Causal Message Sequence Charts*

Thomas Gazagnaire[1], Blaise Genest[2], Loïc Hélouët[3], P.S. Thiagarajan[4], Shaofa Yang[3]

[1] IRISA/ENS Cachan, Campus de Beaulieu, 35042 Rennes Cedex, France
[2] IRISA/CNRS, Campus de Beaulieu, 35042 Rennes Cedex, France
[3] IRISA/INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France
[4] School of computing, NUS, Singapore

Abstract. Scenario languages based on Message Sequence Charts (MSCs) and related notations have been widely studied in the last decade [14, 13, 2, 9, 5, 12, 8]. The high expressive power of scenarios renders many basic problems concerning these languages undecidable. The most expressive class for which several problems are known to be decidable is one which possesses a behavioral property called "existentially bounded". However, scenarios outside that class are frequently exhibited by asynchronous distributed systems such as sliding window protocols. We propose here an extension of MSCs called causal Message Sequence Charts that preserves decidability without requiring existential bounds. Interestingly, it can also model scenarios from sliding window protocols. We establish the expressive power and complexity of decision procedures for various subclasses of causal Message Sequence Charts.

1 Introduction

Scenario languages based on Message Sequence Charts (MSCs) have met considerable interest in the last ten years. The attractiveness of this notation can be explained by two major characteristics. Firstly, from the engineering point of view, MSCs have a simple and appealing graphical representation based on just a few concepts: processes, messages and internal actions. Secondly, from a mathematical standpoint, scenario languages admit an elegant formalization: they can be defined as languages generated by finite state automata over an alphabet of MSCs. These automata are usually called High-level Message Sequence Charts (HMSCs)[10].

An MSC is a restricted kind of labelled partial order and an HMSC is a generator of a (usually infinite) set of MSCs, that is, a language of MSCs. For example, the MSC M shown in Figure 2 is a member of the MSC language generated by the HMSC of Figure 1 while the MSC N shown in Figure 2 is not.

HMSCs are very expressive and hence a number of basic problems associated with them can not be solved effectively. For instance, it is undecidable whether two HMSCs generate the same collection of MSCs [14], or whether an HMSC generates a regular MSC language (an MSC language is regular if the collection of

* Work supported by the CASDS Associated Team and the ANR projects DOTS.

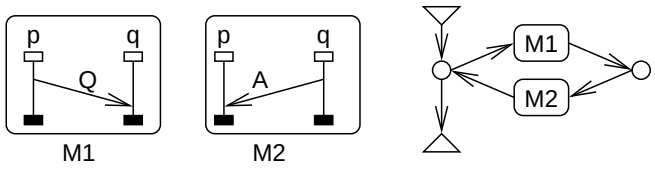

Fig. 1. An HMSC over two MSCs

all the linearizations of all the MSCs in the language is a regular string language in the usual sense). Consequently, subclasses of HMSCs have been identified [13, 2, 5] and studied.

On the other hand, a basic limitation of HMSCs is that their MSC languages are finitely generated. More precisely, each MSC in the language can be defined as the sequential composition of elements chosen from a fixed finite set of MSCs [12]. However, the behaviours of many protocols constitute MSC languages that are *not* finitely generated. This occurs for example with scenarios generated by the alternating bit protocol. Such protocols can induce a collection of braids like N in Figure 2 which can not be finitely generated.

One way to handle this is to work with so called safe (realizable) *Compositional* HMSCs (CHMCs, for short) in which message emissions and receptions are decoupled in individual MSCs but matched up at the time of composition, so as to yield a (complete) MSC. CHMSCs are however notationally awkward and do not possess the visual appeal of HMSCs. Furthermore, several positive results on HMSCs rely on a decomposition of MSCs in atoms (the smallest non-trivial MSCs) [9, 12, 5], which does not apply for CHMSCs, and results in a higher complexity [6]. It is also worth noting that without the restriction to safety (realizability), compositional HMSC languages embed the full expressive power of communicating automata [3] and consequently inherit all their undecidable results.

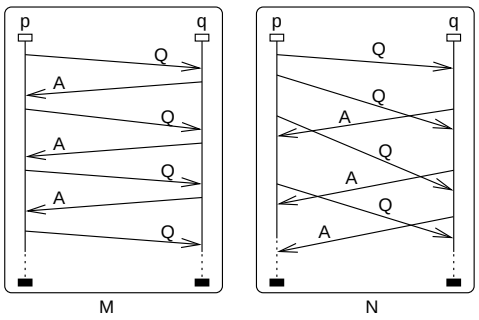

Fig. 2. Two MSCs M and N

This paper proposes another approach to extend HMSCs in a tractable manner. The key feature is to allow the events belonging to a lifeline to be partially ordered. More specifically, we extend the notion of an MSC to that of *causal* MSC in which the events belonging to each lifeline(process), instead of being linearly ordered, are allowed to be partially ordered. To gain modelling power, we do not impose any serious restrictions on the nature of this partial order. Secondly, we assume a suitable Mazurkiewicz trace alphabet for each lifeline and use this to define a composition operation for causal MSCs. This leads to the notion of causal HMSCs which generate tractable languages of causal MSCs.

A causal HMSC is *a priori* not existentially bounded in the sense defined in [6]. Informally, this property of an MSC language means that there is a uniform upper bound K such that for every MSC in the language *there exists* an execution along which—from start to finish—all FIFO channels remain K-bounded. Since this property fails, in general, for causal MSC languages, the main method used to solve decidability for safe CMSCs [6] is not applicable. Instead, to characterize regularity and decidability of certain subclasses of causal HMSCs, we need to generalize the method of [13] and of [5] in a non-trivial way.

In the next section we introduce causal MSCs and causal HMSCs. We also define the means for associating an ordinary MSC language with a causal HMSC. In the subsequent section we develop the basic theory of causal HMSCs. In section 4, we identify the property called "bounded-window", an important ingredient of the "braid"-like MSC languages generated by many protocols. Basically, this property bounds the number of messages a process p can send to a process q without waiting for an acknowledge to be received. We then show one can decide if a causal HMSC generates a bounded window language. In section 5 we compare the expressive power of languages based on causal HMSCs with other known HMSC-based language classes. Proofs are omitted due to lack of space, but can be found in the full version of the paper available at:
`http://www.irisa.fr/distribcom/Personal_Pages/gazagnaire/full.pdf` .

2 MSCs, causal MSCs and causal HMSCs

Through the rest of the paper, we fix a finite nonempty set $\mathcal{P}$ of process names with $|\mathcal{P}| > 1$, a finite nonempty set Msg of message names and Act, a finite nonempty set of internal action names. We define the alphabets $\Sigma_! = \{p!q(m) \mid p, q \in \mathcal{P}, p \neq q, m \in Msg\}$, $\Sigma_? = \{p?q(m) \mid p, q \in \mathcal{P}, p \neq q, m \in Msg\}$, and $\Sigma_{act} = \{p(a) \mid p \in \mathcal{P}, a \in Act\}$. The letter $p!q(m)$ means the sending of message m from p to q; $p?q(m)$ the reception of message m at p from q; and $p(a)$ the execution of internal action a by process p. Let $\Sigma = \Sigma_! \cup \Sigma_? \cup \Sigma_{act}$. We define the *location* of a letter α in Σ, denoted $loc(\alpha)$, by $loc(p!q(m)) = p = loc(p?q(m)) = loc(p(a))$. For each process p in $\mathcal{P}$, we set $\Sigma_p = \{\alpha \in \Sigma \mid loc(\alpha) = p\}$.

Definition 1. *A* causal MSC *over* $(\mathcal{P}, \Sigma)$ *is a structure* $B = (E, \lambda, \{\sqsubseteq_p\}_{p \in \mathcal{P}}, \ll)$, *where:*

- E *is a finite nonempty set. Members of* E *are called* events.

- $\lambda : E \to \Sigma$ *is a labelling function. This labelling function allows for a partition of* E *into three subsets,* $E_! = \lambda^{-1}(\Sigma_!)$,$E_? = \lambda^{-1}(\Sigma_?)$ *and* $E_{act} = \lambda^{-1}(\Sigma_{act})$.
- *For each process* p *in* $\mathcal{P}$, $\sqsubseteq_p \subseteq E_p \times E_p$ *is a* partial *order on* E_p, *where* $E_p = \{e \in E \mid loc(\lambda(e)) = p\}$.
- $\ll \subseteq E_! \times E_?$ *is a bijection identifying message sending-reception pairs. For each* $(e, e') \in \ll$, $\lambda(e) = p!q(m)$ *iff* $\lambda(e') = q?p(m)$.
- *The transitive closure of the relation* $(\bigcup_{p\in\mathcal{P}} \sqsubseteq_p) \cup \ll$, *denoted* $\leq$, *is a partial order.*

For convenience, we will often drop the subscript $p \in \mathcal{P}$ when it is clear from the context. Let $B = (E, \lambda, \{\sqsubseteq_p\}_{p\in\mathcal{P}}, \ll)$ be a causal MSC. If for all p, $\sqsubseteq_p$ is a total order over E_p, then B is called an *MSC*. We define $Alph(B) = \{\lambda(e) \mid e \in E\}$, and $Alph_p(B) = Alph(B) \cap \Sigma_p$ for each $p \in \mathcal{P}$. A *linearization* of B is a word $a_1 a_2 \ldots a_\ell$ over Σ such that $E = \{e_1, \ldots, e_\ell\}$ with $\lambda(e_i) = a_i$ for each i; and $e_i \leq e_j$ implies $i \leq j$ for any i, j. We let $Lin(B)$ denote the set of linearizations of B. $Lin(B)$ is not empty since $\leq$ is acyclic. For a set $\mathcal{B}$ of causal MSCs, and for $B \in \mathcal{B}$, we denote by $|B|$ the number of events of B and by $|\mathcal{B}|$ the size of $\mathcal{B}$, which is equal to $\sum_{B\in\mathcal{B}} |B|$.

Causal MSCs can be represented graphically: each process is represented as a vertical line (also called lifeline), that symbolizes time evolving from top to bottom. Along a lifeline, events are partially ordered by $\sqsubseteq_p$. The left part of Figure 3 depicts a causal MSC M. In M, events located on process q are ordered and events located on process p are unordered. This is symbolized by a box attached to instance name p, containing two unordered events. For instance, $p!q(Q).q!p(A).q?p(Q).p?q(A)$ and $q!p(A).p?q(A).p!q(Q).q?p(Q)$ are linearizations of M.

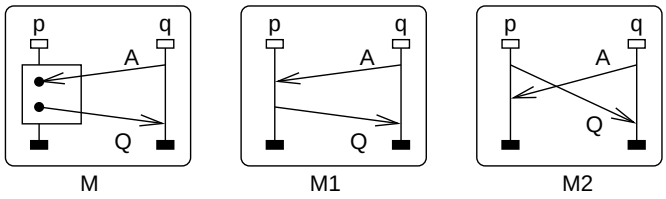

Fig. 3. A causal MSCs M and its two visual extensions M_1, M_2.

A *visual extension* of a causal MSC $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ is an MSC $(E, \lambda, \{\leq_p\}, \ll)$ (i.e. $\leq_p$ is a total order) such that for each $p \in \mathcal{P}$, $\sqsubseteq_p \subseteq \leq_p$. We will denote by $Vis(B)$ the set of all visual extensions of the causal MSC B. For example, Figure 3 shows the visual extensions $Vis(M) = \{M_1, M_2\}$ of causal MSC M.

We next need to define a composition operation for causal MSCs. To this end, for each process p in $\mathcal{P}$, we fix a concurrent alphabet (Σ_p, I_p) ([4]), where I_p is an

independence relation over the alphabet of actions Σ_p, i.e. $I_p \subseteq \Sigma_p \times \Sigma_p$ is a symmetric and irreflexive relation. We denote the dependence relation $(\Sigma_p \times \Sigma_p) - I_p$ by D_p. For convenience, we also define $I = \bigcup_{p\in\mathcal{P}} I_p$ and $D = \bigcup_{p\in\mathcal{P}} D_p$. Following the usual definitions of Mazurkiewicz traces, for each concurrent alphabet (Σ_p, I_p), we define the associated trace equivalence $\sim_p$ over $\Sigma_p^\star$ to be the least equivalence relation such that, for any u, v in $\Sigma_p^\star$ and α, β in Σ_p, $\alpha\ I_p\ \beta$ implies $u\alpha\beta v \sim_p u\beta\alpha v$. We let $[u]_p$ (or simply $[u]$) denote the equivalence class containing u, with respect to $\sim_p$. We can now define the composition of two causal MSCs.

Definition 2. *Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ and $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ be two causal MSCs. We define the* concatenation *of B with B', denoted by $B \circledcirc B'$, as the causal MSC $B'' = (E'', \lambda'', \{\sqsubseteq''_p\}, \ll'')$ where*

- *E'' is the disjoint union of E and E', $\lambda'' = \lambda \cup \lambda'$ and $\ll'' = \ll \cup \ll'$.*
- *For each process p in $\mathcal{P}$, $\sqsubseteq''_p$ is the transitive closure of $\sqsubseteq_p \bigcup \sqsubseteq'_p \bigcup \{(e, e') \in E_p \times E'_p \mid \lambda(e)\ D_p\ \lambda'(e')\}$.*

Clearly $\circledcirc$ is a well-defined and associative operation. We note that the composition of causal MSCs is more general than the usual composition of traces defined as $[u].[v] = [uv]$, since the concurrency in each causal MSC need not be compatible with the independence relations $\{I_p\}_{p\in\mathcal{P}}$. For instance, we can define a causal MSC containing two events e_1 and e_2 with $\lambda(e_1) D_p \lambda(e_2)$ but $e_1 \not\sqsubseteq_p e_2$ and $e_2 \not\sqsubseteq_p e_1$. Thus for a causal MSC B we may find a word $u \in Lin(B)$ such that $[u] \not\subseteq Lin(B)$. Note also that if B and B' are MSCs and $I_p = \emptyset$ for every $p \in \mathcal{P}$, then $B \circledcirc B'$ is the usual weak (asynchronous) sequential composition of B with B' [15], denoted as $B \circ B'$.

Through the rest of the paper, we fix a family of Mazurkiewicz trace alphabets $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$. We can now define causal High-level MSCs.

Definition 3. *A* causal High-level MSC *("causal HMSC" for short) is a structure $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ where:*

- *N is a finite nonempty set of* nodes*.*
- *$N_{in} \subseteq N$ is the set of initial nodes.*
- *$\mathcal{B}$ is a finite nonempty set of causal MSCs.*
- *$N_{fi} \subseteq N$ is the set of final nodes.*
- *$\longrightarrow \subseteq N \times \mathcal{B} \times N$ is the transition relation.*

Let $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ be a causal HMSC. A *path* in H is a sequence $\rho = n_0 \xrightarrow{B_1} n_1 \xrightarrow{B_2} n_2 \cdots n_{\ell-1} \xrightarrow{B_\ell} n_\ell$. We refer to n_0 as the starting node and to n_ℓ as the ending node of path ρ. The path ρ is *accepting* iff $n_0 \in N_{in}$ and $n_\ell \in N_{fi}$. A *cycle* in H is a path starting and ending in the same node. The causal MSC generated by ρ, denoted by $\circledcirc(\rho)$, is $B_1 \circledcirc B_2 \circledcirc \cdots \circledcirc B_\ell$. A causal MSC B is said to be generated by H iff B is generated by an accepting path of H. We let $CaMSC(H)$ denote the set of causal MSCs generated by H. Similarly, we will denote by $Vis(H)$ and $Lin(H)$ the set of visual extensions and linearizations of all

causal MSCs in $CaMSC(H)$. We note that $Lin(H) = \bigcup\{Lin(v) \mid v \in Vis(H)\}$, i.e. the set of linearizations generated by H is the set of linearizations generated by its visual extensions. Finally, there is *a priori*, no ordering among receptions of messages (even for messages of the same kind). Hence, messages exchanged between two processes in the same direction can overtake each other, and no FIFO ordering is required.

Definition 4. *An MSC language L is* finitely generated *[12] iff there exists a finite set of MSCs X such that any MSC B in L can be defined as the weak sequential composition of elements of X, i.e. $B = B_1 \circ B_2 \cdots \circ B_t$ for some $B_1, \ldots, B_t$ in X.*

By definition, HMSCs define finitely generated MSC languages. For a causal HMSC H, in general, $Vis(H)$ is *not* finitely generated. Consider for instance the causal HMSC H of Figure 1, with the independence relations $I_p = \{((p!q(Q), p?q(A)), (p?q(A), p!q(Q)))\}$ and $I_q = \emptyset$. The language of visual extensions of H contains MSCs like N of Figure 2, and thus is not finitely generated.

3 Regularity and model-checking for causal HMSCs

3.1 Semantics for causal HMSCs

As things stand, a causal HMSC H defines three syntactically different languages, namely its linearizations language $Lin(H)$, its visual extensions (MSC) language $Vis(H)$ and its causal MSC language $CaMSC(H)$. The next proposition shows that they are also semantically different in general. It also identifies the restrictions under which they match semantically.

Proposition 1. *Let I be an independence relation. Let G and H be two causal HMSCs using I. Consider the following three hypotheses:*

1. $CaMSC(G) = CaMSC(H)$
2. $Vis(G) = Vis(H)$
3. $Lin(G) = Lin(H)$

- *Then $1 \implies 2$ and $2 \implies 3$, but the converses do not hold in general.*
- *Suppose for every causal MSC M labeling transitions of G and H, M does not have autoconcurrency and is FIFO, then $3 \implies 2$. We say M does not have autoconcurrency iff, for any events e, e' in M, $\lambda(e) = \lambda(e')$ implies $e \leq e'$ or $e' \leq e$. We say that M is FIFO if for any messages (e, f), (e', f') in M where e, e' belong to process p, f, f' belong to process q, we have $e \leq e'$ iff $f \leq f'$.*
- *Suppose for every causal MSC M labeling transitions of G and H, M respects the independence relation, then $2 \implies 1$. We say that M respects the independence relation iff, for every $p \in \mathcal{P}$, if e, e' are events belonging to p, $\lambda(e) D_p \lambda(e')$ implies that $e \leq e'$ or $e' \leq e$, and $e \preccurlyeq_p e'$ implies that $\lambda(e) D_p \lambda(e')$, where $\preccurlyeq_p$ denotes the cover relation of $\sqsubseteq_p$.*

For most purposes, the relevant semantics for causal HMSCs seems to be the set of its visual extensions.

3.2 Regular sets of linearizations

It is undecidable in general whether an HMSC has a regular linearization language [13]. A subclass of HMSCs called regular [13] (or bounded [2]) HMSCs, was identified. The linearization language of every regular HMSC is regular, and it is decidable whether an HMSC is regular. We extend this result to causal HMSCs. First, let us recall the notions of connectedness from Mazurkiewicz trace theory [4], and of communication graphs [2, 13, 5]. Let $p \in \mathcal{P}$, and $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ be a causal MSC. We say that $\Gamma \subseteq \Sigma_p$ is D_p-connected iff the (undirected) graph $(\Gamma, D_p \cap (\Gamma \times \Gamma))$ is connected. Moreover, we define the *communication graph* of B, denoted by CG_B, to be the directed graph $(Q, \rightsquigarrow)$, where $Q = \{p \in \mathcal{P} \mid E_p \neq \emptyset\}$ and $\rightsquigarrow \subseteq Q \times Q$ is given by $(p, q) \in \rightsquigarrow$ iff $\ll \cap (E_p \times E_q) \neq \emptyset$. Now we say the causal MSC B is *tight* iff its communication graph CG_B is weakly connected and for every $p \in \mathcal{P}$, $Alph_p(B)$ is D_p-connected. If moreover CG_B is strongly connected and the independence alphabet and B does not have autoconcurrency (that is, events in B with the same labels are totally ordered), then B is *rigid*. We will focus here on rigidity and study the notion of tightness in section 3.3.

Let $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ be a causal HMSC. We say that H is *regular* iff for every cycle ρ in H, the causal MSC $\circledcirc(\rho)$ is rigid. For instance, the simple protocol modeled by the causal HMSC of Figure 4, is regular, since the only loop is labeled by two local events a, b, one message from p to q and one message from q to p. The communication graph associated to this loop is then strongly connected, $p!q(m) - b - p?q(n)$ on process p is connected, and $q!p(n) - a - q?p(m)$ on process q is connected. Equivalently, H is regular iff for every strongly connected subgraph G of H with $\{B_1, \ldots, B_\ell\}$ being the set of causal MSCs appearing in G, we have $B_1 \circledcirc \ldots \circledcirc B_\ell$ is rigid. Note that the rigidity of $B_1 \circledcirc \ldots \circledcirc B_\ell$ does not depend on the order in which $B_1, \ldots, B_\ell$ are listed. This leads to a co-NP-complete algorithm to test whether a causal HMSC is regular.

In the same way, H is *globally-cooperative* iff for every strongly connected subgraph G of H with $\{B_1, \ldots, B_\ell\}$ being the set of causal MSCs appearing in G, we have $B_1 \circledcirc \ldots \circledcirc B_\ell$ is tight.

Theorem 1. *Let $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ be a regular causal HMSC. Then $Lin(H)$ is a regular subset of $\Sigma^\star$, i.e. we can build an automaton $\mathcal{A}_H$ over Σ that recognizes $Lin(H)$. Furthermore, $\mathcal{A}_H$ has at most $\left(|N|^2 \cdot 2^{|\Sigma|} \cdot 2^{|N| \cdot |\Sigma| \cdot 2^{\mathcal{B}|}}\right)^{|N| \cdot |\Sigma| \cdot 2^{|\mathcal{B}|}}$ nodes.*

Intuitively, for a word σ in $\Sigma^\star$, $\mathcal{A}_H$ guesses an accepting path ρ of H and checks whether σ is in $Lin(\circledcirc(\rho))$. Upon reading a prefix σ' of σ, $\mathcal{A}_H$ memorizes a sequence of subpaths from which σ' was "linearized". The crucial step is to ensure that at any time, it suffices to remember a *bounded* number of such subpaths, and moreover, a *bounded* amount of information for each subpath.

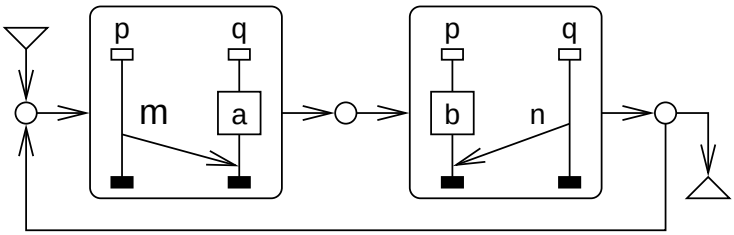

Fig. 4. A non finitely generated tight causal HMSC

3.3 Inclusion and intersection of causal HMSCs

It is known that verification (of inclusion, intersection and equality) of HMSCs is undecidable [13]. Clearly, if a class of scenarios has a regular set of linearizations, as is the case for regular causal HMSCs, then these questions become decidable. However, we do not need $Lin(H)$ to be regular for effective verification. This is so since for a suitable choice of K, the set of K-bounded linearizations of any globally cooperative HMSC is regular, and this suffices for doing effective verification [6]. Unfortunately, this result uses Kuske's encoding [11] into traces that is based on the existence of an (existential) bound on communication. Consider the causal HMSC H of Figure 5. It is globally cooperative (but not regular), and generates a set of visual extensions that contains the causal MSC language shown on the right of Figure 5: in order to receive the first message from p to r, the message from p to q and the message from q to r have to be sent and received. Hence every message from p to r has to be sent before receiving the first message from p to r, which means that H is not existentially bounded. As a consequence, we cannot use Kuske's encoding to show decidability of model-checking for globally cooperative causal HMSCs.

To show decidability of model-checking for globally cooperative causal HMSCs, we adapt the notion of atoms [1, 9] and the proofs from [5]. Let us first introduce a notion of decomposition into basic parts for causal MSCs.

Definition 5. *A causal MSC B is a* basic part *(w.r.t. a dependency relation D) if it can not be decomposed into two causal MSCs B_1 and B_2 such that $B = B_1 \circledcirc B_2$.*

Note that causal MSCs B_1 and B_2 in this definition can not be empty as causal MSCs are defined over non empty sets of events. Extending the above definition, a decomposition of a causal MSC B is a word $B_1.B_2 \cdots B_k$ of basic parts of B such that $B = B_1 \circledcirc B_2 \cdots \circledcirc B_k$. We can extend the independence relation I_p from letters to alphabets, writing $A\ I_p\ B$ if for all a_i in A and b_i in B,

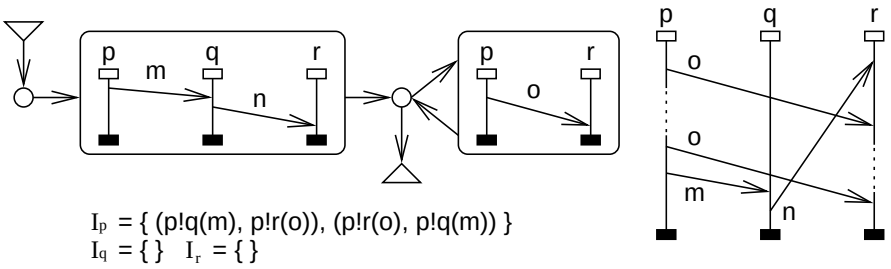

Fig. 5. A globally-cooperative causal HMSC that is not existentially bounded

$a_i\, I_p\, b_i$. Let $\mathcal{B}_{\text{Basic}}$ denote a set of basic parts. We can define the trace alphabet $(\mathcal{B}_{\text{Basic}}, \|)$ with $B_1 \parallel B_2$ if their labels are instance-wise mutually independent, i.e. for every p in $\mathcal{P}$, $Alph_p(B_1)\, I_p\, Alph_p(B_2)$. We denote by $\sim$ the associated equivalence relation. For a language $L \subseteq \mathcal{B}^\star$, we define its closure $[L] = \{u \mid \exists v \in L \wedge u \sim v\}$ by the independence relation $\|$.

Proposition 2. *The decomposition of any causal MSC B is unique up to commutation. That is, given any decomposition $B_1.B_2 \cdots B_k$ of B, any other decomposition of B belongs to $[B_1 \dots B_k]$.*

Without loss of generality, we assume through the rest of the paper that every transition of every causal HMSC is labeled by a basic part (if not, we just decompose the label into basic parts and decompose the transition into a sequence of transitions labeled by these basic parts). Given a causal HMSC H, we denote by $\mathcal{B}_{\text{Basic}}(H)$ the labels of transitions of H. Proposition 2 implies that a causal MSC is uniquely defined by its basic part decomposition. Then instead of the linearization language (which can not reflect autoconcurrency), we can use the basic part language of H, denoted by $BP(H) = \{B_1 \dots B_k \in \mathcal{B}_{\text{Basic}}(H)^\star \mid B_1 \circledcirc \dots \circledcirc B_k \in CaMSC(H)\}$. Notice that $BP(H) = [BP(H)]$ is closed by commutation since for all $A_1 \cdots A_n \sim B_1 \cdots B_n$, we have $A_1 \circledcirc \cdots \circledcirc A_n = B_1 \circledcirc \cdots \circledcirc B_n$. We can also view H as an automaton over alphabet $\mathcal{B}_{\text{Basic}}(H)$, and denote by $\mathcal{L}_{\text{Basic}}(H) = \{B_1 \cdots B_l \in \mathcal{B}_{\text{Basic}}(H)^\star \mid n_0 \xrightarrow{B_1} n_1 \cdots \xrightarrow{B_\ell} n_\ell$ is an accepting path of H$\}$ its associated (regular) language. We now relate $BP(H)$ and $\mathcal{L}_{\text{Basic}}(H)$.

Theorem 2. *Let H be a causal HMSC. Then $BP(H) = [\mathcal{L}_{Basic}(H)]$.*

Assuming we know how to compute the trace closure of the regular language $\mathcal{L}_{Basic}(H)$, we can obtain $BP(H)$ with the help of Theorem 2. In general, we cannot effectively compute this language. However if H is globally cooperative, then $[\mathcal{L}_{Basic}(H)]$ is effectively regular. Further it can be represented as an automaton of size exponential in the size of H [4, 13]. It is not difficult to see that *globally cooperative* causal HMSCs (see section 3.2) are exactly the *globally cooperative* automata over basic parts. We can now obtain the following corollary.

Corollary 1. *Let I be an independance relation. Let H and H' be two causal HMSC using I. If H' is globally cooperative, then we can build an automaton $\mathcal{A}'$ such that $\mathcal{L}_{Basic}(A') = [\mathcal{L}_{Basic}(H')]$. Furthermore, denoting by b the cardinality of $\mathcal{B}_{Basic}(H)$, $\mathcal{A}'$ has at most $2^{O(b\cdot|H'|)}$ nodes. It implies the complexity to decide the following questions:*

- *$CaMSC(H) \subseteq CaMSC(H')$ is PSPACE-complete.*
- *$CaMSC(H) \cap CaMSC(H') = \emptyset$ is EXPSPACE-complete.*

This corollary shows that we can model check a causal HMSC against a globally cooperative causal HMSC specification. We note that the independence relations for concatenation must be the same for both causal HMSCs in order to be able to check inclusion or intersection of basic parts languages. Suppose, in addition, the causal MSCs labeling the transitions of H and H' respect I. Then we will be able to compare the MSC semantics $Vis(H)$ and $Vis(H')$.

On the other hand, if the independence relations are different, then the only way to compare two causal HMSCs is to compare their linearization languages. Consequently, we would need to work with regular causal HMSCs.

3.4 Decomposition into basic parts

We still need an algorithm to check whether a causal MSC is a basic part, and compute its decomposition in order to transform a causal HMSC into a basic part automaton. We now show that both these questions can be answered with the same algorithm without checking every possible partition of the event set and hence avoiding an exponential time complexity. The following proposition shows that decomposition construction can be reduced to the detection of strongly connected components in a directed graph defined over events of E. This construction is analogous to the technique in [9], and computes a decomposition of a causal MSC B in quadratic time by finding strongly connected components of a graph.

Proposition 3. *Let $\preccurlyeq_p$ denote the cover relation of $\sqsubseteq_p$, i.e. $e \preccurlyeq_p e'$ iff $e \sqsubseteq_p e'$ and there is no e_i is such that $e \sqsubseteq_p e_i \sqsubseteq_p e'$. These two problems are equivalent:*

1. *$B = (E, \lambda, \sqsubseteq_p, \ll)$ can be decomposed into $B_1 \dots B_k$, where $B_i = (E_i, \lambda_i, \sqsubseteq_p^i, \ll_i)$ is a basic part.*
2. *The graph $\mathcal{G}_B = (E, R)$ has $\{E_{B_i}\}_{i\in[1..k]}$ as strongly connected components and $\forall i < j, (E_{B_j} \times E_{B_i}) \cap R = \emptyset$. A pair of events $(e, e') \in E$ belongs to R if and only if:*
   - *they are locally related in B, i.e. $e \preccurlyeq_p e'$ for some p in $\mathcal{P}$.*
   - *or they are reception and emission of the same message, i.e. $e \ll e'$ or $e' \ll e$.*
   - *or they are not compatible with local independency relation, i.e. for some p in $\mathcal{P}$, we have $e' \preccurlyeq_p e$ and $\lambda(e) I_p \lambda(e')$,*
   - *or they are not compatible with local dependency relation, i.e. for some p in $\mathcal{P}$, we have $e' \not\sqsubseteq_p e$, $e \not\sqsubseteq_p e'$, and $\lambda(e) D_p \lambda(e')$.*

4 Bounded-window causal HMSCs

One of the chief attractions of causal MSCs is they enable the specification of behaviors containing braids of arbitrary size such as those generated by sliding windows protocols. Very often, sliding windows protocols appear in a situation where two processes p and q exchange bidirectional data. Messages from p to q are of course used to transfer information, but also to acknowledge messages from q to p. If we abstract the type of messages exchanged, these protocols can be seen as a series of query messages from p to q and answer messages from q to p. Implementing a sliding window means that a process may send several queries in advance without needing to wait for an answer to each query before sending the next query. Very often, these mechanisms tolerate losses, i.e. the information sent is stored locally, and can be retransmitted if needed (as in the alternating bit protocol). To avoid memory leaks, the number of messages that can be sent in advance is often bounded by some integer k, that is called the size of the sliding window. Note however that for scenario languages defined using causal HMSCs, such window sizes do not always exist. This is the case for example for the causal HMSC depicted Figure 1 with independence relations $I_p = \{((p!q(Q), p?q(A)), (p?q(A), p!q(Q)))\}$ and $I_q = \{((q?p(Q), q!p(A)), (q!p(A), q?p(Q))\}$. The language generated by this causal HMSC contains scenarios where an arbitrary number of messages from p to q can cross an arbitrary number of messages from q to p. A question that naturally arises is to know if the number of messages crossings is bounded by some constant in all the executions of a protocol specified by a causal HMSC. In what follows, we define these crossings, and show that their boundedness is a decidable problem.

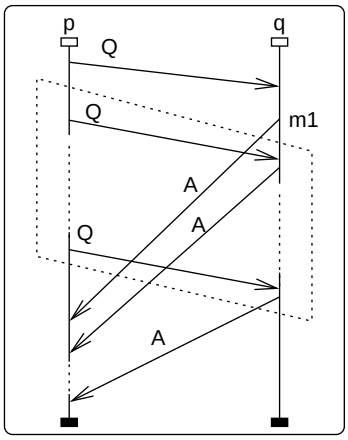

Fig. 6. Window of message m_1

Definition 6. *Let M be a visual extension of a causal MSC, and let (e, f) be a message of M. The* window *of (e, f) is the set $W_M(e, f) = \{(e', f') \in \ll| loc(e') = loc(f) \wedge loc(f') = loc(e) \wedge e \leq f' \wedge e' \leq f\}$.*

We say that a causal HMSC H is window bounded *if and only if there exists an integer n such that for all $M \in Vis(H)$ and for all message (e, f) of M, $|W_M(e, f)| \leq n$.*

Figure 6 illustrates the definition of windows, where the window of the message m_1 is symbolized by the area delimited by dotted lines. It consists of all but the first message Q from p to q. Clearly, the causal HMSC H of Figure 1 is not window bounded. We now describe an algorithm to check whether a causal HMSC is window bounded. It builds an automaton that remembers the labels of events that must appear in the future of messages (respectively in the past) in any visual extension of $CaMSC(H)$.

Let $B = (E, \lambda, \{\sqsubseteq_p\}_{p\in\mathcal{P}}, \ll)$ be a causal MSC, and (e, f) one of its message. We can define the two following sets:

$$Future_B(e, f) = \{a \in \Sigma \mid \exists x \in E, f \leq x \wedge \lambda(x) = a\}$$
$$Past_B(e, f) = \{a \in \Sigma \mid \exists x \in E, x \leq e \wedge \lambda(x) = a\}$$

Consider an occurrence m of a message. Clearly, messages with emission in $Future_B(m)$ or reception in $Past_B(m)$ do not belong to the window of m. On Figure 6, $Past_B(m_1) = \{p!q(Q), q?p(Q), q!p(A)\}$.

Proposition 4. *Let M and N be two causal MSCs, and let m be a message of M. Then we have:*

$$Future_{M\circledcirc N}(m) = Future_M(m) \cup \{a' \in \Sigma \mid \exists x, y \in E_N \\ \exists a \in Future_M(m) \text{ s.t. } \lambda(y) = a' \wedge x \leq_N y \wedge a \ D \ \lambda(x)\}$$

The previous proposition states that $Future_{M_1 \cdots M_n}$ is increasing. Furthermore, it can be computed on the fly, that is with an automaton. Similarly we can compute $Past_{M_1 \cdots M_n}$ on the fly. It now follows that if the window size of a message (e, f) is bounded, then this window size is bounded by some number where this number depends only on the size of the graph and the size of the alphabet.

Proposition 5. *Let $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ be a causal HMSC, and (e, f) be a message of H. Then if the window size of (e, f) is bounded, it is smaller than $|\mathcal{B}||N|(|\Sigma| + 1)$.*

Moreover, to decide whether the maximal window bound for a given message m can be reached in an efficient way by construction of an automaton that memorizes $Future(m)$ and $Past(m)$.

Theorem 3. *Let $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ be a causal HMSC. The boundedness of H is decidable in time $O(|\mathcal{B}||N|^2 2^{|\Sigma|})$.*

5 Relationship with other scenario models

We compare here the expressive powers of different HMSC-based scenario languages with causal HMSCs. Two important strict HMSC subclasses are (i) *regular* [13] (also called bounded in [2]) HMSCs which ensure that the linearizations form a regular set and (ii) *globally-cooperative* HMSCs [5], which ensure that the B-bounded linearizations form a regular set for some bound B. By their very definition, causal HMSCs, regular causal HMSCs and globally-cooperative causal HMSCs extend respectively HMSCs, regular HMSCs and globally-cooperative HMSCs.

On the other hand, Figure 5 proves that regular causal HMSCs is a strict subclass of globally-cooperative causal HMSCs, which is trivially a strict subclass of causal HMSCs. Figure 4 is an example of a regular causal HMSC whose language is not finitely generated, hence it proves that (regular/globally-cooperative) causal HMSCs are strictly more powerful than (regular/globally-cooperative) HMSCs.

Another extension of HMSCs consists of *Compositional* HMSCs [7], or CHMSCs for short. CHMSCs are HMSCs which allow dangling communication edges. i.e. where the message relation $\ll$ is only a partial non surjective mapping contained in $E_! \times E_?$. The concatenation of two Compositional MSCs $M1 \circ M2$ performs the instance-wise concatenation as for HMSCs, and computes a new message pairing relation $\ll$ defined over $(E_{1_!} \cup E_{2_!}) \times (E_{1_?} \cup E_{2_?})$ extending $\ll_1 \cup \ll_2$, and preserving the FIFO ordering of messages of same content (actually, in the definition [7] considers, there is no channel content).

A CHMSC H generates a set of MSCs, denoted $Vis(H)$ by abuse of notation, obtained by concatenation of MSCs along a path of the graph. With this definition, some path of a CHMSC may not generate any correct MSC. Moreover, a path of a CHMSC generates at most one MSC. The class of CHMSC for which each path generates exactly one MSC is called *safe* CHMSC, still a strict extension over HMSCs. Regular and globally cooperative HMSCs have also their strict extensions in terms of safe CHMSCs, namely as regular CHMSC and globally cooperative CHMSCs.

It is not hard to build a regular Compositional HMSC H with $Vis(H) = \{M_i \mid i = 0, 1, \ldots\}$ where each M_i consists of an emission event e from p to r, then a sequence of i blocks of three messages: a message from p to q followed by a message from q to r then a message from r to p. And at last the reception event on r from p matching e. That is, H is not finitely generated. A causal HMSC cannot generate the same language. Assume for contradiction, a causal HMSC G with $Vis(G) = Vis(H)$. Let k be the number of messages of the biggest causal MSC which labels a transition of G. We know that M_k is in $Vis(G)$, hence $N_1, \ldots, N_m$ labels a path of G with $M_{k+1} \in N_1 \circledcirc \cdots \circledcirc N_m$ and $m \geq 2$ because of the size k. It also means that $N_1 \circ \cdots \circ N_m \in Vis(G)$, that is, there exists j with $M_j = N_1 \circ \cdots \circ N_m$, a contradiction since the left part of the equality is a basic part and the right part is not. That is (regular) compositional HMSCs are not included into causal HMSCs. On the other hand, regular causal HMSCs have a regular set of linearizations (Theorem 1) and it has been shown that regular

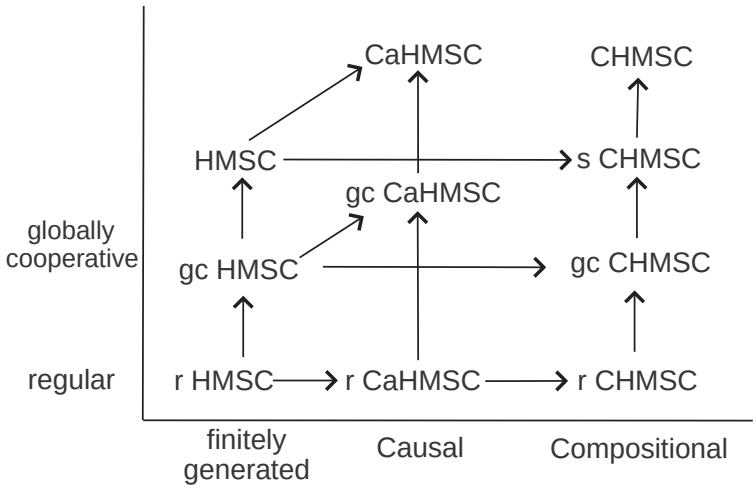

Fig. 7. Comparison of Scenario languages and CFMs

compositional HMSC captures all the MSC languages that have a regular set of linearizations, hence regular causal HMSC are included into regular compositional HMSC. Last, we already know with Figure 5 that globally-cooperative causal HMSCs are not necessarily existentially bounded, hence they are not included into safe Compositional HMSC. Furthermore, globally-cooperative causal HMSCs are not included into CHMSC because of possible autoconcurrency.

The relationships between these scenario models are summarized by Figure 7, where arrows denote *strict* inclusion of languages. Two classes are uncomparable when they are not connected by a transitive sequence of arrows. We use the abbreviation *r* for regular, *gc* for globally-cooperative, *s* for safe, *CaHMSC* for causal HMSCs and *CHMSC* for compositional HMSCs.

6 Conclusion

We have defined an extension of HMSCs called causal HMSCs that allows the definition of braids, such as those appearing in sliding windows protocols. We also identified in this setting, many subclasses of scenarios that were defined for HMSCs which have decidable verification problems. An interesting class that emerges is globally-cooperative causal HMSCs. This class is incomparable with safe Compositional HMSCs because the former can generate non existentially bounded behaviors. Yet, for this class the generic model-checking problems are decidable.

An interesting open problem is deciding whether the set of visual extensions of a causal HMSC is finitely generated. Yet another interesting issue is to consider the class of causal HMSCs that have bounded crossing windows for all their messages. The set of behaviors generated by these causal HMSCs seems to exhibit a kind of regularity that could be exploited. Finally, designing suitable

machine models (along the lines of Communicating Finite Automata [3]) is also an important future line of research.

References

1. M. Ahuja, A.D. Kshemkalyani, and T. Carlson. A basic unit of computation in distributed sytems. In *Proc. of ICDS'90*, pages 12–19, 1990.
2. R. Alur and M. Yannakakis. Model checking of message sequence charts. In *Proc. of CONCUR'99*, number 1664 in LNCS, pages 114–129. Springer, 1999.
3. D. Brand and P. Zafiropoulo. On communicating finite state machines. Technical Report RZ1053, IBM Zurich Research Lab, 1981.
4. V. Diekert and G. Rozenberg, editors. *The book of traces*. World Scientific, 1995.
5. B. Genest, A. Muscholl, H. Seidl, and M. Zeitoun. Infinite-state high-level MSCs: Model-checking and realizability. *Journal of Computer and System Sciences*, 72(4):617–647, 2006.
6. Blaise Genest, Dietrich Kuske, and Anca Muscholl. A kleene theorem and model checking for a class of communicating automata. *Information and Computation.*, 204(6):920–956, 2006.
7. E. Gunter, A. Muscholl, and D. Peled. Compositional message sequence charts. In *Proc. of TACAS'01*, number 2031 in LNCS. Springer, 2001.
8. J.G. Henriksen, M. Mukund, K. Narayan Kumar, M. Sohoni, and P.S. Thiagarajan. A theory of regular MSC languages. *Information and Computation*, 202(1):1–38, 2005.
9. L. Hélouët and P. Le Maigat. Decomposition of message sequence charts. In *Proc. of SAM'00*, 2000.
10. ITU-TS. *ITU-TS Recommendation Z.120: Message Sequence Chart (MSC)*. ITU-TS, 1999.
11. D. Kuske. Regular sets of infinite message sequence charts. *Information and Computation*, 187(1):80–109, 2003.
12. R. Morin. Recognizable sets of message sequence charts. In *Proc. of STACS'02*, number 2285 in LNCS, pages 523–534. Springer, 2002.
13. A. Muscholl and D. Peled. Message sequence graphs and decision problems on Mazurkiewicz traces. In *Proc. of MFCS'99*, number 1672 in LNCS. Springer, 1999.
14. A. Muscholl, D. Peled, and Z. Su. Deciding properties for message sequence charts. In *Proc. of FoSSaCS'98*, number 1378 in LNCS, pages 226–242. Springer, 1998.
15. M. Reniers. *Message Sequence Chart: Syntax and Semantics*. PhD thesis, Eindhoven University of Technology, 1999.