be partitioned as $\Sigma = V \uplus C \uplus N$ with visible, confidential and internal (neutral) labels. Actions with labels in V can be observed while actions labeled in C are confidential and should be hidden. Internal actions have labels in N and are not observable *a priori*, but need not be kept secret. Subsequently, depending on their labels, events are also called visible, confidential, or internal events.

Various observation functions can be defined from such a partition. The most natural ones are restrictions to visible events, and pruning of confidential actions, which are standard operations in language based non-interference literature, but need to be precisely defined in a partial order setting. Let $M = (E, (\leq_p)_{p\in\mathbb{P}}, \alpha, \mu, \phi)$ be an MSC with labeling alphabet Σ. We consider the following observation functions:

- **identity:** the identity $id(M) = M$ outputs the same LPO as the executed MSC;
- **Restriction:** $\mathcal{O}^V(M)$ is the LPO obtained by restriction of M to $E \cap \alpha^{-1}(V)$. Intuitively, $\mathcal{O}^V(M)$ represents the visible events and their causal dependencies that one may observe during the complete execution of M; Note that restriction to $\alpha^{-1}(V)$ suffices, as $\leq$ is a transitive relation.
- **Pruning:** $\mathcal{O}^V_{\setminus C}(M) = \mathcal{O}^V(M \setminus \downarrow (\alpha^{-1}(C)))$ is a function that prunes out the future of confidential events from M. Intuitively, $\mathcal{O}^V_{\setminus C}(M)$ represents the visible events and their causal dependencies, observed when no confidential event is executed within M;

- **Localization:** $\mathcal{O}^p(M) = \mathcal{O}^V(M_{|E_p})$, for a given process $p \in \mathbb{P}$, is the observation of visible events of M restricted to those events located on process p. Note that $\mathcal{O}^p(M)$ is a total order. In a distributed setting, $\mathcal{O}^p(M)$ is particularly interesting, as it represents the point of view of a single process $p \in \mathbb{P}$, considered as the attacker of the system. We hence assume no restriction on the set of events that can be executed and observed by p, and let $V = \Sigma_p = \alpha(E_p)$ when using $\mathcal{O}^p$.

As noticed by [11] in a language setting, information flow properties of a system S are usually defined as compositions of atomic propositions of the form $op_1(S) \subseteq op_2(S)$. Changing the observation functions (or the partition of Σ) leads to a variety of atomic properties. Information flow properties of MSCs can be defined similarly.

Definition 7: Let $\mathcal{O}_1, \mathcal{O}_2$ be two observation functions over $\mathcal{M}sc(\mathbb{P}, \mathbb{M}, A)$. An MSC M *satisfies* the *inclusion property* for $\mathcal{O}_1, \mathcal{O}_2$, written $\sqsubseteq_{\mathcal{O}_1,\mathcal{O}_2}(M)$, if $\mathcal{L}(\mathcal{O}_1(M)) \subseteq \mathcal{L}(\mathcal{O}_2(M))$.

For a single MSC M, the classical notion of non-interference by language equivalence translates as follows:

Definition 8: An MSC M is *non-interferent* if $\mathcal{L}(\mathcal{O}^V(M)) = \mathcal{L}(\mathcal{O}^V_{\setminus C}(M))$. Otherwise M is said *interferent*.

In order to extend an observation $\mathcal{O}$ to an HMSC H, a first way consists in applying $\mathcal{O}$ to all MSC in $\mathcal{F}_H$, defining $\mathcal{O}(H) = \{\mathcal{O}(M) \mid M \in \mathcal{F}_H\}$. In particular:

$$\begin{aligned} &\mathcal{O}^{V,\circ}(H) = \{\mathcal{O}^V(M) \mid M \in \mathcal{F}_H\}, \\ &\mathcal{O}^{V,\circ}_{\setminus C}(H) = \{\mathcal{O}^V_{\setminus C}(M) \mid M \in \mathcal{F}_H\}, \text{ and} \\ &\mathcal{O}^{p,\circ}(H) = \{\mathcal{O}^p(M) \mid M \in \mathcal{F}_H\} \end{aligned}$$

Observation functions $\mathcal{O}^{V,\circ}, \mathcal{O}^{V,\circ}_{\setminus C}$ and $\mathcal{O}^{p,\circ}$, however, do not take in account the structure of the HMSC generating $\mathcal{F}_H$, and furthermore, they are not necessarily compositional. In general, an observation function $\mathcal{O}$ is not a morphism with respect to the concatenation, that is, $\mathcal{O}(M_1 \circ M_2) \neq \mathcal{O}(M_1) \circ \mathcal{O}(M_2)$. This drawback was already observed in [16] for projections of MSCs: in general, $\mathcal{O}^V(M_1 \circ M_2) \neq \mathcal{O}^V(M_1) \circ \mathcal{O}^V(M_2)$. Hence, checking inclusion for HMSCs may require to consider properties of complete sequences of MSCs as a whole, raising algorithmic difficulties, or even undecidability. Other, more interesting ways to extend observations to HMSCs, are to assemble observations of MSCs piecewise, following the automaton structure of HMSCs, or to forbid MSCs containing confidential events:

$$\mathcal{O}^{V,\bullet}(H) = \{\mathcal{O}^V(M_1) \circ \cdots \circ \mathcal{O}^V(M_k) \mid M_1 \circ \cdots \circ M_k \in \mathcal{F}_H\},$$

$$\begin{aligned} \mathcal{O}^{V,\bullet}_{\setminus C}(H) = \{\mathcal{O}^V(M_1 \circ \cdots \circ M_k) \mid \quad & M_1 \circ \cdots \circ M_k \in \mathcal{F}_H \\ & \wedge \forall i, \alpha(E_i) \cap C = \emptyset\}, \end{aligned}$$

$$\mathcal{O}^{p,\bullet}(H) = \{\mathcal{O}^p(M_1) \circ \cdots \circ \mathcal{O}^p(M_k) \mid M_1 \circ \cdots \circ M_k \in \mathcal{F}_H\},$$

where concatenation of LPOs is performed processwise like for MSCs. The observation $\mathcal{O}^{V,\bullet}_{\setminus C}(H)$ is of particular interest, as it describes observations of MSCs in $\mathcal{F}_H$ that do not contain confidential events. Moreover, since $\mathcal{O}^p(M)$ is a total order, $\mathcal{O}^p$ satisfies the morphism property, which implies $\mathcal{O}^{p,\circ}(H) = \mathcal{O}^{p,\bullet}(H)$. The definitions of inclusion and non-interference can now be extended to HMSCs:

Definition 9: An HMSC H satisfies the *inclusion* problem for $\mathcal{O}_1, \mathcal{O}_2$ (written $\sqsubseteq_{\mathcal{O}1,\mathcal{O}_2}(H)$) if $\mathcal{L}(\mathcal{O}_1(H)) \subseteq \mathcal{L}(\mathcal{O}_2(H))$. It is *non-interferent* if $\mathcal{L}(\mathcal{O}^{V,\circ}_{\setminus C}(H)) = \mathcal{L}(\mathcal{O}^{V,\circ}(H))$.

We say that an observation function $\mathcal{O}$ for a set of HMSCs $\mathcal{H}$ is *regular* if $\mathcal{L}(\mathcal{O}(H))$ is regular for every $H \in \mathcal{H}$ and that $\mathcal{O}$ is *effectively regular* if for every $H \in \mathcal{H}$, one can compute a finite automaton recognizing $\mathcal{L}(\mathcal{O}(H))$. Observation function $\mathcal{O}^p$ is an example of effectively regular observation function. Defining (effective) regularity for sets of HMSCs leads to characterize observation functions that have good properties for infinite classes of HMSCs. For instance, for the class of locally-synchronized HMSCs defined in [14], that have regular linearization languages, $\mathcal{O}^{V,\circ}$ is effectively regular.

HMSC languages are not always regular and the observation of an HMSC needs not be regular either. It was proved in [16] that HMSC projections are close to Compositional Message Sequence Charts. Even when a projection of an HMSC is an HMSC language (*i.e.*, a language recognizable by an HMSC), equivalence, inclusion or emptiness of intersection are undecidable. In fact, due to the close relationship between HMSCs and Mazurkiewicz traces, most properties requiring to compare languages or partial order families are undecidable for HMSCs ([14], [15], [17]). So, given two HMSCs H_1 and H_2, one can not decide if $\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$, nor if $\mathcal{F}_{H_1} \subseteq \mathcal{F}_{H_2}$. We hence have the following result:

Proposition 10: Let H be an HMSC. The inclusion problem $\sqsubseteq_{\mathcal{O}_1,\mathcal{O}_2}(H)$ is undecidable in general, even if $\mathcal{O}_1$ **or** $\mathcal{O}_2$ is an effective regular observation function. It is *decidable* if $\mathcal{O}_1$ **and** $\mathcal{O}_2$ are both effective regular functions.

Proof Sketch. The proof is a reduction from the inclusion problem $\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$ for two HMSCs H_1 and H_2. We build an HMSC H, that behaves like H_1 or H_2 if a confidential action can occur, and like H_2 otherwise, and choose observation functions $\mathcal{O}_1 = id, \mathcal{O}_2 = \mathcal{O}^V_{\setminus C}$. Then inclusion $\sqsubseteq_{\mathcal{O}_1,\mathcal{O}_2}(H)$ holds iff $\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$. The construction of this HMSC H is detailed in Appendix. The decidability part is an immediate consequence of decidability of inclusion for regular languages. □

Note that the exact complexity of inclusion problems in the decidable cases depends on the size of the automata recognizing $\mathcal{L}(\mathcal{O}_1(H))$ and $\mathcal{L}(\mathcal{O}_2(H))$, and hence does not immediately fall into a determined complexity class holding for any pair of effectively regular observation functions. From Proposition 10, we obtain the following undecidability result:

Corollary 11: Non-interference for HMSCs is undecidable.

In the rest of the paper, we propose to weaken the requirements of NI, by choosing appropriate observation functions which will be shown effectively regular for any set $\mathcal{H}$ of HMSCs. This restriction is necessary, due to the fact that effective regularity of an observation function is undecidable (see proof of Theorem 12 in Appendix).

Theorem 12: Let $\mathcal{O}$ be an observation function on alphabet $\Sigma = C \uplus V \uplus N$ and let $\mathcal{H}$ be a set of HMSCs. One cannot decide in general if $\mathcal{O}$ is effectively regular for $\mathcal{H}$.

IV. MSCs coloring

Interference is frequently described as causal dependencies between confidential actions and observable ones. The problem, however, is often defined in terms of languages, with interleaved representation, even for true concurrency models. In this section, we first show that interference in a single MSC can be defined in terms of causal dependencies from confidential events (in C) to visible ones (in V). We then show that checking existence of such dependencies can be performed via a coloring of events.

For a single MSC, comparing observations $\mathcal{O}^V$ and $\mathcal{O}^V_{\setminus C}$ defined in Section III suffices to highlight dependencies between confidential and visible actions. Hence, interference in a *single* MSC can be defined through causality:

Proposition 13: Let M be an MSC with labeling alphabet $\Sigma = C \uplus V \uplus N$ and set of events E. Then, M is *interferent* if and only if there are two events e, f such that $\alpha(e) \in C$, $\alpha(f) \in V$, and $e \leq f$.

The proof of this proposition is given in Appendix. The result shows that even if interference in a MSC M was defined in terms of languages equivalence (Def. 8), it can also be equivalently characterized as a property of its causal dependencies, *i.e.*, without computing the whole interleaved representation of M. This relation between causal dependencies and interference calls for a graphical interpretation of interference in MSCs, represented as a propagation of a black token inherited from confidential actions along causal dependencies. Intuitively, any confidential action and successors of actions marked with a black token are also marked with a black token and every process containing a black action is also marked as black. Though the black/white coloring of MSCs is not essential to prove interference, it will be used later to detect information flows in HMSCs.

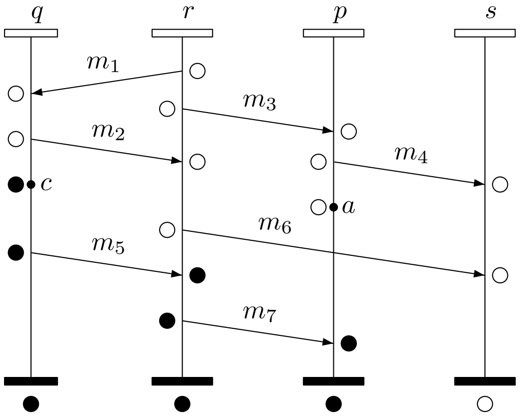

Fig. 1. An MSC M_{bw} tagged with black and white tokens

Definition 14 (MSC and process coloring): Let M be an MSC. An event e is *black* if $\alpha(\uparrow e) \cap C \neq \emptyset$, and *white* otherwise. A process $p \in \mathbb{P}$ is *black after* M (resp. white after M) if there exists a black event located on p (resp. no black event on p).

Intuitively, a black process can detect occurrences of confidential events, as it executes events that are causal consequences of confidential events. Clearly, an MSC is non-interferent if and only if it does not contain visible black events. Figure 1 shows a coloring of an MSC in black and white. The alphabet of confidential actions is $C = \{q(c)\}$ and contains the label of the atomic action c executed by process q. We attach a black token to every black event and a white token to other events. Similarly, we indicate with a black/white token below process lines whether a process has met a black token during its execution. In this example, process p can detect occurrences of c (it is black after M_{bw}), but process s cannot.

Deciding if an MSC is interferent, or equivalently if it contains a visible black event then consists in finding a path from a confidential event to a visible one in an acyclic graph where events are seen as vertices and pairs of events (e, f) in $(\cup_{p \in \mathcal{P}} \leq_p) \cup \mu$ as edges. Since an event has at most two immediate successors, the graph to consider has at most $n = |E_M|$ vertices and $2n$ edges. Hence, coloring of MSCs and interference detection can be performed in linear time as a graph exploration starting from confidential events.

We now show that deciding the black/white status of a process along a sequence of MSCs of arbitrary size can be performed with bounded memory.

Proposition 15: Let M_1, M_2 be two MSCs with labels in $\Sigma = C \uplus V \uplus N$. Then, process $p \in \mathbb{P}$ is black after $M_1 \circ M_2$ iff it is black after M_1, or it is black after M_2, or there exists a process q black after M_1 and a pair of events $e \leq f$ in M_2 such that e is located on q and f is located on p.

This important property means that it is sufficient to remember the black/white status of each process after concatenation $M_1 \circ \cdots \circ M_k$ along a path of an HMSC to compute the status of process p after concatenation $M_1 \circ \cdots \circ M_k \circ M_{k+1}$.

V. Local and causal non interference

Despite general undecidability of the inclusion problem and non-interference (Proposition 10), the problem becomes decidable with regular observation functions. As effective regularity of an observation function is undecidable (Thm. 12), we must rely on subclasses of observations for which this property is guaranteed. We show in this section that observation functions describing the discriminating power of a *single process* are effectively regular. In this restricted setting, it is then possible to decide whether a process $p \in \mathbb{P}$ can detect occurrences of confidential actions. As HMSCs explicitly specify distribution of actions on processes, exhibiting the behavior of a fixed process within an HMSC specification is an easy task. In this section, we show that this *local* setting allows for the definition of two decidable notions of non-interference.

A. Local interference

Considering the attacker of a system as a single process $p \in \mathbb{P}$, with action labels in some alphabet $\Sigma_p = \alpha(E_p)$, we should assume that process p does not execute confidential actions, that is $C \cap \Sigma_p = \emptyset$. In a similar way, the observation power of a single process should be restricted to its own events, hence we can safely set $V = \Sigma_p$. The definition of non-interference (Def. 8) proposed in section III can accommodate this particular partition of the alphabet. From now on, we consider this restricted form of non-interference, and call it *local non-interference*.

For a single MSC, it is then defined as satisfaction of two inclusion problems, with $\mathcal{O}^V_{\setminus C}$ and $\mathcal{O}^p$ as observation functions. This property can be verified by checking whether $\downarrow(\alpha^{-1}(C)) \cap E_p = \emptyset$ that is checking if no causal consequence of a confidential action is located on process p. In other words, one need to check that p is not marked with a black token. As explained in section IV, this can be performed in linear time. We can now look at local interference for HMSCs.

Definition 16: Let H be an HMSC over a set of processes $\mathbb{P}$, with labeling alphabet $\Sigma = V \uplus C \uplus N$, such that $\Sigma = \uplus_{p \in \mathbb{P}} \Sigma_p$ with $V = \Sigma_p$. Then HMSC H is said *locally non-interferent* (w.r.t. process p) if $\mathcal{L}(\mathcal{O}^{V,\bullet}_{\setminus C}(H)) = \mathcal{L}(\mathcal{O}^{V,\circ}(H))$.

Intuitively, local interference holds when an observer can not distinguish in $\mathcal{F}_H$ behaviors that are concatenations of MSCs containing no confidential event, and other behaviors.

Proposition 17: $\mathcal{O}^p$ is effectively regular, and if $V = \Sigma_p$, then $\mathcal{O}^{V,\bullet}_{\setminus C}$ and $\mathcal{O}^{V,\circ}$ are effectively regular.

Proof Sketch. For any H, we can build an automaton $\mathcal{A}_p(H)$ that recognizes the projection of all MSCs in $\mathcal{F}_H$ on p. As concatenation of MSCs imposes a total order on events of the same process, these projection are concatenations of finite sequences of events (local projections of MSCs along transitions of H). Hence $\mathcal{A}_p(H)$ has transitions using labels of event located on process p, and just needs to remember the transition of H that is recognized (the current MSC under execution), and an integer symbolizing the last event of the current MSC executed by p. Similarly, we can design an HMSC $H_{\setminus C}$ where transitions are labeled by MSC that do not contain confidential events, and hence an automaton $\mathcal{A}'_p(H)$ that accepts only projections on p of sequences of MSCs with only white events. Hence $\mathcal{A}'_p(H)$ recognizes $\mathcal{O}^{V,\bullet}_{\setminus C}(H)$. Last, if $V = \Sigma_p$, then $\mathcal{O}^{V,\circ}(H) = \mathcal{O}^p(H)$. □

Corollary 18: The problem of deciding local interference of an HMSC H with respect to a given process $p \in \mathbb{P}$ is PSPACE-complete.

Proof Sketch. Using the results of proposition 17, the problem consists in comparing the languages of two automata (whence the complexity in PSPACE). For the hardness part, we can also show that any regular language inclusion problem can be encoded as a local interference problem. □

Local interference is decidable, and describes a situation where a process can discover that the running execution of the system contains *or will contain* a confidential action. However, local interference does not distinguish between a situation where an observation is a causal consequence of some confidential action and a situation where observation and confidential action highlighted by the interference are concurrent. Language-based comparison of observations (and also in general most of language-based non-interference settings) only characterize the possibility for an attacker to reveal occurrence of confidential actions during a run of a system.

B. *Causal interference*

We first give a concrete example showing that interference is much more dangerous when the confidential event that is detected lays within the causal past of some observation. Nowadays, a lot of attention is devoted to privacy. However, it is well known that users spread a lot of information to visited sites when browsing the web. This information is not always local information (cookies, cache, etc.) that can be erased by users if needed. It can also be information stored elsewhere on the web: logs, forms, etc.. When observation of a causal consequence of a confidential action (Mr X has bought a book on commercial site Y) by an attacker indicates that a confidential operation has occurred, this may also mean that classified information might be available at some vulnerable site (the credit card details of X are stored somewhere on Y's website). Hence, characterizing interference where confidential actions and observations are causally related, is important.

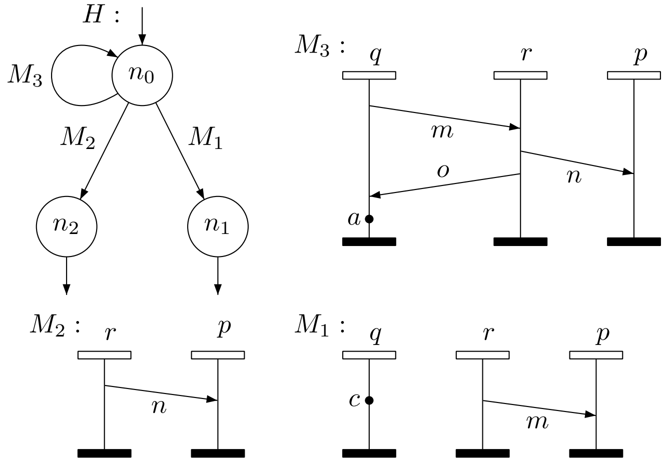

Fig. 2. An interferent HMSC

On the HMSC depicted in Figure 2, the projection of MSCs recognized by H on p is the language $(?n)^*.(?m+?n)$, and every MSC with projection on p in $(?n)^*.?m$ is the projection of a concatenation of several occurrences of M_3, followed by one occurrence of M_1, which contains a confidential event. According to definition 16, this HMSC is locally interferent. However, when observing arrival of message m, process p can deduce that it is currently executing a behavior *in which a confidential action occurs*, but not that this action *has already occurred*. This means in particular that NI does not always characterize a cause to effect relation among hidden actions and observation. To overcome this weakness of language-based information flow characterizations, the notion of NDC (Non-Deducibility on Composition) has been proposed to detect when confidential actions *cause* observable effects. Formally, NDC says that a system S composed with any process R (that enables/forbids confidential events) is observationally equivalent to S.

In the rest of this section, we propose a decidable notion of *causal interference* (still with respect to a fixed attacker $p \in \mathbb{P}$). It emphasizes on causal dependencies between confidential and visible actions of the system. Bearing in mind that a black event located on process p is a consequence of a confidential event, we show that causal dependencies can be highlighted in terms of an observation function built using the black/white tokens attached to events and processes within an MSC. We want to check if a process p can detect whether some confidential action has occurred in the causal past of its observed events. In other words, this means that the projection on p of an execution of H contains a black event, and that equivalent projections also contain black events.

Definition 19: For an HMSC H and a process $p \in \mathbb{P}$, H is *causally non-interferent* (with respect to p) if for every MSC M in $\mathcal{F}_H$ such that M contains a black event on process p, there exists another MSC M' in $\mathcal{F}_H$ such that

- M' contains no black event on process p, and
- $\mathcal{L}(\mathcal{O}^p(M)) = \mathcal{L}(\mathcal{O}^p(M'))$

Causal non-interference is weaker that NDC: it compares the observations of an HMSC with the observations that are still possible whitout confidential events. NDC compares a behavior of a specification with a specification controlled by a process R, in which some confidential events can be allowed.

Theorem 20: For a fixed set of processes $\mathbb{P}$, deciding causal non-interference of an HMSC H with respect to a process $p \in \mathbb{P}$ is PSPACE-complete.

We prove this theorem in several steps. We use the result of Proposition 15, *i.e.*, the fact that black/white coloring of processes at the end of a sequence of concatenated MSCs can be done by remembering the status of processes after each MSC. This property holds for MSCs built along paths of HMSCs, and is used (in Proposition 21) to build HMSCs that recognize MSCs that belong to $\mathcal{F}_H$ and after which a fixed process is black (or similarly remains white). These HMSCs contain nodes of H, but remember for each node n whether processes are black or white after an MSC built along a path ending in n. Then causal interference will be reduced to an inclusion problem of effectively regular observation functions.

Proposition 21: Let H be an HMSC, $p \in \mathbb{P}$, and $\Sigma = C \uplus V \uplus N$. Then, one can build:

- an HMSC $H^{B,p}$ that recognizes MSCs from $\mathcal{F}_H$ after which p is a black process.
- an HMSC $H^{W,p}$ that recognizes MSCs from $\mathcal{F}_H$ after which p is a white process.

of sizes in $O(|H|.2^{|\mathbb{P}|})$.

Proof Sketch. The nodes of the HMSCs built in the proof memorize a node of the original HMSC, plus information on the color of each process (according to Proposition 15, this is the only information needed to remember the color of a process along a path of H). Accepting nodes require p to be black in $H^{B,p}$, and white in $H^{W,p}$. □

We are now ready to prove theorem 20:

Proof of theorem 20. Following the construction of $H^{B,p}$ or $H^{W,p}$, we can define $\mathcal{A}_p^{B,p}$ and $\mathcal{A}_p^{W,p}$ as the automata that recognize the projections of $H^{B,p}$ or $H^{W,p}$. Let us denote by $\mathcal{O}^{B,p}(H) = \{\mathcal{O}_p(M) \mid M \in \mathcal{F}_H \wedge p \text{ is black after } M\}$ the observation function that returns the projection and by $\mathcal{O}^{W,p}(H) = \{\mathcal{O}_p(M) \mid M \in \mathcal{F}_H \wedge p \text{ is white after } M\}$. Clearly, we have $\mathcal{L}(\mathcal{A}_p^{B,p}) = \mathcal{L}(\mathcal{O}^p(H^{B,p})) = \mathcal{L}(\mathcal{O}^{B,p}(H))$ and $\mathcal{L}(\mathcal{A}_p^{W,p}) = \mathcal{L}(\mathcal{O}^p(H^{W,p})) = \mathcal{L}(\mathcal{O}^{W,p}(H))$, so $\mathcal{O}^{B,p}$ and $\mathcal{O}^{W,p}$ are effectively regular.

Deciding causal interference of H with respect to $p \in \mathbb{P}$ consists in deciding the inclusion problem $\sqsubseteq_{\mathcal{O}^{B,p},\mathcal{O}^{W,p}}$ for H, that is checking whether $\mathcal{L}(\mathcal{A}_p^{B,p}) \subseteq \mathcal{L}(\mathcal{A}_p^{W,p})$. Clearly, if H is of size n, then $H^{B,p}$ and $H^{W,p}$ are of size in $O(n.2^{|\mathbb{P}|})$, and so are $\mathcal{A}_p^{B,p}$ and $\mathcal{A}_p^{W,p}$. Then, checking inclusion of $\mathcal{L}(\mathcal{A}^{B,p})$ into $\mathcal{L}(\mathcal{A}_p^{B,p})$ is equivalent to checking $\mathcal{L}(\mathcal{A}_p^{B,p}) \cap \mathcal{L}(\overline{\mathcal{A}^{B,p}}) = \emptyset$. Emptiness of regular language is an NLOGSPACE problem, but the size of the automaton that recognizes the intersection is in $O(n.2^{|\mathbb{P}|}.2^{n.2^{|\mathbb{P}|}})$, that is inclusion can be performed with space in $O(log(n) + |\mathbb{P}| + n.2^{|\mathbb{P}|})$. For a fixed set of processes, the space needed to check causal interferences is hence polynomial in the size of the input HMSC.

As for local non-interference, the hardness result can be proved by polynomial encoding of a regular language inclusion problem. Consider two regular languages L_1, L_2. Then one can design two HMSCs H_1, H_2 with initial nodes n_0^1, n_0^2 such that $\mathcal{L}(\mathcal{O}_p(H_i)) = L_i$, for $i \in \{1, 2\}$. Then using the MSC M'_c of Figure 4 (in Appendix), one can design a new HMSC H that contains all transitions and accepting nodes of H_1, H_2, with initial node n_0^2 and an additional transition $t_1 = (n_0, M'_c, n_0^1)$. The MSC M'_c contains one confidential event on some process P_c, followed by messages from P_c to all processes in $\mathbb{P}$. This way, any path of H that starts with transition t_1 generates an MSC in which p is black, and whose projection is in L_1. Other paths that do not start with t_1 generate MSCs from $\mathcal{F}_{H_2}$, and in particular MSCs in which p is white and whose projection on p is in L_2. Clearly, H is causally interferent with respect to p if and only if $L_1 \subseteq L_2$. □

Causal interference can be checked in $O(log(|H|) + |\mathbb{P}| + |H|.2^{|\mathbb{P}|})$. It is polynomial in space in the size of the HMSC, and exponential in the number of processes, but HMSC specifications are usually defined for small sets of processes. Also remark that reusing the construction of $H^{W,p}$, we can easily design an automaton recognizing $\mathcal{O}_{\setminus C}^{V,\circ}(H)$ as soon as $V = \Sigma_p$. Hence, $\mathcal{O}_{\setminus C}^{V,\circ}(H)$ is effectively regular if $V = \Sigma_p$.

VI. Declassification

Non-interference considers confidential information as secrets that should remain undisclosed along all runs of a system. This point of view is too strict to be of practical interest: In many cases, confidentiality of a secret action has a limited duration and secrets can be downgraded. Consider the following example: a user wants to buy an item online, and pays by sending his credit card information. Everything from this transaction between the online shop and the buyer (even if encryption is used) should remain secret. Within this setting, all payment steps should be considered confidential, and flow from these actions to observable events should be prevented. However, if a buyer uses a one time credit card (i.e. a virtual credit card number generated on request that can be used only once for a transaction), then all information on the card is valueless as soon as the payment is completed. Hence, after completing the transaction, learning that a payment occurred is harmless and the sequence of interactions implementing a secured online payment need not be kept secret. This declassification possibility was first proposed as *conditional interference* by [1] and later defined in [2] as intransitive interference. Intransitive non interference (INI) can be formulated as follows: any run of the system that contains a confidential action that is not declassified has an equivalent run from the observer's point of view. Usually, INI is defined using a pruning function that removes from a run all confidential actions that are not declassified, and compares observations of pruned and normal runs (see [7] for a definition of INI for transition systems).

From now on, we assume that the alphabet $\Sigma = C \uplus V \uplus N$ contains a particular subset $D \subseteq V \uplus N$ of declassification events. Intuitively, declassification events downgrade all their confidential causal predecessors.

Definition 22: Let M be an MSC. An event $e \in E_M$ is *classified* if it is a confidential event ($\alpha(e) \in C$), it has an observable successor $v \in V$ and it is not declassified before v, i.e. there exists no d such that $e \leq d \leq v$ and $\alpha(d) \in D$. We denote by $Clas(M)$ the set of classified events of M. The observation function $\mathcal{O}^V_{\setminus C,D}$ is defined by $\mathcal{O}^V_{\setminus C,D}(M) = \mathcal{O}^V(M \setminus Clas(M))$. An MSC M is *intransitively non-interferent* (INI) iff $\mathcal{L}(\mathcal{O}^V_{\setminus C,D}(M)) = \mathcal{L}(\mathcal{O}^V(M))$.

We can characterize INI in a single MSC M as a property depending on the causal order in M and on the sets of confidential, declassification, and observable events.

Proposition 23: An MSC M is intransitively non-interferent w.r.t. an alphabet $\Sigma = C \uplus V \uplus N$ and a set of declassification letters D iff for every pair of events $c \leq v$ such that $\alpha(c) \in C$ and $\alpha(v) \in V$, we have $(\downarrow(c) \cap \uparrow(v)) \cap \alpha^{-1}(D) \neq \emptyset$.

This proposition means that a declassification must occur between every confidential event and a causally related visible event. We now define observation functions for HMSCs and propose a definition of intransitive non interference for HMSCs. We define $\mathcal{O}^V_{II,D}(H) = \{\mathcal{O}^V(M) \mid M \text{ is not } INI\}$ and $\mathcal{O}^V_{INI,D}(H) = \{\mathcal{O}^V(M) \mid M \text{ is } INI\}$. We follow the definition of [7] to define INI for HMSCs. An HMSC is INI if for every intransitively interferent (II for short) MSC M in $\mathcal{F}_H$, there exists another MSC M' in $\mathcal{F}_H$ such that M' that is INI and such that $\mathcal{L}(\mathcal{O}^V(M')) = \mathcal{L}(\mathcal{O}^V(M))$.

Definition 24: An HMSC is intransitively non-interferent w.r.t. a declassification alphabet D if $\mathcal{L}(\mathcal{O}^V_{INI,D}(H)) = \mathcal{L}(\mathcal{O}^V(H))$.

Obviously, $\mathcal{O}^V_{INI,D}(H) \subseteq \mathcal{O}^V(H)$, so proving INI boils down to proving $\mathcal{L}(\mathcal{O}^V(H)) \subseteq \mathcal{L}(\mathcal{O}^V_{INI,D}(H))$. Note that all II MSCs are also interferent, and that checking non-interference amounts to checking INI with $D = \emptyset$. This remark extends to HMSCs: all intransitively interferent HMSCs are also causally interferent, and checking causal interference amount to checking INI with $D = \emptyset$. We then establish the following result:

Theorem 25: INI for HMSCs is undecidable. For a fixed set of processes, if $V \subseteq \Sigma_p$, then INI is PSPACE-complete.

We prove the decidability part of this theorem in three steps detailed below. We first show that INI can be decided for a sequence of MSCs without remembering the whole sequence. We then show that HMSCs can be designed to recognize respectively II MSCs of $\mathcal{F}_H$, and INI MSCs of $\mathcal{F}_H$. An immediate consequence is that $\mathcal{O}^V_{INI,D}(H)$ is effectively regular if $V \subseteq \Sigma_p$. A second consequence is that checking INI is PSPACE-complete. Let us first show that INI can be decided in a compositional way.

Proposition 26: Let M_1, M_2 be two MSCs. Then, $M_1 \circ M_2$ is INI if and only if M_1 and M_2 are INI, and for each pair of events $c \in M_1$, $v \in M_2$ such that $\alpha(c) \in C$, $\alpha(v) \in V$, and $c \leq_{1\circ 2} v$, there exists a process q, with

- $c \leq f$, where f is the maximal event on process q in M_1,
- $f' \leq v$, where f' is the minimal event on q in M_2,

and an event d such that $\alpha(d) \in D$, and $c \leq d \leq f$ or $f' \leq d \leq v$.

This proposition can be intuitively seen as a property of causal chains. A causal chain from c to v is a sequence of events $c \leq e_1 \leq \dots e_n \leq v$. We say that a chain from c to v is declassified if $\alpha(e_i) \in D$ for some $i \in 1..n$. Then an MSC is INI if for any pair (c, v) of confidential/visible events such that $c \leq v$ there exists at least one declassified causal chain from c to v. If so, the confidential event c **must be** declassified by the occurrence of some declassifying action **before** the execution of v occurs.

A causal chain from c to v in $M_1 \circ M_2$ can be decomposed into a chain from c to the maximal event f on a process q in M_1, a causal ordering from f to a minimal event f' located on process q in M_2 coming from the sequential composition of M_1 and M_2, and then a causal chain from the minimal event f' on q to v. However, one does not need to know precisely the contents of M_1 to decide whether $M_1 \circ M_2$ is INI. It suffices to remember for each process p the confidential events of M_1 that are not yet declassified and are predecessors of the maximal event executed by process p in M_1.

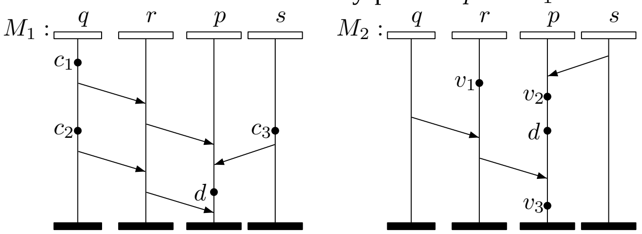

Fig. 3. An example of non INI sequence of MSCs

On the example depicted in Figure 3, MSC M_1 (left) contains three confidential actions c_1, c_2, c_3, and a declassification operation d. On the right, MSC M_2 contains three visible actions v_1, v_2, v_3, and a declassification operation d. All other events belong to $\alpha^{-1}(N)$. Both MSCs are INI, since no observation depends on a confidential action in M_1 or in M_2. However, in the concatenation $M_1 \circ M_2$, execution of v_1 or v_2 reveals the occurrence of c_2. Also note that c_1 is declassified by the first occurrence of d in M_1. This example is particularly interesting, as it shows that in order to abstract an arbitrarily long execution, it is not sufficient to remember a boolean value indicating whether there exists a not yet declassified action on a process, as two confidential events can be declassified via different ways. Indeed, some confidential actions could be declassified for a process while some others could not, even when located on the same process.

We can characterize II MSCs in a set $\mathcal{F}_H$ by remembering finite sets of shapes of causal chains. In order to define these shapes, let M be a MSC, let c be a confidential event in M. We define a function $cl(c, M) : \mathbb{P} \longrightarrow \{\bot, +, \top\}$ such that $cl(c, M)(p) = \bot$ if there exists no causal chain from c to an event located on p, $cl(c, M)(p) = +$ if there exists a causal chain from c to a maximal event f located on p, and $(\downarrow c \cap \uparrow f) \cap \alpha^{-1}(D) = \emptyset$, and $cl(c, M)(p) = \top$ otherwise. This function classifies processes according to the existence and classification degree (declassified or not) of causal chains between the confidential event c and the last event seen on each process. For a set $\mathbb{P}$ of processes, any such map $cl(c, M)$ can have at most $3^{|\mathbb{P}|}$ distinct values. Let $\mathcal{C}l = \{\bot, +, \top\}^{\mathbb{P}}$ denote the set of all maps.

By proposition 26, $M_1 \circ M_2$ is not INI if M_1 or M_2 is not INI, or if there exists $c \in M_1$ and $v \in M_2$ such that:

- there exists a process p such that $cl(c, M_1)(p) = +$, and an event f located on p in M_2, such that no causal chain from f to v is declassified.
- for every process q such that $cl(c, M_1)(q) = \top$ there exists no event $f \leq v$ located on q, and v is not located on q.

One can furthermore compute $cl(c, M_1 \circ M_2 \circ \cdots \circ M_k)(p)$ incrementally with finite memory. We have $cl(c, M_1 \circ M_2)(p) = \bot$ if $cl(c, M_1)(p) = \bot$, and if there exists no pair of events $e \leq f$ in M_2 with f is located of p, and $cl(c, M_1)(\phi(e)) \neq \bot$.

We have $cl(c, M_1 \circ M_2)(p) = +$ if $cl(c, M_1)(p) \in \{\bot, +\}$, there exists a process q such that $cl(c, M_1)(q) = +$, and a pair of events $e \leq f$ in M_2 such that e is minimal on q, f is maximal on process p, and furthermore, no causal chain from e to f is declassified, and for every process $q' \neq q$, if $cl(c, M_1)(q') = +$, then no declassified causal chain from an event on q' to f exists in M_2, if $cl(c, M_1)(q') = \top$ then no causal chain from an event on q' to f exists in M_2.

We have $cl(c, M_1 \circ M_2)(p) = \top$ if $cl(c, M_1)(p) = \top$, or

- there exist a process q such that $cl(c, M_1)(q) = +$ and a declassified chain from an event e located on process q to an event f located on process p, or
- there exist a process q such that $cl(c, M_1)(q) = \top$, and a causal chain from an event e located on process q to an event f located on process p.

Last, $cl(c, M_1 \circ M_2)(p) = \bot$ if $cl(c, M_1)(p) = \bot$ and M_2 does not contain a pair of events $e \leq f$ such that e is located on q with $cl(c, M_1)(q) \neq \bot$, and f is located on p.

Now, if M_1 contains two confidential events c_1, c_2 such that $cl(c_1, M_1) = cl(c_2, M_1)$, then $cl(c_1, M_1 \circ M_2) = cl(c_2, M_1 \circ M_2)$. It means that to detect interferences, one does not have to remember events, but only the shape of causal relations (existing, declassified or not) from confidential events to their successors on each process. There are at most $3^{|\mathbb{P}|}$ such distinct shapes in a MSC, so one can check INI along arbitrarily long sequences of MSCs with finite memory.

Proposition 27: Let H be an HMSC, with labeling alphabet Σ and set D of declassification letters. Then, one can build an HMSC H^{II} generating all II MSCs in $\mathcal{F}_H$ and an HMSC H^{INI} generating all INI MSCs in $\mathcal{F}_H$, with sizes at most $2.|H|.2^{3^{|\mathbb{P}|}}$

Proof Sketch. We build HMSC H^{II} as follows: a state (n, b, X) of H^{II} memorizes a node n of H, a boolean b indicating whether an interference has been detected, and a set $X = \{cl_1, \ldots cl_\ell)$, where each cl_i is a function from $\mathbb{P}$ to $\{\bot, +, \top\}$ that memorizes the shape of causal chains from a confidential event to maximal events on processes. H^{II} follows transitions of H, and updates cl_i's. For each new confidential event c occurring in a transition labeled by an MSC M, a new function $cl(c, M)$ is appended to the current state. As soon as an interference is detected, b is set to true. Accepting states of H^{II} are states where n is accepting in H, and b is true. H^{INI} can be designed with a similar construction where accepting states are states with n accepting and b false. □

We are now ready to prove Theorem 25:

Proof (of Theorem 25). Undecidability is easily obtained from undecidability of causal interference, and by setting $D = \emptyset$. Let us now consider the decidability part, with $V \subseteq \Sigma_p$. Following the proof of proposition 27, one can build an automaton $\mathcal{A}_p(H^{INI})$ of size at most $2.|H|.2^{3^{|\mathbb{P}|}}$ that recognizes $\mathcal{O}^V(H^{INI})$. One can easily prove that when $V \subseteq \Sigma_p$, we have $\mathcal{O}^V(H^{INI}) = \mathcal{O}^V_{INI,D}(H)$, and hence $\mathcal{L}(\mathcal{A}_p(H^{INI})) = \mathcal{L}(\mathcal{O}^V_{INI,D}(H))$, and $\mathcal{O}^V_{INI,D}(H)$ is effectively regular.

From proposition 17, we can build an automaton $\mathcal{A}_p(H)$ of size in $O(k.H)$, where k is the maximal number of events in an MSC of H, that recognizes $\mathcal{O}^V(H)$. Then it is sufficient to check whether $\mathcal{L}(\mathcal{A}_p(H)) \subseteq \mathcal{L}(\mathcal{A}_p(H^{INI}))$ to decide if H is intransitively interferent, which is again an inclusion problem that can be checked in space in $O(2.|H|.2^{3^{|\mathbb{P}|}})$. Hardness is proved by showing a polynomial reduction from a language inclusion problem to an INI problem with $D = \emptyset$. □

The declassification setting can be refined to consider selective declassification. Following the definition of [9], in addition to the declassification alphabet D, we define a map $h : D \rightarrow 2^C$, where $h(\alpha_d)$ defines the labels of confidential events that an action with label α_d declassifies. Definition 22 easily adapts to this setting, simply by requiring that a causal chain from a confidential event c to a visible event v is declassified by an event d such that $\alpha(c) \in h(\alpha(d))$. We then say that an event c is classified if it is a confidential event ($\alpha(c) \in C$), it has an observable successor v, and it is not declassified by one of the actions that can declassify it, that is, $\alpha(c) \notin h\left(\alpha(\downarrow (c) \cap \uparrow (v)) \cap D\right)$. INI with selective declassification (INISD) adapts the definitions of INI to consider declassification without changing observations. Like for standard declassification, we can build an HMSC that recognizes INISD MSCs of $\mathcal{F}_H$. The only change w.r.t. INI is that one has to remember in the HMSC construction the label of confidential events from which chains originate, yielding automata of sizes in $2.|H|.2^{|C|.3^{|\mathbb{P}|}}$. If $V \subseteq \Sigma_p$, then $\mathcal{O}^V_{II,D}$ and $\mathcal{O}^V_{INI,D}$ are effectively regular. We hence have:

Corollary 28: INISD is undecidable for HMSCs. For a fixed set of processes, it is PSPACE-complete when $V \subseteq \Sigma_p$.

VII. Related work and conclusion

Related work. Non-interference was seldom studied for scenario formalisms. A former work considers non-interference for Triggered Message Sequence Charts [18]. The interference property is defined in terms of comparison of ready sets (sets of actions that are fireable after a given sequence of actions w). However, this work mainly considers finite scenarios, and does not address decidability and complexity issues.

A first work considering non-interference for true concurrency models appears in [6]. The authors consider interference for elementary nets (*i.e.*, nets where firing rules allow places to contain at most one token). They characterize *causal places*, where firing a high-level transition causally precedes the firing of a low-level one and *conflict places*, where firing a high-level transition inhibits the firing of a low-level one. Reachability of causal or conflict places is shown equivalent to BNDC (Bisimulation-based NDC, the variant using bisimulation instead of language equality). In [7], the notion of intransitive non-interference from [2] is revisited for transition systems, and non-interference with downgraders is considered for elementary nets. A structural characterization is given in terms of reachable causal and conflict places. As in [6], causal and conflict places are characterized in terms of possible fireable sequences of transitions, hence considering the interleaving semantics of the net.

Darondeau *et al.* [8] study (B)NDC and INI for **unbounded** labeled Petri nets, and extend their results to selective declassification in [9]. The authors obtain decidability results of these properties for injectively labeled nets by a very clever exploitation of decidability/undecidability results for language inclusion. The characterization relies on sequences of transitions, and not on causal properties of nets.

A contrario, Baldan et al [10] emphasize the fact that characterizing BNDC in terms of structural conditions expressing causality or conflict between high and low-level transitions, is a way to provide efficient algorithms to check interference. They propose a definition of complete unfolding w.r.t. non-interference, and reduce BNDC for safe nets to checking that a complete unfolding is weak-conflict and weak causal place free. Weak causal places characterize dependencies and conflicts between high and low transitions. Their results show that one can identify interferences in concurrency models without relying on interleaving semantics. The characterization of BNDC via weak conflict and causal places holds only for safe nets, *i.e.*, for finite state systems.

Conclusion. We have proposed a partial order framework for information flow properties analysis, and shown that non-interference is undecidable. However, as soon as observations are effectively regular, information flow properties become decidable. This can be enforced for instance by considering observation performed by a single process in the system, which leads to the notions of local non-interference and its extensions with declassification, that are all decidable. These problems are PSPACE-complete, with procedures that never compute the interleaving semantics of the original HMSC.

A possible refinement of the landscape is to consider sufficient conditions for decidability of interference when several processes can observe the system. We would also like to extend the current definitions to use the full dicriminating power of partial orders, i.e. consider non-interference properties of the form $\mathcal{O}_1(H) \equiv \mathcal{O}_2(H)$, where $\equiv$ denotes isomorphism. For observations localized on a single process, the current results suffice. When observations are effectively regular, but contain events located on more than one process, showing isomorphism could require other tools such as graph grammars to compare observations, and decidability of interference problems may require additional conditions.

Another line of research is to consider security issues when an attacker can interact with the system in order to gain information (active interference), or when he can get information on the current configuration of the system (state-based interference). Extending definitions of information flows in HMSCs to quantify the amount of information disclosure by mean of measures (e.g. probability measure, average number of bits leaked per action,...) is also a challenging task.

References

[1] J. Goguen and J. Meseguer, “Security policies and security models,” in *Proc. of IEEE Symposium on Security and Privacy*, 1982, pp. 11–20.

[2] J. Rushby, “Noninterference, transitivity, and channel-control security policies,” SRI International, Tech. Rep. CSL-92-02, 1992.

[3] H. Mantel, “Possibilistic definitions of security - an assembly kit,” in *Proc. of the 13th IEEE Computer Security Foundations Workshop, (CSFW'00)*, 2000, pp. 185–199.

[4] ——, “Information flow control and applications - bridging a gap,” in *Proc. of FME 2001*, ser. LNCS, vol. 2021, 2001, pp. 153–172.

[5] R. Focardi and R. Gorrieri, “Classification of security properties (part i: Information flow),” in *FOSAD 2000*, 2000, pp. 331–396.

[6] N. Busi and R. Gorrieri, “Structural non-interference in elementary and trace nets,” *Mathematical Structures in Computer Science*, vol. 19, no. 6, pp. 1065–1090, 2009.

[7] R. Gorrieri and M. Vernali, “On intransitive non-interference in some models of concurrency,” in *FOSAD VI Tutorial Lectures*, ser. LNCS, vol. 6858, 2011, pp. 125–151.

[8] E. Best, P. Darondeau, and R. Gorrieri, “On the decidability of non interference over unbounded Petri nets,” in *Proc. of SecCo*, ser. EPTCS, vol. 51, 2010, pp. 16–33.

[9] E. Best and P. Darondeau, “Deciding selective declassification of Petri nets,” in *Proc. of POST 2012*, ser. LNCS, vol. 7215, 2012, pp. 290–308.

[10] P. Baldan and A. Carraro, “Non-interference by unfolding,” in *Petri Nets*, ser. LNCS, vol. 8489, 2014, pp. 190–209.

[11] D. D'Souza, R. Holla, K. Raghavendra, and B. Sprick, “Model-checking trace-based information flow properties,” *Journal of Computer Security*, vol. 19, no. 1, pp. 101–138, 2011.

[12] ITU-T, “Z.120 : Message sequence charts (MSC),” International Telecommunication Union, Tech. Rep., 2011.

[13] B. Bérard and J. Mullins, “Verification of information flow properties under rational observation,” in *Proc. of AVOCS 2014, ECEASST 70*, 2014.

[14] A. Muscholl and D. Peled, “Message sequence graphs and decision problems on Mazurkiewicz traces,” in *MFCS*, 1999, pp. 81–91.

[15] B. Caillaud, P. Darondeau, L. Hélouët, and G. Lesventes, “HMSCs en tant que spécifications partielles et leurs complétions dans les réseaux de Petri,” INRIA, RR-3970, 2000.

[16] B. Genest, L. Hélouët, and A. Muscholl, “High-level message sequence charts and projections,” in *CONCUR*, ser. LNCS, vol. 2761, 2003, pp. 308–322.

[17] A. Muscholl and D. Peled, “Analyzing message sequence charts,” in *SAM 2000, 2nd Workshop on SDL and MSC*, 2000, pp. 3–17.

[18] A. Ray, B. Sengupta, and R. Cleaveland, “Secure requirements elicitation through triggered message sequence charts,” in *ICDCIT 2004*, ser. LNCS, vol. 3347, 2004, pp. 273–282.

APPENDIX

This appendix details missing proofs for propositions in the text. Proofs for propositions 15, 23 and 26 are straightforward consequences of definitions 14 and 22, and are not provided.

A. Proof of Proposition 10

Proposition 10. Let H be a HMSC. The inclusion problem $\sqsubseteq_{\mathcal{O}_1,\mathcal{O}_2}(H)$ is undecidable in general, even if $\mathcal{O}_1$ **or** $\mathcal{O}_2$ is an effective regular observation function. It is *decidable* if $\mathcal{O}_1$ **and** $\mathcal{O}_2$ are both effective regular functions.

Proof. The proof is a reduction from the language inclusion problem for HMSCs (which is known undecidable). Let $H_1 = (N_1, \rightarrow_1, \mathcal{M}_1, n_{0,1}, F_1)$ and $H_2 = (N_2, \rightarrow_2, \mathcal{M}_2, n_{0,2}, F_2)$ be two HMSCs, defined over an alphabet of visible actions V, and with at least two processes. We design an inclusion problem such that $\sqsubseteq_{\mathcal{O}_1,\mathcal{O}_2}(H)$ iff $\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$.

Let $P_c \notin \mathbb{P}$ be a new process and c a new confidential action. We define M_c as the MSC containing the single atomic action c on process P_c, as illustrated on Figure 4 bottom left. A new HMSC $H = (N_1 \uplus N_2, \rightarrow, \mathcal{M}, n_{0,2}, F_1 \uplus F_2)$ is defined over alphabet $\Sigma' = V \cup C$, where $C = \{P_c(c)\}$, as follows: $\mathcal{M} = \mathcal{M}_1 \uplus \mathcal{M}_2 \uplus \{M_c\}$ and $\rightarrow = \rightarrow_1 \uplus \rightarrow_2 \uplus \{(n_{0,2}, M_c, n_{0,1})\}$, as illustrated on the upper part of Figure 4.

Choosing $\mathcal{O}_1 = \mathcal{O}^{V,\circ}$ and $\mathcal{O}_2 = \mathcal{O}^{V,\circ}_{\backslash C}$, we clearly have $\mathcal{O}_2(H) = \mathcal{L}(H_2)$ and $\mathcal{O}_1(H) = \mathcal{L}(H_1) \cup \mathcal{L}(H_2)$. Thus $\sqsubseteq_{\mathcal{O}_1,\mathcal{O}_2}$ (H) if and only if $\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$, which concludes the proof.

Note that undecidability is not due to a particular choice of observation function: a similar proof is obtained for $\mathcal{O}_1 = \mathcal{O}^{V,\circ}$ or $\mathcal{O}_1 = \mathcal{O}^{V,\bullet}$ and $\mathcal{O}_2 = \mathcal{O}^{V,\bullet}_{\backslash C}$, by replacing M_c by an MSC M'_c in which process P_c sends a message to all other processes after performing action c, as depicted on the bottom right of Figure 4.

Similarly, if $\mathcal{O}_1$ is an effective regular function, that is if $\mathcal{O}_1(H)$ is a regular language L, the inclusion problem is brought back to testing whether $L \subseteq \mathcal{L}(H_1)$, which is also known to be an undecidable property.

The decidability result of the theorem is an immediate consequence of decidability for regular languages inclusion. □

B. Proof of Theorem 12

Theorem 12. Let $\mathcal{O}$ be an observation function on alphabet $\Sigma = C \uplus V \uplus N$ and let $\mathcal{H}$ be a set of HMSCs. One cannot decide in general if $\mathcal{O}$ is effectively regular for $\mathcal{H}$.

Proof. It was proved in [15] that one cannot decide whether $\mathcal{L}(H)$ is regular, *i.e.*, if it can be recognized by a finite automaton. Undecidability comes from the fact that HMSCs can be used to encode rational traces, for which regularity is undecidable. Let us set $\Sigma = V$. Then, checking whether $\mathcal{O} = \mathcal{O}^{V,\circ}$ is effectively regular for a partition of alphabet Σ amounts to checking regularity of $\mathcal{L}(H)$, for every $H \in \mathcal{H}$. However, this is already an undecidable problem for a single HMSC, obtained if we set $\mathcal{H} = \{H\}$. □

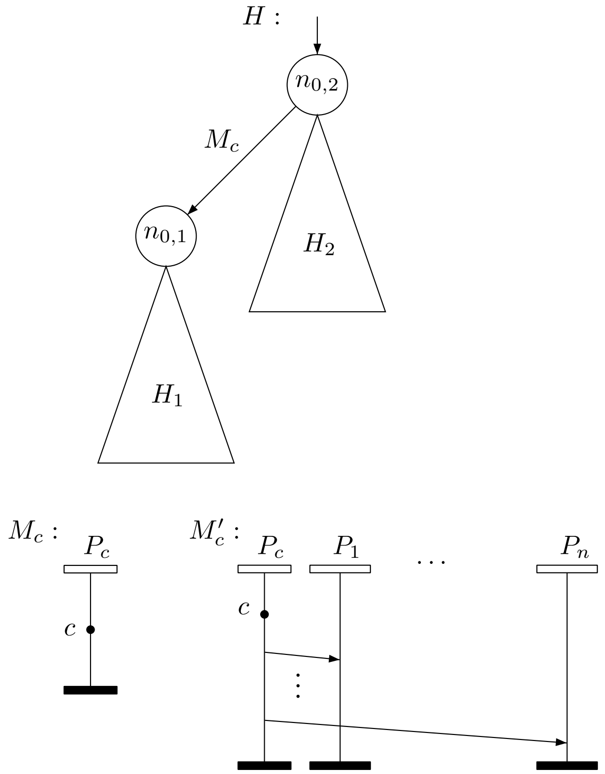

Fig. 4. Non-interference in HMSCs as a language problem

C. Proof of Proposition 13

Proposition 13. Let M be an MSC with labeling alphabet $\Sigma = C \uplus V \uplus N$ and set of events E. Then, M is *interferent* if and only if there are two events e, f such that $\alpha(e) \in C$, $\alpha(f) \in V$, and $e \leq f$.

Proof. We prove this lemma by showing the two directions of the implication.

First, let us suppose that there exists no pair of events e, f in M such that $\alpha(e) \in C$, $\alpha(f) \in V$. If there is no event $e \in E$ such that in $\alpha(e) \in C$, then $M \backslash \downarrow (\alpha^{-1}(C)) = M$, which gives the required equality. If $\alpha^{-1}(C) \neq \emptyset$, then for each $e \in \alpha^{-1}(C)$, $\downarrow (e) \cap \alpha^{-1}(V) = \emptyset$. In this case, $M \backslash \downarrow (\alpha^{-1}(C)) = M \setminus \alpha^{-1}(C)$. This means that $\mathcal{O}^V(M) = \mathcal{O}^V_{\backslash C}(M)$, which yields the result.

$ii)$ Let us now prove the converse direction. Suppose that there exists a pair of events $e \leq f$ such that e is a confidential event, and f is a visible one. Then, all linear extensions of M are of the form $u = v.e.v'.f.v''$. For each of them, the largest common prefix between $\pi_V(u)$ and any word in $\pi_V(M \backslash \downarrow (\alpha^{-1}(C)))$ is a prefix of $\pi_V(v)$. This implies that $\mathcal{O}^V_{\backslash C}(M) \neq \mathcal{O}^V(M)$. □

D. Proof of Proposition 17

Proposition 17. $\mathcal{O}^p$ is effectively regular, and if $V = \Sigma_p$, then $\mathcal{O}^{V,\bullet}_{\backslash C}$ and $\mathcal{O}^{V,\circ}$ are effectively regular.

Proof Let us show that for any HMSC $H = (N, \rightarrow, \mathcal{M}, n_0, F)$ one can effectively build a finite state automaton $\mathcal{A}_p(H)$ recognizing $\mathcal{L}(\mathcal{O}^{V,\circ}(H))$ or equivalently $\mathcal{O}^p$.

Let k be the maximal size of a projection of a MSC in $\mathcal{M}$. The automaton $\mathcal{A}_p(H)$ is defined by $\mathcal{A}_p = (N \times \{0, \dots, k-1\}, \delta, (n_0, 0), F \times \{0\})$. Let (n, M, n') be a transition in H. The observation $\mathcal{L}(\mathcal{O}^p(M)$ is a possibly empty word of Σ_p. If $\pi_p(\mathcal{L}(M)) = \varepsilon$, then δ contains the transition $((n, 0), \varepsilon, (n', 0))$. If $\pi_p(M) = a_1 \dots a_q$ (with $q \leq k$), then δ contains the transitions $((n, i-1), a_i, (n, i))$ for each $i \in \{1, .., q-1\}$, and $((n, q-1), a_q, (n', 0))$.

An easy induction shows that for every path $(n_0, M_1, n_1) \dots (n_{\ell-1}, M_\ell, n_\ell)$, such that the projection of each M_i on p is a word $w_i = a_{i,1} \dots a_{i,q_i}$ there exists a path $(n_0, 0) \xrightarrow{a_{1,1}} (n_0, 1) \xrightarrow{a_{1,2}} \dots \xrightarrow{a_{\ell,q_\ell}} (n_\ell, 0)$, and conversely. Furthermore, if n_ℓ is an accepting state of H, then $(n_\ell, 0)$ is an accepting state of $\mathcal{A}_p$. Hence, $\mathcal{A}_p$ recognizes $\mathcal{L}(\mathcal{O}^p(H))$. The size of $\mathcal{A}_p(H)$ is in $O(|N|.k)$.

Let us now show that one can design an automaton that recognizes $\mathcal{L}(\mathcal{O}^{V,\bullet}_{\setminus C}(H)$ for any HMSC H. Let us first recall the definition of $\mathcal{O}^{V,\bullet}_{\setminus C}(H)$. We have $\mathcal{O}^{V,\bullet}_{\setminus C}(H) = \{\mathcal{O}^V(M_1 \circ \dots \circ M_k) \mid M_1 \circ \dots \circ M_k \in \mathcal{F}_H \wedge \forall i, \alpha(E_i) \cap C = \emptyset\}$. Let us now design a new HMSC $H_{\setminus C} = (N, \rightarrow_{\setminus C}, \mathcal{M}, n_0, F)$ such that $(n, M, n') \in \rightarrow_{\setminus C}$ iff $(n, M, n') \in \rightarrow$ and $\alpha(E_M) \cap C = \emptyset$. Clearly, $\mathcal{F}_{H_{\setminus C}}$ is the set of MSCs generated by H that do not contain actions from C, and $H_{\setminus C}$ is also an HMSC. We have $\mathcal{O}^p(H_{\setminus C}) = \mathcal{O}^{V,\bullet}_{\setminus C}(H)$, and hence we can apply the technique above to design an automaton $\mathcal{A}'_p(H) = \mathcal{A}_p(H_{\setminus C})$ of size in $O(|N|.k)$ that recognizes $\mathcal{L}(\mathcal{O}^{V,\bullet}_{\setminus C}(H))$. Hence $\mathcal{O}^p$ and $\mathcal{O}^{V,\bullet}_{\setminus C}$ are effectively regular. □

E. Proof of Corollary 18

Corollary 18. The problem of deciding local interference of an HMSC H with respect to a given process $p \in \mathbb{P}$ is PSPACE-complete.

Proof From proposition 17, for any HMSC H and any process p, we can design an automaton $\mathcal{A}_p(H)$ that recognizes $\mathcal{L}(\mathcal{O}^p((H))$, and an automaton $\mathcal{A}'_p(H)$ that recognizes $\mathcal{L}(\mathcal{O}^{V,\bullet}_{\setminus C}(H))$.

Note that these automata are of size linear in the size of H. One can notice that $\mathcal{L}(\mathcal{A}'_p(H)) \subseteq \mathcal{L}(\mathcal{A}_p(H))$. So, checking local non-interference of an HMSC H amounts to a single inclusion problem $\sqsubseteq_{\mathcal{O}^p, \mathcal{O}^{V,\bullet}_{\setminus C}}$ for HMSC H, i.e checking that $\mathcal{L}(\mathcal{A}_p(H)) \subseteq \mathcal{L}(\mathcal{A}'_p(H))$. Language inclusion for finite automata is a well-known PSPACE-complete problem, hence checking local non-interference is in PSPACE.

For the hardness part, let $\mathcal{A} = (Q_A, \delta_A, q_{0A}, F_A)$ and $\mathcal{B} = (Q_B, \delta_B, q_{0B}, F_B)$ be two automata over alphabet Σ, with disjoint set of states. Similarly to Figure 4, we design a HMSC $H = (Q_A \uplus Q_B, \rightarrow, q_{0B}, F_A \uplus F_B)$ over a set of processes $\{p_1, p_2, P_c\}$ and alphabet $\Sigma \cup \{c\}$, with $V = \Sigma$ and $C = \{c\}$, such that $\rightarrow$ contains:

- a transition (q_{0B}, M_h, q_{0A}) in which M_c is an MSC with a single atomic confidential action located on process P_c (like in Figure 4),
- for each $(q, a, q') \in \delta_A \cup \delta_B$, a transition (q, M_a, q') where M_a is a MSC containing a single message m_a from p_2 to p_1.

Then the language inclusion problem $\mathcal{L}(A) \subseteq \mathcal{L}(B)$ can be reduced in polynomial time to local non-interference of H with respect to process p_1. Hence, local non-interference is PSPACE-complete. □

F. Proof of Proposition 21

Proposition 21. Let H be an HMSC, $p \in \mathbb{P}$, and $\Sigma = C \uplus V \uplus N$. Then, one can build:

- an HMSC $H^{B,p}$ that recognizes MSCs from $\mathcal{F}_H$ after which p is a black process.
- an HMSC $H^{W,p}$ that recognizes MSCs from $\mathcal{F}_H$ after which p is a white process.

of size in $O(|H|.2^{|\mathbb{P}|})$.

Proof. We build $H^{B,p} = (N^{B,p}, \rightarrow^{B,p}, \mathcal{M}, n_0^{B,p}, F^{B,p})$ as follows:

- $N^{B,p} \subseteq N \times 2^{\mathbb{P}}$ is a set of nodes. In a pair (n, P), n denotes a node of H, and P a subset of black processes. We set $n_0^{BW} = (n_0, \emptyset)$.
- the set of transitions and nodes of $H^{B,p}$ is built inductively as follows: from a node (n, P), if there exists a transition (n, M, n') in H, we add (n', P') to $N^{B,p}$, with $P' = P \cup \{p \in \mathbb{P} \mid \exists e \leq_M f \wedge \phi(f) = p \wedge \phi(e) \in P\} \cup \{p \in \mathbb{P} \mid \exists e \leq f, \alpha(e) \in C \wedge \phi(f) = p\}$, and we add transition $\big((n, P), M, (n', P')\big)$ to $\rightarrow^{B,p}$
- $F^{B,p} = F \times \{P \in 2^{\mathbb{P}} \mid p \in P\}$ is the set of accepting nodes. A path of $H^{B,p}$ is accepting if it ends after recognizing an MSC $M \in \mathcal{F}_H$ such that p is black after M.

Building $H^{W,p} = (N^{W,p}, \rightarrow^{W,p}, \mathcal{M}, n_0^{W,p}, F^{W,p})$ can be done in a similar way, but setting $F^{W,p} = F \times \{P \in 2^{\mathbb{P}} \mid p \notin P\}$.

The status of a process is built progressively along transitions in a path. Following proposition 15, the process part of a node in $H^{B,p}$ or $H^{W,p}$ faithfully encodes the status of a process in the MSCs generated by sequences of transitions ending in this node. Hence, $H^{B,p}$ (resp. $H^{W,p}$) recognize MSCs of $\mathcal{F}_H$ after which p is black (resp. white).

As the nodes of these HMSCs belong to $N \times 2^{\mathbb{P}}$, the size of $H^{B,p}$ or $H^{W,p}$ is in $O(|H|.2^{|\mathbb{P}|})$. □

G. Proof of Proposition 27

Proposition 27. Let H be an HMSC, Σ an alphabet and D be a set of declassification letters. Then, one can build

- an HMSC H^{II} that generates the set of II MSCs in $\mathcal{F}_H$.
- an HMSC H^{INI} that generates the set of INI MSCs in $\mathcal{F}_H$.

that are of sizes at most $2.|H|.2^{3^{|\mathbb{P}|}}$

Proof. We first show how $H^{II} = (N^{II}, \rightarrow^{II}, \mathcal{M}, n_0^{II}, F^{II})$ is computed, then we show that H^{II} recognizes intransitively interferent MSC generated by H. We first define the following functions. A map $cl : \mathbb{P} \rightarrow \{+, \perp, \top\}$ represents existing causal dependencies from a confidential event to maximal event of processes, plus gives information on whether a causal chain ending on a process declassifies this confidential event. We denote by $CL(M)$ the set of functions that are computed starting from all confidential events. Note that if M contains no confidential event, then $CL(M) = \emptyset$. Given two MSCs M_1, M_2, we have seen that $M_1 \circ M_2$ is intransitively interferent if M_1 is II, or M_2 is II, or there exists $cl \in CL(M_1)$ such that $cl(p) = +$ and M_2 contains a chain from an event located on process to an observable event v such that there exists no chain from an event on process q to v with $cl(q) = \top$. Hence, knowing if M_1 is *II* or not, and $CL(M_1)$, one can decide whether $M_1 \circ M_2$ is *II*. We denote by $\mathit{II}(CL, M)$ the predicate that is true when a set of maps CL representing causal chains and declassification in an MSC M' allows to prove that $M' \circ M$ contains an intransitive interference.

The crux is then to be able to maintain $CL(M_1 \circ \cdots \circ M_k)$ and the II information along path of H. For a given map cl and an MSC M, we define the map $Update(cl, M)$ as follows:

We have $Update(cl, M)(p) = \bot$ if $cl(p) = \bot$, and if there exists no pair of events $e \leq f$ in M_2 with f is located of p, and $cl(\phi(e)) \neq \bot$.

We have $Update(cl, M)(p) = +$ if $cl(p) \in \{\bot, +\}$, and there exists a process q such that $cl(q) = +$, and pair of events $e \leq f$ in M_2 such that e is minimal on q, f is maximal on process p, and furthermore, no causal chain from e to f is declassified, and for every process $q' \neq q$, if $cl(q') = +$, then no declassified causal chain from an event on q' to f exists in M_2, if $cl(q') = \top$ then no causal chain from an event on q' to f exists in M_2.

We have $Update(cl, M)(p) = \top$ if

- $cl(p) = \top$, or
- there exist a process q such that $cl(q) = +$ and a declassified chain from an event e located on process q to an event f located on process p, or
- there exist a process q such that $cl(q) = \top$, and a causal chain from an event e located on process q to an event f located on process p.

The map updating function extends to sets of maps the obvious way : $Update(X, M) = \bigcup\limits_{cl \in X} Update(cl, M)$.

We are now ready to define $H^{\mathit{II}} = (N^{\mathit{II}}, \rightarrow^{\mathit{II}}, \mathcal{M}, n_0^{\mathit{II}}, F^{\mathit{II}})$. We have:

- $N^{\mathit{II}} \subseteq N \times \{tt, \mathit{ff}\} \times 2^{\mathcal{C}l}$ is a set of nodes that are reachable from n_0^{II}. Each node of N^{II} is hence a triple of the form (n, b, X), where n is a node of H, b is a boolean that indicates if *II* has already been discovered, and X is a set of maps depicting (declassified) causal chains from confidential events in the sequence $M_1 \circ \ldots M_k$ read so far along transitions of H and ending at node n. We set $n_0^{\mathit{II}} = (n_0, \mathit{ff}, \emptyset)$.
- We define the transitions relation as follows. We have $((n, b, X), M, (n', b', X')) \in \rightarrow^{\mathit{II}}$ iff
  - $(n, M, n') \in \rightarrow$ (the transition exists in H),
  - $b' = b \vee \bigvee\limits_{cl \in X} \mathit{II}(cl, M) \wedge M$ is *II*, that is if *II* was detected before, then the concatenated MSCs remain *II*, and otherwise become *II* if M is *II*, or one of the maps depicting chains starting from a confidential events in the formerly assembled MSC $M_1 \circ \cdots \circ M_k$ witnesses an *II* in $M_1 \circ \cdots \circ M_k \circ M$.
  - $F^{\mathit{II}} = F \times \{tt\} \times 2^{\mathcal{C}l}$
  - $X' = Update(X, M) \cup CL(M)$. The representation of chains originating from confidential events is updated to consider chains of M and their declassifications, and new observable events may occur in M, starting new chains and potential new witnesses for *II* MSCs.

Obviously, all MSCs generated by H^{II} belong to $\mathcal{F}_H$, as $\rightarrow^{\mathit{II}}$ always agrees with $\rightarrow$. Furthermore, due to compositionality of cl computation, updating of a chain cl can be done incrementally while concatenating MSCs without remembering the whole sequence. Now, it suffices to remember once the shape of causal chains from observables actions to maximal events on processes (the maps cl) to detect II. One needs not differentiate similar occurrences of maps computed for chains originating from distinct observable events. Hence, updating of sets of causal chains representation suffices to represent all classified chains in a sequence of MSCs recognizes between n_0 and the current node, and hence to detect all occurrences of intransitive interferences. we can conclude that all MSCs recognized by H^{II} contain an intransitive interference.

The HMSC $H^{\mathit{INI}} = (N^{\mathit{INI}}, \rightarrow^{\mathit{INI}}, \mathcal{M}, n_0^{\mathit{INI}}, F^{\mathit{INI}})$ can be built with the same nodes and transition functions, but with final sates $F^{\mathit{INI}} = F \times \{\mathit{ff}\} \times 2^{\mathcal{C}l}$. The sizes of H^{II} and H^{INI} are at most $2.|H|.2^{3^{|\mathbb{P}|}}$.

□