Causal Message Sequence Charts*

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

[1] IRISA/ENS Cachan, Campus de Beaulieu, 35042 Rennes Cedex, France
thomas.gazagnaire@irisa.fr
[2] IRISA/CNRS, Campus de Beaulieu, 35042 Rennes Cedex, France
blaise.genest@irisa.fr
[3] IRISA/INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France
{loic.helouet, shaofa.yang}@irisa.fr
[4] School of Computing, NUS, Singapore
thiagu@comp.nus.edu.sg

Abstract. Scenario languages based on Message Sequence Charts (MSCs) and related notations have been widely studied in the last decade [14, 13, 2, 9, 6, 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 this 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, which 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.

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

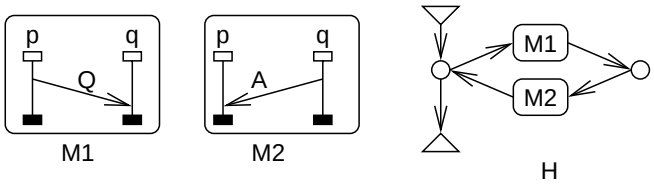


Fig. 1. An HMSC over two MSCs

HMSCs are very expressive and hence a number of basic problems associated with them cannot 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 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, 6] 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 cannot 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 into atoms (the minimal non-trivial MSCs) [9, 12, 6], which does not apply for CHMSCs, and results in a higher complexity [5]. 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 undecidability results.

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 [4] 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.

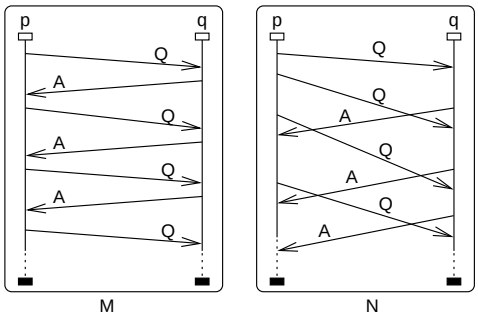


Fig. 2. Two MSCs M and N

A causal HMSC is *a priori* not existentially bounded in the sense defined in [5]. 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 gain decidability for safe CMSCs [5] 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 [6] 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 "window-bounded", 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 acknowledgement to be received. We then show that one can decide if a given causal HMSC generates a window-bounded MSC language. In section 5 we compare the expressive power of languages based on causal HMSCs with other known HMSC-based language classes.

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 $|\mathcal{P}| > 1$. For convenience, we let p, q range over $\mathcal{P}$ and drop the subscript $p \in \mathcal{P}$ when there is no confusion. We also fix finite nonempty sets Msg, Act of message types and internal action names respectively. 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 $\Sigma_p = \{\alpha \in \Sigma \mid loc(\alpha) = p\}$. In order to define a concatenation operation for causal MSCs, we fix a family of Mazurkiewicz trace alphabets $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$ ([4]), one for each p. That is, $I_p \subseteq \Sigma_p \times \Sigma_p$ is an irreflexive and symmetric relation, called the independence relation. We denote the dependence relation $(\Sigma_p \times \Sigma_p) - I_p$ by D_p. Following the usual definitions of Mazurkiewicz traces, for each (Σ_p, I_p), the associated trace equivalence relation $\sim_p$ over $\Sigma_p^\star$ is 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 $u\alpha\beta v \sim_p u\beta\alpha v$. Equivalence classes of $\sim_p$ are called *traces*. For u in $\Sigma_p^\star$, we let $[u]_p$ denote the trace containing u.

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 of* events*,* $\lambda : E \to \Sigma$ *is a labelling function. And the following conditions hold:*

- *For each process* p*,* $\sqsubseteq_p \subseteq E_p \times E_p$ *is a* partial order*, where* $E_p = \{e \in E \mid \lambda(e) \in \Sigma_p\}$*. We let* $\widehat{\sqsubseteq}_p \subseteq E_p \times E_p$ *denote the least relation such that* $\sqsubseteq_p$ *is the reflexive and transitive closure of* $\widehat{\sqsubseteq}_p$*.*
- $\ll \subseteq E_! \times E_?$ *is a bijection, where* $E_! = \{e \in E \mid \lambda(e) \in \Sigma_!\}$ *and* $E_? = \{e \in E \mid \lambda(e) \in \Sigma_?\}$*. For each* $(e, e') \in \ll$*,* $\lambda(e) = p!q(m)$ *iff* $\lambda(e') = q?p(m)$*.*
- *The transitive closure of the relation* $\big(\bigcup_{p\in\mathcal{P}} \sqsubseteq_p\big) \cup \ll$*, denoted* $\leq$*, is a partial order.*

For each p, the relation $\sqsubseteq_p$ dictates the "causal" order in which events of E_p may be executed. The relation $\ll$ identifies pairs of message-emission and message-reception events. We say $\sqsubseteq_p$ *respects* the trace alphabet (Σ_p, I_p) iff for any $e, e' \in E_p$, the following hold: (i) $\lambda(e)\ D_p\ \lambda(e')$ implies $e\ \sqsubseteq_p\ e'$; (ii) $e\ \widehat{\sqsubseteq}_p\ e'$ implies $\lambda(e)\ D_p\ \lambda(e')$. The causal MSC B is said to respect $\{(\Sigma_p, I_p)\}$ iff $\sqsubseteq_p$ respects (Σ_p, I_p) for every p. In order to gain modelling power, we allow each $\sqsubseteq_p$ to be *any* partial order, not necessarily respecting (Σ_p, I_p). We say that the causal MSC B is *FIFO*[1] iff for any $(e, f) \in \ll$, $(e', f') \in \ll$ such that $\lambda(e) = \lambda(e') = p!q(m)$ (and thus $\lambda(f) = \lambda(f') = q?p(m)$), we have either $e \sqsubseteq_p e'$ and $f \sqsubseteq_q f'$; or $e' \sqsubseteq_p e$ and $f' \sqsubseteq_q e'$. Note that we do not demand *a priori* that a causal MSC must be FIFO.

Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ be a causal MSC. 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. Clearly, $Lin(B)$ is nonempty. We set $Alph(B) = \{\lambda(e) \mid e \in E\}$, and $Alph_p(B) = Alph(B) \cap \Sigma_p$ for each p.

The leftmost part of Figure 3 depicts a causal MSC M. In this diagram, we enclose events of each process p in a vertical box and show the partial order $\sqsubseteq_p$ in the standard way. In case $\sqsubseteq_p$ is a total order, we place events of p along a

[1] There are two notions of FIFOness for MSCs in the literature. One allows overtaking of messages with different message types via the same channel, while the other does not. As our results hold for both notions, we choose the more permissive one.

vertical line with the minimum events at the top and omit the box. In particular, in M, the two events on p are not ordered (i.e. $\widehat{\sqsubseteq}_p$ is empty) and $\sqsubseteq_q$ is a total order. Members of $\ll$ are indicated by horizontal or downward-sloping arrows labelled with the transmitted message. Both words $p!q(Q).q!p(A).q?p(Q).p?q(A)$ and $q!p(A).p?q(A).p!q(Q).q?p(Q)$ are linearizations of M.

An *MSC* $B = (E, \lambda, \{\sqsubseteq_p\}_{p\in\mathcal{P}}, \ll)$ is defined in the same way as a causal MSC except that every $\sqsubseteq_p$ is required to be a *total order*. In an MSC B, the relation $\sqsubseteq_p$ must be interpreted as the visually observed order of events in one sequential execution of p. Let $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ be a causal MSC. Then we say the MSC B is a *visual extension* of B' if $E' = E$, $\lambda' = \lambda$, $\sqsubseteq'_p \subseteq \sqsubseteq_p$ and $\ll' = \ll$. We let $Vis(B')$ denote the set of visual extensions of B'. In Figure 3, $Vis(M)$ consists of MSCs $M1, M2$.

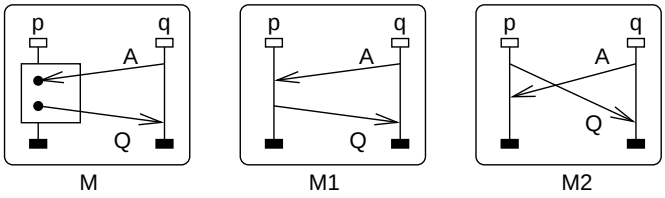


Fig. 3. A causal MSC M and its visual extensions $M1, M2$.

We shall now define the concatenation operation of causal MSCs using the trace alphabets $\{(\Sigma_p, I_p)\}$.

Definition 2. *Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ and $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ be 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'. λ'' is given by: $\lambda''(e) = \lambda(e)$ if $e \in E$ and $\lambda''(e) = \lambda'(e)$ if $e \in E$. And $\ll'' = \ll \cup \ll'$.*
- *For each 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. Note that in case B and B' are MSCs and $D_p = \Sigma_p \times \Sigma_p$ for every p, then the result of $B \circledcirc B'$ is the asynchronous concatenation (also called weak sequential composition) of B with B' [15], which we denote by $B \circ B'$. We also remark that the concatenation of causal MSCs is different from the concatenation of traces. The concatenation of trace $[u]_p$ with $[v]_p$ is the trace $[uv]_p$. However, a causal MSC B need not respect $\{(\Sigma_p, I_p)\}$. Consequently, for a process p, $Lin(B)$ may contain a word u such that the projection of u on $Alph_p(B)$ is *not* a trace.

We can now define causal HMSCs.

Definition 3. *A* causal HMSC *over* $(\mathcal{P}, \{(\Sigma_p, I_p)\})$ *is a structure* $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ *where* N *is a finite nonempty set of* nodes, $N_{in} \subseteq N$ *the set of initial nodes,* $\mathcal{B}$ *a finite nonempty set of causal MSCs,* $\longrightarrow \subseteq N \times \mathcal{B} \times N$ *the transition relation, and* $N_{fi} \subseteq N$ *the set of final nodes.*

A *path* in the causal HMSC H is a sequence $\rho = n_0 \xrightarrow{B_1} n_1 \xrightarrow{B_2} \cdots \xrightarrow{B_\ell} n_\ell$. If $n_0 = n_\ell$, then we say ρ is a cycle. The path ρ is *accepting* iff $n_0 \in N_{in}$ and $n_\ell \in N_{fi}$. The causal MSC generated by ρ, denoted $\circledcirc(\rho)$, is $B_1 \circledcirc B_2 \circledcirc \cdots \circledcirc B_\ell$. Note that the concatenation operation $\circledcirc$ is associative. We let $CaMSC(H)$ denote the set of causal MSCs generated by accepting paths of H. We also set $Vis(H) = \bigcup\{ Vis(M) \mid M \in CaMSC(H)\}$ and $Lin(H) = \bigcup\{Lin(M) \mid M \in CaMSC(H)\}$. Obviously, $Lin(H)$ is also equal to $\bigcup\{Lin(M) \mid M \in Vis(H)\}$. We shall refer to $CaMSC(H)$, $Vis(H)$, $Lin(H)$, respectively, as the causal language, visual language and linearization language of H.

An *HMSC* $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ is defined in the same way as a causal HMSC except that $\mathcal{B}$ is a finite set of MSCs and every MSC in $\mathcal{B}$ is FIFO. A path ρ of H generates an MSC by concatenating the MSCs along ρ. We let $Vis(H)$ denote the set of MSCs generated by accepting paths of H with $\circ$, and call $Vis(H)$ the visual language of H. Recall that an MSC language (i.e. a collection of MSCs) L is *finitely generated* [12] iff there exists a finite set X of MSCs satisfying the condition: for each MSC B in L, there exist $B_1, \ldots, B_\ell$ in X such that $B = B_1 \circ \cdots \circ B_\ell$. Many protocols exhibit scenario collections that are *not* finitely generated. For example, sliding window protocols can generate arbitarily large MSCs repeating the communication behaviour shown in MSC N of Figure 2. One basic limitation of HMSCs is that their visual languages are *finitely generated*. In contrast, the visual language of a causal HMSC is *not* necessarily finitely generated. For instance, suppose we view H in Figure 1 as a causal HMSC by considering $M1$,$M2$ as causal MSCs and associating H with the independence relations given by: $I_p = \{((p!q(Q), p?q(A)), (p?q(A), p!q(Q)))\}$ and $I_q = \emptyset$. Then clearly $Vis(H)$ is not finitely generated, as it contains infinitely many MSCs similar to N of Figure 2.

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 linearization language $Lin(H)$, its visual language (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* H, H' *be causal HMSCs over the* same *family of trace alphabets* $\{(\Sigma_p, I_p)\}$. *Consider the following three hypotheses: (i)* $CaMSC(H) = CaMSC(H')$; *(ii)* $Vis(H) = Vis(H')$; *and (iii)* $Lin(H) = Lin(H')$. *Then we have:*

– $(i) \implies (ii)$ *and* $(ii) \implies (iii)$*; but the converses do not hold in general.*
– *If every causal MSC labelling transitions of* H, H' *respects* $\{(\Sigma_p, I_p)\}$*, then* $(ii) \implies (i)$.
– *If every causal MSC labelling transitions of* H, H' *is FIFO, then* $(iii) \implies (ii)$.

Proof. – The implications $(i) \implies (ii)$ and $(ii) \implies (iii)$ follow from the definitions. However, as shown in Figure 4, $Vis(G_1) = Vis(H_1)$ but $CaMSC(G_1) \neq CaMSC(H_1)$. And $Lin(G_2) = Lin(H_2)$ but $Vis(G_2) \neq Vis(H_2)$. Note that the independence relation is immaterial in these examples.
– This follows from the observation that, for any MSCs M_1, M_2 in $Vis(H) \bigcup Vis(H')$, we have that $M_1 \neq M_2$ iff $Lin(M_1) \cap Lin(M_2) = \emptyset$.
– This follows from the observation that, for any causal MSCs B_1, B_2 in $CaMSC(H) \bigcup CaMSC(H')$, we have that $B_1 \neq B_2$ iff $Vis(B_1) \cap Vis(B_2) = \emptyset$. □

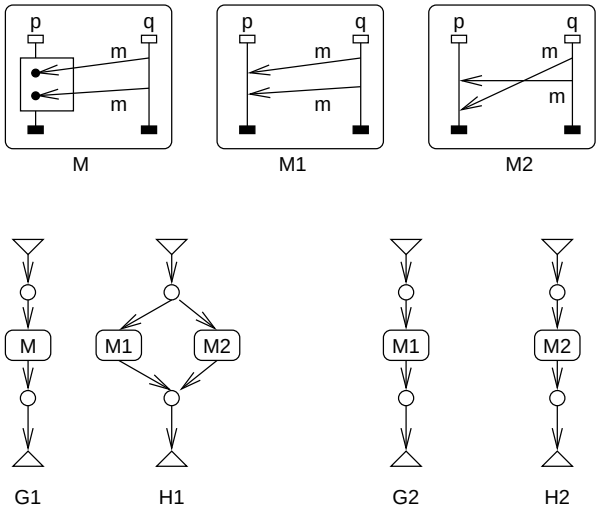


Fig. 4. Relations between linearizations, visual extensions and causal orders

For most purposes, the relevant semantics for a causal HMSC seems to be its visual language.

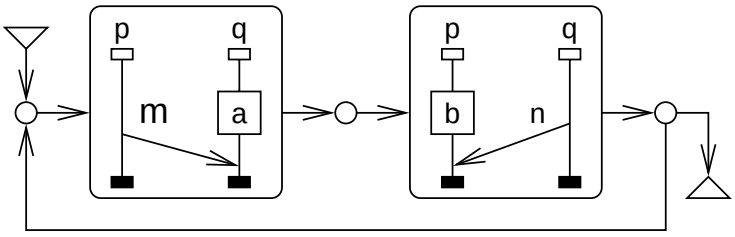
3.2 Regular sets of linearizations

It is undecidable in general whether an HMSC has a regular linearization language [13]. In the literature, a subclass of HMSCs called regular [13] (or

bounded [2]) HMSCs, has been identified. The linearization language of every regular HMSC is regular. And one can effectively whether an HMSC is in the subclass of regular HMSCs. We extend these results to causal HMSCs. First, let us recall the notions of connectedness from Mazurkiewicz traces theory [4], and of communication graphs [2, 13, 6]. 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 connected and for every p, $Alph_p(B)$ is D_p-connected. We say the causal MSC B is *rigid* iff (i) B is FIFO; (ii) CG_B is strongly connected; and (iii) for every p, $Alph_p(B)$ is D_p-connected. We will focus here on rigidity and study the notion of tightness in section 3.3.

Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ 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 5, is regular, since the only cycle 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 cycle 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, we say that 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 that $B_1 \circledcirc \ldots \circledcirc B_\ell$ is tight.



I_p = { (p?q(n), p!q(m)), (p!q(m), p?q(n)) }

I_q = { (q?p(m), q!p(n)), (q!p(n), q?p(m)) }

Fig. 5. A regular causal HMSC which is not finitely generated

Theorem 1. *Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ be a regular causal HMSC. Then $Lin(H)$ is a regular subset of $\Sigma^\star$, that is, we can build a finite state automaton $\mathcal{A}_H$ over Σ that recognizes $Lin(H)$. Furthermore, $\mathcal{A}_H$ has at most $\Big(|N|^2 \cdot 2^{|\Sigma|} \cdot 2^{|N|\cdot|\Sigma|\cdot 2^m}\Big)^{|N|\cdot|\Sigma|\cdot 2^m}$ states, where $m = \max\{|B| \mid B \in \mathcal{B}\}$ with $|B|$ denoting the number of events in B.*

In [11], the regularity of linearization languages of regular HMSC was proved by using an encoding into connected traces and building a finite state automaton which recognizes such connected traces. In our case, finding such embedding into Mazurkiewicz traces seems impossible due to the fact that causal MSCs need not be FIFO. Instead, we shall use techniques from the proof of regularity of trace closures of loop-connected automata from [4, 13].

The rest of this subsection is devoted to the proof of Theorem 1. We fix a regular causal HMSC H as in the theorem, and show the construction of the finite state automaton $\mathcal{A}_H$ over Σ which accepts $Lin(H)$.

First, we establish some technical results.

Lemma 1. *Let $\rho = \theta_1 \dots \theta_2 \dots \theta_{|\Sigma|}$ be a path of H, where for each $i = 1, \dots, |\Sigma|$, the subpath $\theta_i = n_{i,0} \xrightarrow{B_{i,1}} n_{i,1} \dots n_{i,\ell_i-1} \xrightarrow{B_{i,\ell_i}} n_{i,0}$ is a cycle (these cycles need not be contiguous). Suppose further that the sets $\widehat{\mathcal{B}}_i = \{B_{i,1}, \dots, B_{i,\ell_i}\}$, $i = 1, \dots, |\Sigma|$, are equal. Let e be an event in $\circledcirc(\theta_1)$ and e' an event in $\circledcirc(\theta_{|\Sigma|})$. Let $\circledcirc(\rho) = (E, \lambda, \{\sqsubseteq_p\}, \ll)$. Then we have $e \leq e'$.*

Proof. We consider two cases.

—**Case** (i): $loc(\lambda(e)) = loc(\lambda(e'))$.

Let $p = loc(\lambda(e))$. Let us recall that H is regular, thus $Alph_p(\circledcirc(\theta_1)) = \dots = Alph_p(\circledcirc(\theta_K))$ is D_p-connected. Consequently, we can find a set of events $\{e_j\}_{j=1,\dots,t}$, where $t \leq |\Sigma_p| - 2$, each e_j is in $\circledcirc(\theta_{j+1})$, and such that $\lambda(e)\, D_p\, \lambda(e_1)\, D_p \dots\ D_p\ \lambda(e_t)\ D_p\ \lambda(e')$. Thus $e \leq e_1 \leq \dots \leq e_t \leq e'$.

—**Case** (ii): $loc(\lambda(e)) \neq loc(\lambda(e'))$. Let $p_1 p_2 \dots p_t$ be a path from $loc(\lambda(e))$ to $loc(\lambda(e'))$ in the communication graph of $\circledcirc(\rho)$, where $p_1 = loc(\lambda(e))$, $p_t = loc(\lambda(e'))$. In view of the arguments in Case (i), it is easy to see that we can pick events e_i, f_i in $\circledcirc(\theta_{u_i})$, $i = 1, \dots, t-1$, where for each i, $u_i = |\Sigma_{p_1}| + |\Sigma_{p_2}| + \dots + |\Sigma_{p_i}|$, $loc(e_i) = p_i$, $loc(f_i) = p_{i+1}$ and $e_i \ll f_i$. Thus $e \leq e_1 \ll f_1 \leq e_2 \ll f_2 \leq \dots \leq e_{t-1} \ll f_{t-1} \leq e'$. □

Let $\rho = n_0 \xrightarrow{B_1} \cdots \xrightarrow{B_\ell} n_\ell$ be a path in H, where $B_i = (E_i, \lambda_i, \{\sqsubseteq_p^i\}, \ll_i)$ for $i = 1, \dots, \ell$. Let $\circledcirc(\rho) = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$. A *configuration* of ρ is a $\leq$-closed subset of E. Let C be a configuration of ρ. A *C-subpath* of ρ is a maximal subpath $\varrho = n_u \xrightarrow{B_{u+1}} \dots \xrightarrow{B_{u'}} n_{u'}$, such that $C \cap E_i \neq \emptyset$ for each $i = u, \dots, u'$. For such a *C-subpath* ϱ, we define its *C-residue* to be the set $(E_{u+1} \cup E_{u+2} \cup \dots \cup E_{u'}) - C$. Figure 6 illustrates these notions. Each causal MSC is represented by a rectangle. Events in the configuration C are indicated by small filled circles, while events not in C are indicated by small blank circles. The two *C-subpaths* identified on

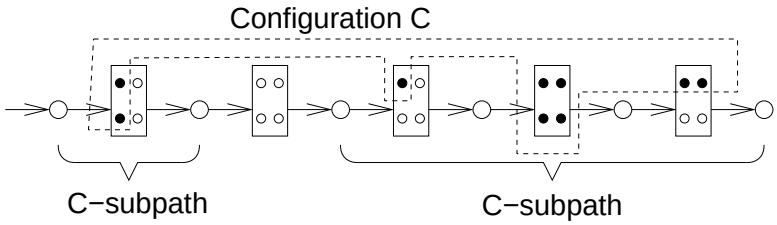


Fig. 6. Events in C-subpaths are indicated by small filled circles. Events in C-residues are indicated by small blank circles.

Figure 6 are the sequences of transitions that provide the events appearing in C.

Lemma 2. *Let ρ be a path in H and C be a configuration of ρ. Then,*

(i) The number of C-subpaths of ρ is at most $K_{subpath} = |N| \cdot |\Sigma| \cdot 2^{|\mathcal{B}|}$.
(ii) Let ϱ be a C-subpath of ρ. Then the number of events in the C-residue of ϱ is at most $K_{residue} = |N| \cdot |\Sigma| \cdot 2^{|\mathcal{B}|} \cdot \max\{|B| \mid B \in \mathcal{B}\}$.

Proof. (i) Suppose the contrary. Let $K = |\Sigma| \cdot 2^{|\mathcal{B}|}$. We can find $K+1$ C-subpaths whose ending nodes are equal. Let the indices of these $K+1$ ending nodes be $i_1 < i_2 < \ldots < i_{K+1}$. For $h = 1, \ldots, K$, let θ_h be the subpath of ρ from n_{i_h} to $n_{i_{h+1}}$; and let $\widehat{\mathcal{B}}_h$ be the set of causal MSCs appearing in θ_h. Hence we can find θ_{j_1}, θ_{j_2}, ..., $\theta_{j_{|\Sigma|}}$, $j_1 < j_2 < \ldots < j_{|\Sigma|}$, such that $\widehat{\mathcal{B}}_{j_1} = \widehat{\mathcal{B}}_{j_2} = \ldots = \widehat{\mathcal{B}}_{j_{|\Sigma|}}$. Pick an event e from $\circledcirc(\theta_{j_1})$ with $e \notin C$. Such an e exists, since, for example, none of the events in the first causal MSC appearing in θ_{j_1} is in C. Pick an event e' from $\circledcirc(\theta_{j_{|\Sigma|}})$ with $e' \in C$. Applying Lemma 1 yields that $e < e'$. This leads to a contradiction, since C is $\leq$-closed.

(ii) Let $\varrho = n_i \xrightarrow{B_{i+1}} \ldots \xrightarrow{B_{i'}} n_{i'}$. Let $\widehat{E}_j = E_j - C$ for $j = i+1, \ldots, i'$. By similar arguments as in (i), it is easy to show that among $\widehat{E}_{i+1}$, ..., $\widehat{E}_{i'}$, at most $|N| \cdot |\Sigma| \cdot 2^{|\mathcal{B}|}$ of them are nonempty. The claim then follows.

We are now ready to define the finite state automaton $\mathcal{A}_H = (S, S_{in}, \Sigma, S_{fi}, \Longrightarrow)$ which accepts $Lin(H)$. As usual, S will be the set of states, $S_{in} \subseteq S$ the initial states, $\Longrightarrow \subseteq S \times \Sigma \times S$ the transition relation, and $S_{fi} \subseteq S$ the final states. Fix $K_{subpath}$, $K_{residue}$ to be the constants defined in Lemma 2. If $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ is a causal MSC and E' a subset of E, then we define the restriction of B to E' to be the causal MSC $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ as follows. As expected, λ' is the restriction of λ to E'; for each p, $\sqsubseteq'_p$ is the restriction of $\sqsubseteq_p$ to $(E' \cap E_p) \times (E' \cap E_p)$; and $\ll'$ is the restriction of $\ll$ to E'.

Intuitively, for a word σ in $\Sigma^\star$, $\mathcal{A}_H$ guesses an accepting path ρ of H and checks whether σ is in $Lin(\circledcirc(\rho))$. After reading a prefix σ' of σ, $\mathcal{A}_H$ memorizes a sequence of subpaths from which σ' was "linearized" (i.e the C-subpath of a path

ρ such that C is a configuration reached after reading σ' and $\circledcirc(\rho)$ contains C). With Lemma 2, it will become clear later that at any time, we should remember at most $K_{subpath}$ such subpaths. Moreover, for each subpath, we need to know only a *bounded* amount of information, which will be stored in a data structure called "segment".

A causal MSC $B_i = (E_i, \lambda, \{\sqsubseteq_p\}, \ll)$ is K-bounded if $|E| \leq K$. A *segment* is a tuple (n, Γ, W, n'), where $n, n' \in N$, Γ is a nonempty subset of Σ, and W is either a non-empty $K_{residue}$-bounded causal MSC, or the special symbol $\perp$. The state set S of $\mathcal{A}_H$ is the collection of finite sequences $\theta_1\theta_2 \dots \theta_\ell$, $0 \leq \ell \leq K_{subpath}$, where each θ_i is a segment. Intuitively, a segment (n, Γ, W, n') keeps track of a subpath ϱ of H which starts at n and ends at n'. Γ is the collection of letters of events in $\circledcirc(\varrho)$ that have been "linearized". Finally, W is the restriction of $\circledcirc(\varrho)$ to the set of events in $\circledcirc(\varrho)$ that are not yet linearized. In case all events in $\circledcirc(\varrho)$ have been linearized, we set $W = \perp$. For convenience, we extend the operator $\circledcirc$ by: $W \circledcirc \perp = \perp \circledcirc W = W$ for any causal MSC W; and $\perp \circledcirc \perp = \perp$.

We define $\mathcal{A}_H = (S, S_{in}, \Sigma, S_{fi}, \Longrightarrow)$ as follows:

- As mentioned above, S is the collection of finite sequence of at most $K_{subpath}$ segments.
- The initial state set is $S_{in} = \{\varepsilon\}$, where ε is the null sequence.
- A state is final iff it consists of a single segment $\theta = (n, \Gamma, \perp, n')$ such that $n \in N_{in}$ and $n' \in N_{fi}$ (and Γ is any nonempty subset of Σ).
- The transition relation $\Longrightarrow$ of $\mathcal{A}_H$ is the least set satisfying the following conditions.

  —**Condition** (i):

  Suppose $n \xrightarrow{B} n'$ where $B = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$. Let e be a minimal event in B (with respect to $\leq$) and let $a = \lambda(e)$. Let $\theta = (n, \Gamma, W, n')$ where $\Gamma = \{a\}$. Let $R = E - \{e\}$. If R is nonempty, then W is the restriction of B to R; otherwise we set $W = \perp$. Suppose $s = \theta_1 \dots \theta_k\theta_{k+1} \dots \theta_\ell$ is a state in S where $\theta_i = (n_i, \Gamma_i, W_i, n_i')$ for each i. Suppose, for every $e' \in E$ with $e \leq e'$, it is the case that $\lambda(e')I_p\gamma$ for any $\gamma \in \Sigma_p \cap (\bigcup_{k+1 \leq i \leq \ell} \Gamma_i)$, where $p = loc(e')$.
  - ("create a new segment") Let $\hat{s} = \theta_1 \dots \theta_k\theta\theta_{k+1} \dots \theta_\ell$. If $\hat{s}$ is in S, then $s \overset{a}{\Longrightarrow} \hat{s}$. In particular, for the initial state ε, we have $\varepsilon \overset{a}{\Longrightarrow} \theta$.
  - ("add to the beginning of a segment") Suppose $n' = n_{k+1}$. Let $\hat{\theta} = (n, \Gamma \cup \Gamma_{k+1}, \widehat{W}, n'_{k+1})$, where $\widehat{W} = W \circledcirc W_{k+1}$. Let $\hat{s} = \theta_1 \dots \theta_k\hat{\theta}\theta_{k+2} \dots \theta_\ell$. If $\hat{s}$ is in S, then $s \overset{a}{\Longrightarrow} \hat{s}$.
  - ("append to the end of a segment") Suppose $n = n'_k$. Let $\hat{\theta} = (n_k, \Gamma_k \cup \Gamma, \widehat{W}, n')$, where $\widehat{W} = W_k \circledcirc W$. Let $\hat{s} = \theta_1 \dots \theta_{k-1}\hat{\theta}\theta_{k+1} \dots \theta_\ell$. If $\hat{s}$ is in S, then $s \overset{a}{\Longrightarrow} \hat{s}$.
  - ("glue two segments") Suppose $n = n'_k$ and $n' = n_{k+1}$. Let $\hat{\theta} = (n_k, \Gamma_k \cup \Gamma \cup \Gamma_{k+1}, \widehat{W}, n'_{k+1})$, where $\widehat{W} = W_k \circledcirc W \circledcirc W_{k+1}$. Let $\hat{s}$ be $\theta_1 \dots \theta_{k-1}\hat{\theta}\theta_{k+2} \dots \theta_\ell$. If $\hat{s}$ is in S, then $s \overset{a}{\Longrightarrow} \hat{s}$.

  —**Condition** (ii):

  Suppose $s = \theta_1 \dots \theta_k\theta_{k+1} \dots \theta_\ell$ is a state in S where $\theta_i = (n_i, \Gamma_i, W_i, n_i')$ for $i = 1, 2, \dots, \ell$. Suppose $W_k \neq \perp$. Let $W_k = (R_k, \lambda_k, \{\sqsubseteq_p^k\}, \ll_k, \leq_k)$. Let

e be a minimal event in W_k and $a = \eta_k(e)$. Suppose, for every $e' \in R_k$ with $e \leq e'$, it is the case that $\eta_k(e')\ I_p\ \gamma$ for any $\gamma \in \Sigma_p \cap (\bigcup_{k+1 \leq i \leq \ell} \Gamma_i)$, where $p = loc(e')$. Let $\hat{\theta} = (n_k, \Gamma_k \cup \{a\}, \widehat{W}, n'_k)$, where $\widehat{W}$ is as follows: Let $\widehat{R} = R_k - \{e\}$. If $\widehat{R}$ is nonempty, then $\widehat{W}$ is the restriction of W to $\widehat{R}$; otherwise $\widehat{W} = \perp$. Let $\hat{s} = \theta_1 \dots \theta_{k-1} \hat{\theta} \theta_{k+1} \dots \theta_\ell$. Then we have $s \stackrel{a}{\Longrightarrow} \hat{s}$. (Note that $\hat{s}$ is guaranteed to be in S.)

We have now completed the construction of $\mathcal{A}_H$. It remains to show that $\mathcal{A}_H$ recognizes $Lin(H)$.

Lemma 3. *Let $\sigma \in \Sigma^\star$. Then σ is accepted by $\mathcal{A}_H$ iff σ is in $Lin(H)$.*

Proof. Let $\sigma = a_1 a_2 \dots a_k$. Suppose σ is in $Lin(H)$. Let $\rho = n_0 \xrightarrow{B_1} \dots \xrightarrow{B_\ell} n_\ell$ be an accepting path in H such that σ is a linearization of $\circledcirc(\rho)$. Hence we may suppose that $\circledcirc(\rho) = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$ where $E = \{e_1, e_2, \dots, e_k\}$ and $\lambda(e_i) = a_i$ for $i = 1, \dots, k$. And $e_i \leq e_j$ implies $i \leq j$ for any i, j in $\{1, \dots, k\}$. Consider the configurations $C_i = \{e_1, e_2, \dots, e_i\}$ for $i = 1, \dots, k$. For each C_i, we can associate a state s_i in $\mathcal{A}_H$ as follows. Consider a fixed C_i. Let $\rho = \dots \varrho_1 \dots \varrho_2 \dots \varrho_h \dots$ where ϱ_1, ϱ_2, …, ϱ_h are the C_i-subpaths of ρ. Then we set $s_i = \theta_1 \dots \theta_h$ where $\theta_j = (n_j, \Gamma_j, W_j, n'_j)$ with n_j being the starting node of ϱ_j, and Γ_j the collection of all $\lambda(e)$ for all events e that are in both $\circledcirc(\varrho_j)$ and C_i. Let R_j be the C_i-residue of ϱ_j. If R_j is nonempty, W_j is the causal MSC $(R_j, \lambda_j, \{\sqsubseteq_p^j\}, \ll_j, \leq_j)$ where λ_j is the restriction of λ to R_j; $\sqsubseteq_p^j$ is the restriction of $\sqsubseteq_p$ to those events in R_j that belong to process p, for each p; and $\ll_j$ the restriction of $\ll$ to R_j. If R_j is empty, then set $W_j = \perp$. Finally, n'_j is the ending node of ϱ_j.

Now it is routine (though tedious) to verify that $\varepsilon \stackrel{a_1}{\Longrightarrow} s_1 \dots s_{k-1} \stackrel{a_k}{\Longrightarrow} s_k$ is an accepting run of $\mathcal{A}_H$. Conversely, given an accepting run of $\mathcal{A}_H$ over σ, it is straightforward to build a corresponding accepting path of H.

□

With Lemma 3, we establish Theorem 1. As for complexity, the bound on the number of states in $\mathcal{A}_H$ stated in Theorem 1 is clear from the construction of $\mathcal{A}_H$.

3.3 Inclusion and intersection non-emptiness of causal HMSCs

As the linearization languages of regular causal HMSCs are regular, verification for regular causal HMSCs can be effectively solved. It is natural to ask whether we can still obtain positive results of verification beyond the subclass of regular causal HMSCs. As for HMSCs, one can show that for a suitable choice of K, the set of K-bounded linearizations of any globally cooperative HMSC is regular, and this is sufficient for effective verification [5]. Unfortunately, this result uses Kuske's encoding [11] into traces that is based on the existence of an (existential) bound on communication. Consequently, this technique does not apply to

globablly cooperative causal HMSCs, as the visual language of a causal HMSC needs not be existentially bounded. For instance, consider the causal HMSC H of Figure 7. It is globally cooperative (but not regular), and its visual language contains MSCs shown in the right part of Figure 7: 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.

It is known that problems of inclusion, intersection non-emptiness and equality of visual languages of HMSCs are undecidable [13]. Clearly, these undecidability results also apply to causal HMSCs. In [13], decidability results for inclusion and intersection non-emptiness of globally cooperative HMSCs are established. Our goal here is to extend these results to globally cooperative causal HMSCs.

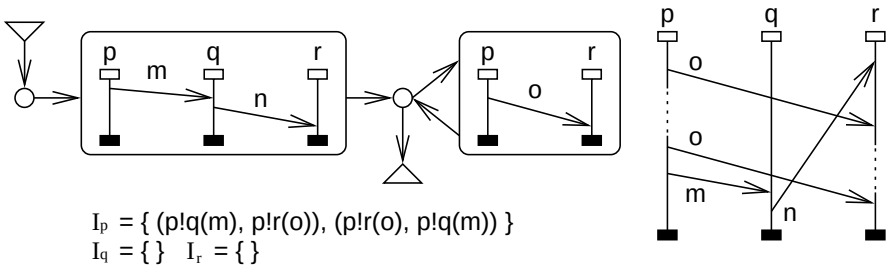


Fig. 7. A globally-cooperative causal HMSC that is not existentially bounded

We shall adapt the notion of atoms [1, 9] and the techniques from [6]. Let us first introduce a notion of decomposition of causal MSCs into basic parts.

Definition 4. *A causal MSC B is a* basic part *(w.r.t. the trace alphabets $\{(\Sigma_p, I_p)\}$) if there do not exist causal MSCs B_1, B_2 such that $B = B_1 \circledcirc B_2$.*

Note that we require that the set of events of a causal MSC is not empty. Now for a causal MSC B, we define a *decomposition* of B to be a sequence $B_1 \cdots B_\ell$ of basic parts such that $B = B_1 \circledcirc \cdots \circledcirc B_\ell$. For a set $\mathcal{B}$ of basic parts, we associate a trace alphabet $(\mathcal{B}, I_\mathcal{B})$ (w.r.t. the trace alphabets $\{(\Sigma_p, I_p)\}$) where $I_\mathcal{B}$ is given by: $B\ I_\mathcal{B}\ B'$ iff for every p, for every $\alpha \in Alph_p(B)$, for every $\alpha' \in Alph_p(B')$, it is the case that $\alpha\ I_p\ \alpha'$. We let $\sim_\mathcal{B}$ be the corresponding trace equivalence relation and denote the trace containing a sequence $u = B_1 \ldots . B_\ell$ in $\mathcal{B}^\star$ by $[u]_\mathcal{B}$ (or simply $[u]$). For a language $L \subseteq \mathcal{B}^\star$, we define its trace closure $[L]_\mathcal{B} = \bigcup\{[u]_\mathcal{B} \mid u \in L\}$.

Proposition 2. *For a given causal MSC B, we can effectively construct the smallest finite set of basic parts, denoted $Basic(B)$, such that every decomposition of B is in $Basic(B)^\star$. Further, the set of decompositions of B forms a trace of $(Basic(B), I_{Basic(B)})$.*

Proof. We describe the construction of $Basic(B)$, which is analogous to the technique in [9]. The claim will then be clear from this construction. Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$. We consider the undirected graph (E, R), where R is the symmetric closure of $\ll \cup \big(\bigcup_{p \in \mathcal{P}} R'_p \cup R''_p\big)$. Here,

$$R'_p = \{(e, e') \in E_p \times E_p \mid e \sqsubseteq_p e' \text{ and } \lambda(e)\ I_p\ \lambda(e')\}\ ,$$
$$R''_p = \{(e, e') \in E_p \times E_p \mid e \not\sqsubseteq_p e' \text{ and } e' \not\sqsubseteq_p e \text{ and } \lambda(e)\ D_p\ \lambda(e')\}\ .$$

For each connected component of (E, R) with E' being its set of vertices, we associate a basic part $(E', \lambda', \{\sqsubseteq'_p\}, \ll')$, where λ' is the restriction of λ to E', $\sqsubseteq'_p$ is the restriction of $\sqsubseteq'_p$ to E', and $\ll'$ is the restriction of $\ll$ to E'. The set $Basic(B)$ is then the collection of basic parts obtained from the connected components of (E, R). Note that $Basic(B)$ can be constructed in quadratic time. □

In view of Proposition 2, we assume through the rest of this section that every transition of a causal HMSC H is labelled by a basic part. Clearly this incurs no loss of generality, since we can simply decompose each causal MSC in H into basic parts and decompose any transition of H into a sequence of transitions labeled by these basic parts. Given a causal HMSC H, we let $Basic(H)$ be the set of basic parts labelling transitions of H. Proposition 2 implies that a causal MSC is uniquely defined by its basic part decomposition. Then instead of the linearization language we can use the *basic part language* of H, denoted by $BP(H) = \{B_1 \dots B_\ell \in Basic(H)^\star \mid B_1 \circledcirc \dots \circledcirc B_\ell \in CaMSC(H)\}$. Notice that $BP(H) = [BP(H)]$ by Proposition 2, that is, $BP(H)$ is closed by commutation. We can also view H as a finite state automaton over the alphabet $Basic(H)$, and denote by $\mathcal{L}_{Basic}(H) = \{B_1 \cdots B_\ell \in Basic(H)^\star \mid n_0 \xrightarrow{B_1} n_1 \cdots \xrightarrow{B_\ell} n_\ell$ is an accepting path of H.$\}$ its associated (regular) language. We now relate $BP(H)$ and $\mathcal{L}_{Basic}(H)$.

Proposition 3. *Let H be a causal HMSC. Then $BP(H) = [\mathcal{L}_{Basic}(H)]$.*

Proof. Immediate from Proposition 2. □

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 Proposition 3. In general, we cannot effectively compute this language. However if H is globally cooperative, then $[\mathcal{L}_{Basic}(H)]$ is regular and a finite state automaton recognizing $[\mathcal{L}_{Basic}(H)]$ can be effectively constructed [4, 13]. Considering globally cooperative causal HMSCs as finite state automata over basic parts, we can apply [13] to obtain the following decidability and complexity results:

Theorem 2. *Let H, H' be causal HMSCs over the* same *family of trace alphabets $\{(\Sigma_p, I_p)\}$. Suppose H' is globally cooperative. Then we can build a finite state automaton $\mathcal{A}'$ over $Basic(H')$ such that $\mathcal{L}_{Basic}(A') = [\mathcal{L}_{Basic}(H')]$. And $\mathcal{A}'$ has at most $2^{O(n \cdot b)}$ states, where n is the number of nodes in H and b is the number of basic parts in $Basic(H)$. Consequently, the following problems are decidable:*

(i) Is $CaMSC(H) \subseteq CaMSC(H')$*?*
(ii) Is $CaMSC(H) \cap CaMSC(H') = \emptyset$*?*

Furthermore, the complexity of (i) is PSPACE-complete and that of (ii) is EXPSPACE-complete.

The above theorem shows that we can model check a causal HMSC against a globally cooperative causal HMSC specification. Note that we can only apply Theorem 2 to two causal HMSCs over the *same* family of trace alphabets. If the causal HMSCs H, H' in theorem 2 satisfy the additional condition that every causal MSCs labeling the transitions of H and H' respects $\{(\Sigma_p, I_p)\}$, then we can compare the visual languages $Vis(H)$ and $Vis(H')$, thanks to Proposition 1. On the other hand, when two causal HMSCs are defined with different families of trace alphabets, the only possible comparison between them seems to be on their linearization languages. Consequently, we would need to work with regular causal HMSCs.

4 Window-bounded 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 in 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.

Definition 5. *Let* $M = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ *be an MSC For a message* (e, f) *in* M*, that is,* $(e, f) \in \ll$*, we define the* window *of* (e, f)*, denoted* $W_M(e, f)$*,*

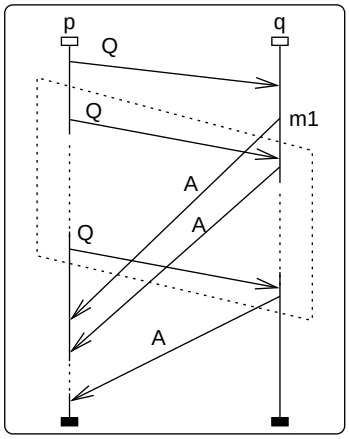


Fig. 8. Window of message m_1

as the set of messages $\{(e', f') \in \ll | \ loc(\lambda(e')) = loc(\lambda(f))$ *and* $loc(\lambda(f')) = loc(\lambda(e))$ *and* $e \leq f'$ *and* $e' \leq f\}$.

We say that a causal HMSC H is K*-window-bounded* iff for every $M \in Vis(H)$ and for every message (e, f) of M, it is the case that $|W_M(e, f)| \leq K$. H is said to be window-bounded iff H is K-window-bounded for some K.

Figure 8 illustrates notion of window, 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 effectively check whether a causal HMSC is window bounded. It builds a finite state automaton whose states remember the labels of events that must appear in the *future* of messages (respectively in the *past*) in any MSC of $Vis(H)$.

Formally, for a causal MSC $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ and $(e, f) \in \ll$ a message of B, we define the future and past of (e, f) in B as follows:

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

In Figure 8, $Past_B(m_1) = \{p!q(Q), q?p(Q), q!p(A)\}$.

Proposition 4. *Let* $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ *and* $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ *be two causal MSCs, and let* $m \in \ll$ *be a message of* B. *Then we have:*

$$Future_{B \circledcirc B'}(m) = Future_B(m) \cup \{a' \in \Sigma \mid \exists x, y \in E'$$
$$\exists a \in Future_B(m) \ s.t. \ \lambda(y) = a' \wedge x \leq' y \wedge a \ D_{loc(a)} \ \lambda(x)\}$$

Proof. Follows from definition. □

Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ be a causal HMSC. Consider a path ρ of H with $\circledcirc(\rho) = B_1 \circledcirc \cdots \circledcirc B_\ell$ and a message m in B_1. Then Proposition 4 implies the sequence of sets $\mathit{Future}_{B_1}(m)$, $\mathit{Future}_{B_1 \circledcirc B_2}(m)$, ..., $\mathit{Future}_{B_1 \circledcirc \cdots \circledcirc B_\ell}(m)$ is non-decreasing. Furthermore, these sets can be computed on the fly, that is with a finite state automaton. Similar arguments hold for the past sets. Now consider a message (e, f) in a causal MSC B labelling some transition t of H. With the above observation on *Future* and *Past*, we can show that, if there is a bound $K_{(e,f)}$ such that the window of a message (e, f) in the causal MSC generated by any path containing t is bounded by $K_{(e,f)}$, then $K_{(e,f)}$ is at most $b|N|(|\Sigma| + 1)$ where $b = \max\{|B| \mid B \in \mathcal{B}\}$. Further, we can effectively determine whether such a bound $K_{(e,f)}$ exists by constructing a finite state automaton whose states memorize the future and past of (e, f). Thus we have the following:

Theorem 3. *Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ be a causal HMSC. Then we have:*

(i) If H is window-bounded, then H is K-window-bounded, where K is at most $b|N|(|\Sigma| + 1)$ with $b = \max\{|B| \mid B \in \mathcal{B}\}$.

(ii) Further, we can effectively determine whether H is window-bounded in time $O(s \cdot |N|^2 \cdot 2^{|\Sigma|})$, where s is the sum of the sizes of causal MSCs in $\mathcal{B}$.

The rest of this subsection is devoted to the proof of Theorem 3. We fix H as in Theorem 3.

Proof (of Theorem 3(i)). Suppose the contrary. That is, H is window-bounded, but not k-window-bounded, where $k = |\mathcal{B}||N|(|\Sigma| + 1)$. Let $B \in \mathcal{B}$ be a causal MSC in $\mathit{CaMSC}(H)$, and let the pair (e, f) be a message of B. Let ρ be a path of H, and $V \in \mathit{Vis}(\circledcirc(\rho))$ be an MSC such that the message (e, f) is crossed by $k + 1$ messages in V. Then, there exists a causal MSC $B' \in \mathcal{B}$ containing a message m' that crosses (e, f) at least $2|N|(|\Sigma| + 1)$ times in V. Without loss of generality, we can consider that B' is repeated at least $|N|(|\Sigma|+1)$ times after B in ρ (symmetric proof holds when considering repetitions of B' occurring before B). That is, ρ has the form

$$\cdots \xrightarrow{B} \cdots n_1 \xrightarrow{B_1} \cdots n_2 \xrightarrow{B_2} \cdots n_t \xrightarrow{B_t} \cdots$$

where $t = |N|(|\Sigma| + 1)$ and $B' = B_1 = B_2 = \ldots = B_t$. Hence, we can find $j_1 \ldots j_{|\Sigma|+1}$ such that $n_{j_1} = \ldots = n_{j_{|\Sigma|+1}}$. Consider the sequence of sets $F_i = \mathit{Future}_{B_1 \circledcirc \ldots \circledcirc B_{j_i}}(e, f)$, $i = 1, 2, \ldots, |\Sigma| + 1$. Each F_i is a subset of Σ and the sequence F_1, F_2, ..., $F_{|\Sigma|+1}$ is non-decreasing. Hence, we can find $\ell \leq |\Sigma|$ such that $F_\ell = F_{\ell+1}$ which does not contain labels of m'. This means that path ρ', which is computed from path ρ by repeating twice the loop between n_{j_ℓ} and $n_{j_{\ell+1}}$, has the same *Future*, and have at least one more m' which can cross m. Thus, we can exhibit a new execution $V' \in \mathit{Vis}(\circledcirc(\rho'))$ such that (e, f) is crossed by at least $k + 2$ messages. As we can iterate this construction, it means that H is not window-bounded. □

We next establish Theorem 3(ii), We shall show that one can decide in an efficient way whether the maximal window bound for a given message m can

be reached, by constructing a finite state automaton that memorizes $\mathit{Future}(m)$ and $\mathit{Past}(m)$.

For a given causal HMSC $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ and a message (e, f), we build the following automaton: $\mathcal{A}_{(e,f)} = (Q, Q_{in}, \mathcal{B}, Q_{fi}, \delta)$ where:

- $Q = N \times 2^{\Sigma}$.
- $Q_{in} = \{(n, \emptyset) \mid n \in N_{in}\}$.
- $(n, X) \in Q_{fi}$ if and only if $n \in N_{fi}$.
- $\delta \subseteq Q \times \mathcal{B} \times Q$ is the least relation such that:
  - $\big((n, \emptyset), B, (n', \emptyset)\big) \in \delta$ if $n \xrightarrow{B} n'$
  - $\big((n, \emptyset), B, (n', \mathit{Future}_B(e, f))\big) \in \delta$ if $n \xrightarrow{B} n'$ and (e, f) belongs to B.
  - $\big((n, X), B, (n', X')\big) \in \delta$, where $B = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$, if $n \xrightarrow{B} n'$, and $X \neq \emptyset$, and $X' = X \cup \{a' \in \Sigma \mid \exists x, y \in E, \exists a \in \mathit{Future}_B(e, f), \lambda(y) = a' \wedge x \leq y \wedge a \; D \; \lambda(x)\}$.

We also build an automaton that computes $\mathit{Past}(e, f)$, by a backward search in the causal HMSC H. More precisely,
$\mathcal{A}'_{(e,f)} = (Q', Q'_{in}, \mathcal{B}, Q'_{fi}, \delta')$ where

- $Q' = N \times 2^{\Sigma}$
- $Q'_{in} = N_{fi} \times \{\emptyset\}$
- $Q'_{fi} = N_{in} \times 2^{\Sigma}$
- $\delta' \subseteq Q \times \mathcal{B} \times Q$ is the least relation such that:
  - $\big((n, \emptyset), B, (n', \emptyset)\big) \in \delta$ if $n' \xrightarrow{B} n$
  - $\big((n, \emptyset), B, (n', \mathit{Past}_B(e, f))\big) \in \delta$ if $n' \xrightarrow{B} n$ and (e, f) belongs to B.
  - $\big(n, X), B, (n', X')\big) \in \delta$, where $B = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$, if $n' \xrightarrow{B} n$, and $X \neq \emptyset$, and $X' = X \cup \{a' \in \Sigma \mid \exists x, y \in E, \exists a \in \mathit{Past}_B(e, f), \lambda(y) = a' \wedge y \leq x \wedge a \; D \; \lambda(x)\}$

More intuitively, a state $q = (n, X)$ in $\mathcal{A}_{(e,f)}$ represents a possible set X of labels in $\mathit{Future}_{\circledcirc(\rho)}(e, f)$ for some path ρ that ends at node n in H, and contains a message (e, f). Slightly abusing the notation, we will denote by $\mathit{Future}(q)$ the set X. The second rule in the transition relation δ (resp. δ') is important, as it allows to chose nondeterministically an occurrence of (e, f), and to start memorizing the labels appearing in its future (resp. in its past). Note that in any strongly connected subset $C = \{q_1, \ldots, q_k\}$ of $\mathcal{A}_{(e,f)}$ (respectively $\mathcal{A}'_{(e,f)}$), $\mathit{Future}(q_1) = \mathit{Future}(q_2) = \cdots = \mathit{Future}(q_k)$ (resp. $\mathit{Past}(q_1) = \mathit{Past}(q_2) = \cdots = \mathit{Past}(q_k)$). Hence, we will denote by $\mathit{Future}(C)$ (resp. $\mathit{Past}(C)$) the set of observed labels on any state of C.

We observe the following properties of the finite state automata $\mathcal{A}_{(e,f)}$ and $\mathcal{A}'_{(e,f)}$.

Lemma 4. *Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ be a causal HMSC. Let B be a causal MSC in $\mathcal{B}$ and (e, f) a message in B with the label of e being $p!q(m)$. Consider the finite state automata $\mathcal{A}_{(e,f)}$ and $\mathcal{A}'_{(e,f)}$ as constructed above. Then, H is window-bounded iff both of the following conditions hold:*

– *There does* not *exist a strongly connected component* C *in* $\mathcal{A}_{(e,f)}$ *and a letter* $q!p(m') \in \Sigma$ *such that* $q!p(m')$ *is in* $Alph(B) - Future(C)$ *for some causal MSC* B *labelling a transition in* C.
– *There does* not *exist a strongly connected component* C *in* $\mathcal{A}'_{(e,f)}$ *and a letter* $q!p(m') \in \Sigma$ *such that* $q!p(m')$ *is in* $Alph(B) - Past(C)$ *for some causal MSC* B *labelling a transition in* C.

Proof. One direction is straightforward. If any of these strongly connected components exists (either before or after m), then there is an unbounded number of path generating an unbounded number of occurrences of $q!p(m')$ that are not causally related to m. Hence, for each of these path, there is a visual extension where all m' generated by occurrences of the cycle cross m, and the window size of m is not bounded. The other direction is a direct consequence of Theorem 3(i). □

Thus Theorem 3(ii) follows from Lemma 4. It remains to establish the complexity claim in Theorem 3(ii). The automaton $\mathcal{A}_{(e,f)}$ has at most $|N| \times 2^{|\Sigma|}$ states, and we have to analyze strongly connected components of $\mathcal{A}_{(e,f)}$. However, as noticed before, every strongly connected component of $\mathcal{A}_{(e,f)}$ enjoys the property to have a second component which is constant. Hence we need to test the property only for *maximal* strongly connected components. Indeed, if C is a strongly connected component of $\mathcal{A}_{(e,f)}$ such that $q!p(m')$ is the label of an event in a causal MSC labeling a transition of C but that is not in $Future(C)$, then we can consider the maximal strongly connected component D of $\mathcal{A}_{(e,f)}$ containing C (it exists since the union of two non disjoint strongly connected components is again a strongly connected component). Since D it is a strongly connected component, its second component $Future(D)$ is constant, hence $Future(D) = Future(C)$. Since $C \subseteq D$, we have that $q!p(m')$ is a label of an event of D and is not in $Future(C) = Future(D)$.

Using Tarjan's algorithm [16], we can compute in quadratic time the partition of $\mathcal{A}_{(e,f)}$ into maximal strongly connected components (for each set $X \subseteq 2^{\Sigma}$, we partition the subpart of $\mathcal{A}_{(e,f)}$ with a constant second component being X). Then for each maximal strongly connected component (C, X), it suffices to compute $\lambda(C)$ and to compare it with X, which is linear in n. Hence, the overall complexity of the algorithm is in $O(|N|^2 2^{|\Sigma|})$. Then, we construct these automaton for each message of each label of H.

5 Relationship with Other Scenario Models

We compare here the expressive power of other HMSC-based scenario languages with causal HMSCs in terms of their visual languages. We consider first 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 [6], which ensure that for a suitable choice of K, the set of K-bounded linearizations form a regular set. By definition, causal

HMSCs, regular causal HMSCs and globally-cooperative causal HMSCs extend respectively HMSCs, regular HMSCs and globally-cooperative HMSCs.

Figure 7 shows a globally-cooperative causal HMSC which is not in the subclass of regular causal HMSCs. Thus, regular causal HMSCs form a strict subclass of globally-cooperative causal HMSCs. Trivially, globally-cooperative causal HMSCs are a strict subclass of causal HMSCs. Figure 5 displays a regular causal HMSC whose visual language is not finitely generated. It follows that (regular/globally-cooperative) causal HMSCs are strictly more powerful than (regular/globally-cooperative) HMSCs.

Another extension of HMSCs is *Compositional* HMSCs [7], or CHMSCs for short. CHMSCs generalize HMSCs by allow dangling message-sending and message-reception events, i.e. where the message pairing relation $\ll$ is only a partial non-surjective mapping contained in $E_! \times E_?$. The concatenation of two Compositional MSCs $M \circ M'$ performs the instance-wise concatenation as for MSCs, and computes a new message pairing relation $\ll''$ defined over $(E_! \cup E'_!) \times (E_? \cup E'_?)$ extending $\ll \cup \ll'$, and preserving the FIFO ordering of messages of the same content (actually, in the definition of [7], 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+1} is in $Vis(G)$, hence $M_{k+1} \in Vis(\circledcirc(\rho))$ for some accepting path ρ of G. Let $N_1, \ldots, N_\ell$ be causal MSCs along ρ, where $\ell \geq 2$ because of the size k. It also means that there exist $N'_1 \in Vis(N_1)$, $\ldots$, $N'_\ell \in Vis(N_\ell)$ such that $N'_1 \circ \cdots \circ N'_m \in Vis(G)$. Thus, $N_1 \circ \cdots \circ N_\ell = M_j$ for some j, a contradiction since M_j is a basic part (i.e. cannot be the concatenation of two MSCs). 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). Also by the results in [8], it is immediate that the class of visual languages of regular compositional HMSCs captures all the MSC languages that have a regular set of linearizations. Hence the class of regular causal HMSCs is included into the class of regular compositional HMSCs. Last, we already know with Figure 7 that globally-cooperative causal HMSCs are not necessarily existentially bounded, hence they are not included into safe

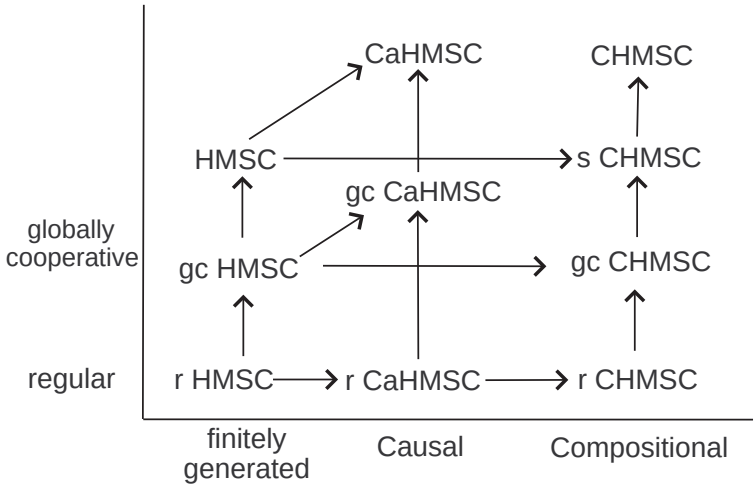


Fig. 9. Comparison of Scenario languages

Compositional HMSC. Furthermore, globally-cooperative causal HMSCs are not included into CHMSCs because the former can generate MSCs that are *not* FIFO.

The relationships among these scenario models are summarized by Figure 9, where arrows denote *strict* inclusion of visual languages. Two classes are incomparable if 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 HMSC called causal HMSC that allows the definition of braids, such as those appearing in sliding window 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 scenario collections that are not existentially bounded. Yet, decidability results of model checking can be obtained for this class.

An interesting open problem is deciding whether the visual language of a causal HMSC is finitely generated. Yet another interesting issue is to consider the class of causal HMSCs whose visual languages are window-bounded. The set of behaviours 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, D. Kuske, and A. Muscholl. A Kleene theorem and model checking for a class of communicating automata. *Information and Computation.*, 204(6):920–956, 2006.
6. 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.
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.
16. R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal of Computing*, 1(2), 1972.

Appendix

We demonstrate the modelling power of causal HMSCs with an example in Figure 10. The causal HMSC H models a subset of the possible scenarios exhibited by the alternating bit protocol. The letters $d0$ indicates p sending to q a data packet with control bit 0, while $a0$ represents q sending to p an packet with acknowledge bit 0. The internal actions $p(b0)$, $q(c0)$ (displayed simply as $b0$,$c0$ in Figure 10) signifies that p setting its control bit to 0, and

that q verifying a data packet with control bit 0 is correctly transmitted. The meanings of $a1, b1, c1, d1$ are analogous. The independence relation I_p is given by: $p(b0)\ I_p\ p?q(a1)$, $p(b1)\ I_p\ p?q(a0)$. And I_q is given by: $q(c0)\ I_q\ q?p(d0)$, $q(c1)\ I_q\ q?p(d1)$. We note that the causal MSCs *Send0*, *Send1*, *Ack0*, *Ack1*, do not respect the trace alphabets $\{(\Sigma_p, I_p), (\Sigma_q, I_q)\}$. The left part of Figure 11 shows a causal MSC in $CaMSC(H)$, and its right part displays an MSC in $Vis(H)$. To reduce clutter, some ordering between events of the same label are omitted in the left part of Figure 11.

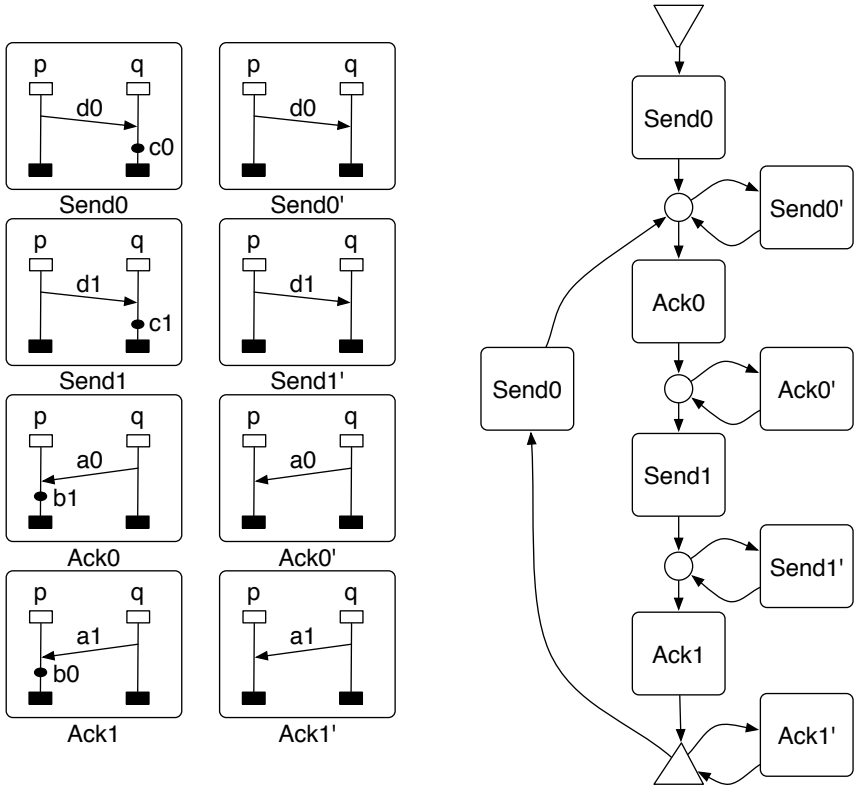


Fig. 10. Modelling the alternating bit protocol

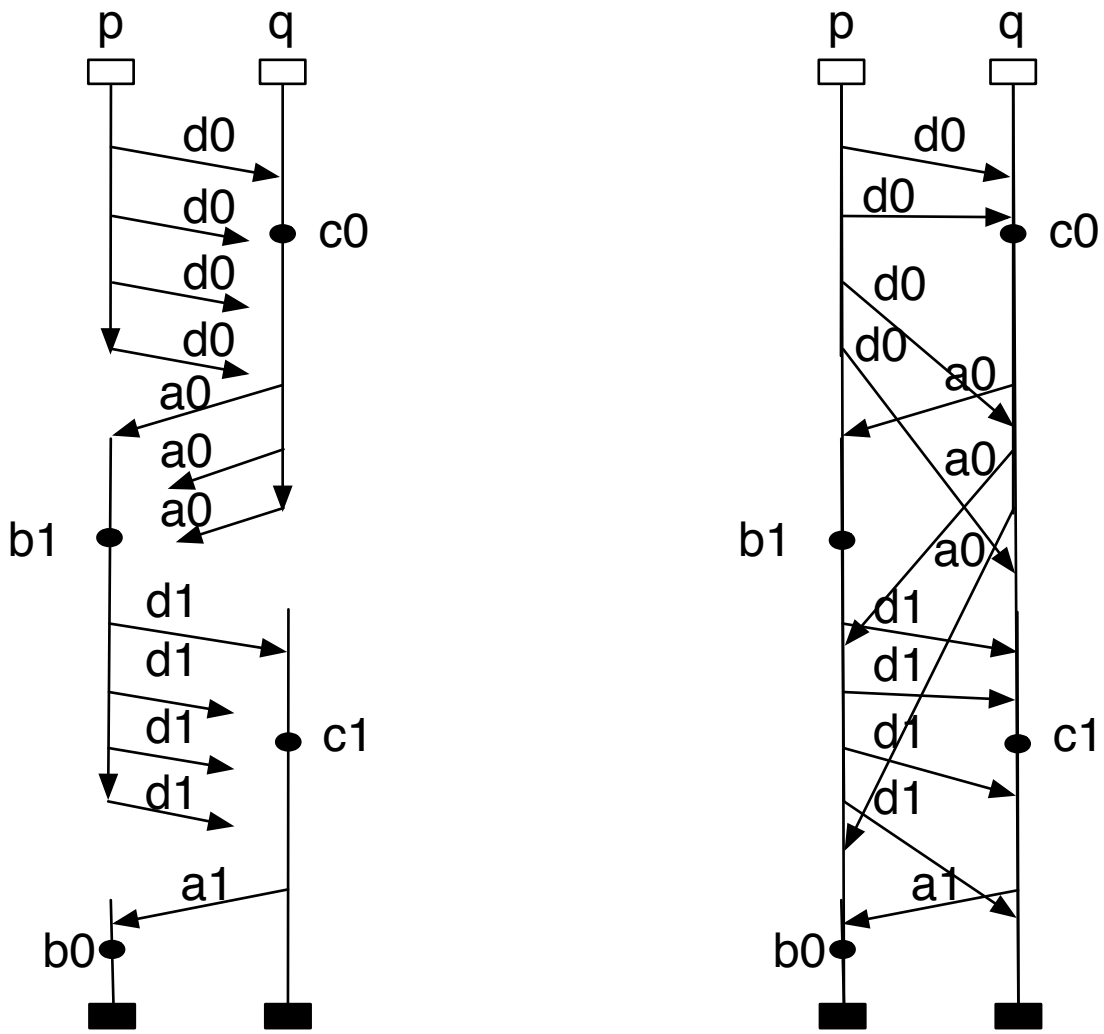


Fig. 11. Modelling the alternating bit protocol