

Products of Message Sequence Charts*

Philippe Darondeau, Blaise Genest, and Loïc Hélouët

IRISA, campus de Beaulieu, F-35042 Rennes Cedex

Abstract. An effective way to assemble partial views of a distributed system is to compute their product. Given two Message Sequence Graphs, we address the problem of computing a Message Sequence Graph that generates the product of their languages, when possible. Since all MSCs generated by a Message Sequence Graph G may be run within fixed bounds on the message channels (that is, G is existentially bounded), a subproblem is to decide whether the considered product is existentially bounded. We show that this question is undecidable, but turns co-NP-complete in the restricted case where all synchronizations belong to the same process. This is the first positive result on the decision of existential boundedness. We propose sufficient conditions under which a Message Sequence Graph representing the product can be constructed.

1 Introduction

Scenario languages, and in particular Message Sequence Charts (MSCs) have met a considerable interest over the last decade in both academia and industry. MSCs allow for the compact description of distributed systems executions, and their visual aspect made them popular in the engineering community. Our experience with industry (France-Telecom) showed us that MSCs are most often used there together with extensions such as optional parts (that is choice) and (weak) concatenation, while iteration is left implicit. (Compositional) Message Sequence Graphs ((C)MSC-graphs) is the academic framework in which choice, weak concatenation and iteration of MSCs are formalized. For a recent survey of Message Sequence Graphs, we refer the reader to [6,9]. A challenging problem is to automatically implement MSC-languages (that is, sets of MSCs) given by (C)MSC-graphs. Apart from the restricted case of Local Choice (C)MSC-graphs [8,7], this problem has received no satisfactory solution, since either deadlocks arise from the implementation [14,4], or implementation may exhibit unspecified behaviors [2]. A further challenge is to help designing (C)MSC-graphs for complex systems, while keeping analysis and implementability decidable. Systems often result from assembling modules, reflecting different aspects. A possible way to help the modular modeling of systems into (C)MSC-graphs is thus to provide a product operator. A first attempt in this direction is [10], where the amalgamation allows the designer to merge 2 nodes of 2 MSG-graphs but not their paths. We feel that a more flexible operation, defined on MSC languages and therefore independent from MSC block decompositions, is needed.

* Work supported by France Telecom R&D (CRE CO2) and ANR project DOTS.

Shuffling the linearizations of the languages of two (C)MSC-graphs is not the right product. On the one hand, such shuffling kills *existential bounds* [11], i.e., there is no upper bound on the size of the message channels within which all MSCs in the shuffled language *can* be run. Existential bounds are an important feature of (safe C)MSC-graphs which allow their analysis. On the other hand, two states of the (C)MSC-graphs (one for each module) may represent incompatible aspects. Hence, one needs some synchronization to control the product operation, in order to avoid incompatibilities and non existentially bounded behaviors. Control may be introduced with synchronization points: one module waits at a synchronization point until the other module reaches a compatible synchronization point, and then both can proceed. Synchronizations may be defined either per process or per state of the (C)MSC-graphs. State oriented synchronization conflicts with weak concatenation since it means that all processes of the *same* module pass simultaneously the synchronization barrier, which diverges strongly from the semantics of (C)MSC-graphs. Second, it harms implementability, since state-synchronized products of implementable (C)MSC-graphs may not be implementable. We therefore choose to define synchronizations per process, by means of shared local events identified by names common to both MSCs. Formally, we define thus a mixed product of MSCs that amounts to shuffling their respective events on each process, simultaneously and independently, except for the shared events that are not interleaved but coalesced. One appealing property of this definition of product is that the product of two implementable (C)MSC-graphs is also implementable (albeit with possible deadlocks), since it suffices to take the product of the implementations processwise, coalescing shared events.

In order to be represented as a (safe C)MSC-graph, an MSC language needs to be existentially bounded. So far, no algorithm is known to check the existential boundedness of an MSC language in a non-trivial case (e.g., existential boundedness is undecidable even for deterministic deadlock-free Communicating Finite State Machines, see <http://perso.crans.org/~genest/GKM07.pdf>). This is the challenging problem studied in this paper. We show that checking existential boundedness of the product of two (safe C)MSC-graphs is in general undecidable, as one expects. Surprisingly, if all shared events (synchronizations) belong to the same process, then this question becomes decidable. Once a product is known to be existentially bounded, results [12,4] on representative linearizations can be used. Namely, languages of MSCs defined by the globally cooperative subclass of safe CMSC-graphs have regular sets of linear representatives, where the regular representations can be computed from the CMSC-graphs and conversely. Thus, given two globally cooperative CMSC-graphs such that their product is existentially bounded, this product can be represented with a globally cooperative CMSC-graph. The authors of [4] ignore the contents of messages in the definition of MSCs. We consider messages with contents, and adapt the FIFO requirement of [4] to both weak ([2,13]) and strong FIFO ([1]). We recast the correspondence established in [4] into these different frameworks, and compare the complexity and decidability of these two semantics.

The paper is organized as follows. Section 2 recalls the background of MSCs and MSC-graphs. Section 3 introduces the product of MSC-languages. Section 4 recalls the definition of existential channel bounds for MSC-languages. It is shown in Sections 5 and 6 that one can, in general, not check the existential boundedness of the product of two existentially bounded MSC-languages, whereas this problem is co-NP-complete (weak FIFO) or PSPACE (strong FIFO) when the synchronizations are attached to a single process. Section 7 defines for that special case an operation of product on CMSC-graphs. Many proofs are skipped or only briefly sketched by lack of space, but they are available in the full version of the paper, available at <http://perso.crans.org/~genest/DGH07.pdf>.

2 Background

To begin with, we recall the usual definition of *compositional Message Sequence Charts* (CMSCs for short), which describe executions of communication protocols, and of CMSC-graphs, which are generators of CMSC sets. Let \mathcal{P} , \mathcal{M} , and \mathcal{A} be fixed finite sets of *processes*, *messages* and *actions*, respectively. Processes may perform *send* events \mathcal{S} , *receive* events \mathcal{R} and *internal* events \mathcal{I} . That is, the set of types of events of an MSC is $\mathcal{E} = \mathcal{S} \cup \mathcal{R} \cup \mathcal{I}$ where $\mathcal{S} = \{p!q(m) \mid p, q \in \mathcal{P}, p \neq q, m \in \mathcal{M}\}$, $\mathcal{R} = \{p?q(m) \mid p, q \in \mathcal{P}, p \neq q, m \in \mathcal{M}\}$, and $\mathcal{I} = \{p(a) \mid p \in \mathcal{P}, a \in \mathcal{A}\}$. For each $p \in \mathcal{P}$, we let $\mathcal{E}_p = \mathcal{S}_p \cup \mathcal{R}_p \cup \mathcal{I}_p$ where \mathcal{S}_p , \mathcal{R}_p , and \mathcal{I}_p are the restrictions of \mathcal{S} , \mathcal{R} , and \mathcal{I} , respectively, to the considered process p (e.g., $p?q(m) \in \mathcal{S}_p$). We define now MSCs over \mathcal{E} .

Definition 1. A *compositional Message Sequence Chart* M is a tuple $M = (E, \lambda, \mu, (<_p)_{p \in \mathcal{P}})$ where

- E is a finite set of events, with types $\lambda(e)$ given by a labeling map $\lambda : E \rightarrow \mathcal{E}$,
- for each $p \in \mathcal{P}$, $<_p$ is a total order on $E_p = \lambda^{-1}(\mathcal{E}_p)$,
- $\mu : E \rightarrow E$ is a partially defined, injective mapping,
- if $\mu(e_1) = e_2$ then $\lambda(e_1) = p!q(m)$ and $\lambda(e_2) = q?p(m)$ for some p, q and m ,
- [**weak FIFO**] if $e_1 <_p e'_1$, $\lambda(e_1) = \lambda(e'_1) = p!q(m)$ and $\mu(e'_1)$ is defined, then $\mu(e_1) <_q \mu(e'_1)$ (in particular, $\mu(e_1)$ is defined).
- the union $<$ of $\cup_{p \in \mathcal{P}} <_p$ and $\cup_{e \in E} \{(e, \mu(e))\}$ is an acyclic relation.
- M is an MSC if the partial map μ is a bijection between $\lambda^{-1}(\mathcal{S})$ and $\lambda^{-1}(\mathcal{R})$.

Def. 1 extends the original definition of [5] (see also [4]) by considering messages with non trivial contents. There are then two alternatives to the FIFO condition. *Strong FIFO* requires that $e_1 <_p e'_1$, $\lambda(e_1) = p!q(m)$, $\lambda(e'_1) = p!q(m')$ and $\mu(e'_1)$ defined entail $\mu(e_1) <_q \mu(e'_1)$, i.e., there is a single channel from p to q . The *weak FIFO* requirement used in Def. 1 means that there are as many FIFO channels from p to q as there are types of events $p!q(m)$. In general, there are undecidable problems in the strong FIFO semantics, as weak realizability [1], which are decidable in the weak FIFO semantics [13]. Anyway, all (un)decidability results established in this paper hold for both FIFO semantics, even though complexity depends on the semantics used.

Given a CMSC $X = (E, \lambda, \mu, (<_p)_{p \in \mathcal{P}})$, let \leq_X be the reflexive and transitive closure of the relation $<$ from Def. 1. A *linear extension* of X is an enumeration of E compatible with \leq_X . A *linearization* of X is the image of a linear extension of X under the map $\lambda : E \rightarrow \mathcal{E}$ (hence it is a word of \mathcal{E}^*). Let $\mathcal{Lin}(X)$ denote the set of linearizations of X . For a set \mathcal{X} of CMSCs, let $\mathcal{Lin}(\mathcal{X})$ denote the union of $\mathcal{Lin}(X)$ for all $X \in \mathcal{X}$. Linearizations can be defined more abstractly as follows:

Definition 2. Let $\mathcal{Lin} \subseteq \mathcal{E}^*$ be the set of all words w such that for all p, q and m , the number of occurrences $q ? p(m)$ is at most equal to the number of occurrences $p ! q(m)$ in every prefix v of w , and both numbers are equal for $v = w$. In the strong FIFO setting, we furthermore require the equality of contents of the i -th emission from p to q and of the i -th reception on q from p .

Any linearization w of an MSC belongs to \mathcal{Lin} (it may not be the case for a CMSC). Conversely, because of weak or strong FIFO, a word $w = \epsilon_1 \dots \epsilon_n \in \mathcal{Lin}$ is the linearization of a *unique* MSC, $Msc(w) = (\{1, \dots, n\}, \lambda, \mu, (<_p))$, with:

- $\lambda(i) = \epsilon_i$ and $i <_p j$ if $i < j$ and $\epsilon_i, \epsilon_j \in \mathcal{E}_p$,
- $\mu(i) = j$ if the letter $\epsilon_i = p ! q(m)$ occurs k times in $\epsilon_1 \dots \epsilon_i$ and the letter $\epsilon_j = q ? p(m)$ occurs k times in $\epsilon_1 \dots \epsilon_j$ for some p, q, m, k .

Definition 3. Two words $w, w' \in \mathcal{Lin}$ are *equivalent* (notation $w \equiv w'$) if $Msc(w)$ and $Msc(w')$ are isomorphic. For any language $\mathcal{L} \subseteq \mathcal{Lin}$, we write $[\mathcal{L}] = \{w \mid w \equiv w', w' \in \mathcal{L}\}$. A language $\mathcal{L} \subseteq \mathcal{Lin}(\mathcal{X})$ is a *representative set* for a set \mathcal{X} if $\mathcal{L} \cap \mathcal{Lin}(X) \neq \emptyset$ for all $X \in \mathcal{X}$, or equivalently, if $[\mathcal{L}] = \mathcal{Lin}(\mathcal{X})$.

We deduce the following properties. For any MSC X , $\mathcal{Lin}(X)$ is an equivalence class in \mathcal{Lin} . For any MSC X and for any $w \in \mathcal{Lin}$, $w \in \mathcal{Lin}(X)$ if and only if X is isomorphic to $Msc(w)$. A similar property does not hold for arbitrary CMSCs. For instance, $(p ! q(m)) (q ? p(m)) (q ? p(m))$ belongs to $\mathcal{Lin}(X)$ for two different CMSCs X , where the emission is matched by μ either with the first or with the second reception.

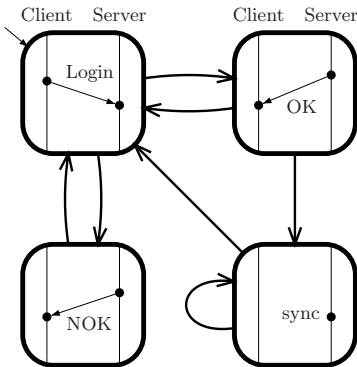


Fig. 1. Identification Scenario G_1

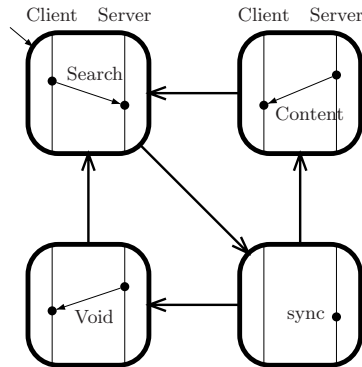


Fig. 2. Searching Scenario G_2

We define the concatenation $X_1 \cdot X_2$ of two CMSCs $X_i = (E^i, \lambda^i, \mu^i, \langle \cdot \rangle_p)_{p \in \mathcal{P}}$ as the set of CMSCs $X = (E^1 \uplus E^2, \lambda^1 \uplus \lambda^2, \mu, \langle \cdot \rangle_p)_{p \in \mathcal{P}}$ such that:

- $\mu \cap (E^i \times E^i) = \mu^i$ and $\langle \cdot \rangle_p \cap (E^i \times E^i) = \langle \cdot \rangle_p^i$ for $i \in \{1, 2\}$ and $p \in \mathcal{P}$,
- $e \in E^2$ and $e \leq_X e'$ entail $e' \in E^2$ for all $e, e' \in E^1 \uplus E^2$.

We let $\mathcal{X}_1 \cdot \mathcal{X}_2$ be the union of $X_1 \cdot X_2$ for all $X_i \in \mathcal{X}_i, i \in \{1, 2\}$. We can now give a description of sets of MSCs with rational operations.

Definition 4. A CMSC-graph is a tuple $G = (V, \rightarrow, \Lambda, V^0, V^f)$ where (V, \rightarrow) is a finite graph, $V^0, V^f \subseteq V$ are the subsets of initial and final vertices, respectively, and Λ maps each vertex v to a CMSC $\Lambda(v)$. We define $\mathcal{L}(G)$ as the set of all MSCs in $\Lambda(v_0) \cdot \Lambda(v_1) \cdot \dots \cdot \Lambda(v_n)$ where v_0, v_1, \dots, v_n is a path in G from some initial vertex $v_0 \in V^0$ to some final vertex $v_n \in V^f$. The CMSC-graph G is safe if any such set $\Lambda(v_0) \cdot \dots \cdot \Lambda(v_n)$ contains at least one MSC.

Intuitively, the semantics of CMSC-graphs is defined using the concatenation of the CMSCs labeling the vertices met along the paths in these graphs. Notice that $\Lambda(v_0) \cdot \dots \cdot \Lambda(v_n)$ may contain an arbitrary number of CMSCs, but at most one of these CMSCs is an MSC. An example of a non-safe CMSC-graph is $G = (V, \rightarrow, \Lambda, \{v_0\}, \{v_f\})$ where $V = \{v_0, v_f\}$, $v_0 \rightarrow v_f$, the CMSC $\Lambda(v_0)$ has a single event labeled with $q?p(m)$, and the CMSC $\Lambda(v_f)$ has a single event labeled with $p!q(m)$. Indeed the two events cannot be matched by μ in $\Lambda(v_0) \cdot \Lambda(v_f)$. Notice that this is a XCMSCG [12]. The reason why we do not allow XCMSCGs is that safe XCMSCGs are not necessarily existentially bounded, hence the Mazurkiewicz trace coding needed for the results of [4] that we use for Theorem 3 fails. Fig. 1 and 2 show two (C)MSC-graphs. Their nodes are labeled with MSCs. Concatenating *OK* and the local event *sync* gives an MSC with 3 events. The reception of *OK* and the event *sync* are unordered (in G_1). On the contrary, the event *sync* and the reception of *Void* are ordered (in G_2).

A safe CMSC-graph G may always be expanded into a safe atomic CMSC-graph G' , that is a graph in which each node is labeled with a single event, such that $\mathcal{L}(G) = \mathcal{L}(G')$. In the following, every safe CMSC-graph is assumed to be atomic. The expansion yields, by the way, a regular representative set for $\mathcal{L}(G)$.

3 Product of MSC-Languages

In order to master the complexity of distributed system descriptions, it is desirable to have at one's disposal a composition operation that allows us to weave different aspects of a system. When system aspects are CMSC-graphs with disjoint sets of processes, the concatenation of their MSC-languages can be used to this effect. Else, some parallel composition with synchronization capabilities is needed. We propose here to shuffle the events of the two MSC-graphs per process, except for the common events that serve to the synchronization. We require that all common events are internal events. Formally, what we define is an extension of the *mixed product* of words. The intersection with a regular

language could be used in place of the synchronizations to control the shuffle, but this would not change significantly the results of this paper. However, synchronizing on messages could change the results, as we can encode shared events using shared messages, but not the other way around.

First, we recall the definition of the *mixed product* $L_1 \parallel L_2$ of two languages L_1, L_2 of words (see [3]), defined on two alphabets Σ_1, Σ_2 not necessarily disjoint. Let $\Sigma = \Sigma_1 \cup \Sigma_2$. For $i = 1, 2$ let $\pi_i : \Sigma^* \rightarrow \Sigma_i^*$ be the unique monoid morphism such that $\pi_i(\sigma) = \sigma$ for $\sigma \in \Sigma_i$ and $\pi_i(\sigma) = \varepsilon$, otherwise. Then $L_1 \parallel L_2 = \{w \mid \pi_i(w) \in L_i, i = \{1, 2\}\}$ is the set of all words $w \in \Sigma^*$ with respective projections $\pi_i(w)$ in L_i ; e.g., $\{ab\} \parallel \{cad\} = \{cabd, cadb\}$ (a is the synchronizing action).

Definition 5. For $i = \{1, 2\}$, let \mathcal{X}_i be an MSC-language over some \mathcal{E}^i , such that $x \in \mathcal{E}^1 \cap \mathcal{E}^2$ implies $x = p(a)$ for some p, a . The mixed product $\mathcal{X}_1 \parallel \mathcal{X}_2$ is $Msc((\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2)) \cap \mathcal{L}in)$ and it is an MSC-language over $\mathcal{E}^1 \cup \mathcal{E}^2$.

The mixed product operation serves to compose the languages of two CMSC-graphs that share only internal events, as is the case for the CMSC-graphs G_1, G_2 of Fig. 1,2. The synchronization *sync* ensures that in any MSC in $\mathcal{L}(G_1) \parallel \mathcal{L}(G_2)$, the server never answers a search request from the client unless the client is logged in. Thus, synchronizations serve to avoid mixing incompatible fragments of the two CMSC-graphs. When a set \mathcal{X} is a singleton $\mathcal{X} = \{X\}$, we abusively write $X \parallel \mathcal{Y}$ instead of $\{X\} \parallel \mathcal{Y}$. Note that even though X_1 and X_2 are MSCs, $X_1 \parallel X_2$ may contain more than one MSC. Under weak FIFO semantics, mixing all linearizations pairwise yields all and only linearizations of a product of MSCs. However, the product of two linearizations of strong FIFO MSCs may contain words that are not linearizations of strong FIFO MSCs. Intersecting with $\mathcal{L}in$ allows us to keep only linearizations of (strong FIFO) MSCs.

Proposition 1. $\mathcal{L}in(\mathcal{X}_1 \parallel \mathcal{X}_2) = (\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2)) \cap \mathcal{L}in$.

Lemma 1. $(\mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2)) \cap \mathcal{L}in$ is closed under \equiv (see Def. 3).

However, $\{X_1\} \parallel \{X_2\}$ may be larger than $Msc(w_1 \parallel w_2)$ for fixed representations $w_1 \in \mathcal{L}in(X_1)$ and $w_2 \in \mathcal{L}in(X_2)$. This situation is illustrated with

$$\begin{aligned} w_1 &= (p!q(m_1))(q?p(m_1))(p!q(m_1))(q?p(m_1)), \\ w'_1 &= (p!q(m_1))^2(q?p(m_1))^2, \\ w_2 &= (q!p(m_2))(p?q(m_2))(q!p(m_2))(p?q(m_2)), \\ w'_2 &= (q!p(m_2))^2(p?q(m_2))^2, \\ w_3 &= (p!q(m_1))^2(q!p(m_2))^2(p?q(m_2))^2(q?p(m_1))^2. \end{aligned}$$

and $X_1 = Msc(w_1) = Msc(w'_1)$, $X_2 = Msc(w_2) = Msc(w'_2)$, $X_3 = Msc(w_3)$. There is no synchronization. Now $X_3 \in Msc(w'_1 \parallel w'_2)$, but $X_3 \notin Msc(w_1 \parallel w_2)$. This observation shows that products must be handled with care. Indeed, an advantage of CMSC-graphs is to represent large sets of linearizations with small subsets of representatives. However, w_1 is a representative for X_1 , w_2 is for X_2 , but $w_1 \parallel w_2$ is not a set of representatives for $X_1 \parallel X_2$.

4 Bounds for MSCs and Products

We review in this section ways of classifying CMSC-graphs based on bounds for communication channels, and we examine how these bounds behave under product of CMSC-languages. We focus on MSC-languages with *regular* representative sets. As indicated earlier, a regular representative set for the language of a safe CMSC-graph G may be obtained by expanding G into an atomic CMSC-graph G' . As observed in [12], it follows from a pumping lemma that whenever $\mathcal{L} \subseteq \text{Lin}$ is a regular representative set for some \mathcal{X} , the words in \mathcal{L} are uniformly B -bounded, for some $B > 0$, as defined hereafter. First, the definition of a channel depends on the semantics. In the weak FIFO setting, a channel is a triple $p, q \in \mathcal{P}, m \in \mathcal{M}$, and $p!q(m)$ is an emission ($q?p(m)$ is a reception) on this channel. In the strong FIFO setting, a channel is a pair $p, q \in \mathcal{P}$, and $p!q(m)$ is an emission ($q?p(m')$ is a reception) on this channel for any $m, m' \in \mathcal{M}$. A word $w \in \mathcal{E}^*$ is B -bounded if, for any prefix v of w and any channel c , the number of emissions on c in v exceeds the number of receptions on c in v by at most B .

A MSC X is \forall - B -bounded if every linearization $w \in \text{Lin}(X)$ is B -bounded. A MSC X is \exists - B -bounded if some linearization $w \in \text{Lin}(X)$ is B -bounded. A set of MSCs \mathcal{X} is \exists - B -bounded if all MSCs $X \in \mathcal{X}$ are \exists - B -bounded; \mathcal{X} is *existentially bounded* if it is \exists - B -bounded for some B . Let $\text{Lin}^B(\mathcal{X})$ denote the set of B -bounded words w in $\text{Lin}(\mathcal{X})$. Clearly, any \mathcal{X} with a regular representative set is existentially B -bounded for some B , but it may not be \forall - B -bounded for any B . Conversely, when an MSC-language \mathcal{X} is \exists - B -bounded, $\text{Lin}^B(\mathcal{X})$ is a representative set for \mathcal{X} , but it is not necessarily a regular language.

Proposition 2. $\text{Lin}^B(\mathcal{X}_1 \parallel \mathcal{X}_2) = (\text{Lin}^B(\mathcal{X}_1) \parallel \text{Lin}^B(\mathcal{X}_2)) \cap \text{Lin}^B$.

The above result shows that the mixed product behaves nicely with respect to bounded linearizations. If \mathcal{X}_1 and \mathcal{X}_2 are \forall - B -bounded, then $\text{Lin}(\mathcal{X}_i) = \text{Lin}^B(\mathcal{X}_i)$, and using Prop. 1, their product is also \forall - B -bounded. However, it may occur that both \mathcal{X}_1 and \mathcal{X}_2 are \exists - B -bounded but their mixed product is not existentially bounded. For instance, for all j , let X_1^j be the MSC with j messages m_1 from p to q and X_2^j be the MSC with j messages m_2 from q to p . All these MSCs are \exists -1-bounded since $(p!q(m_1)q?p(m_1))^j \in \text{Lin}(X_1^j)$ is 1-bounded. Define $\mathcal{X}_1 = \{X_1^j \mid j > 0\}$ and $\mathcal{X}_2 = \{X_2^j \mid j > 0\}$, thus $\mathcal{X}_1, \mathcal{X}_2$ are \exists -1-bounded, but $\mathcal{X}_1 \parallel \mathcal{X}_2$ is not \exists - B -bounded for any B since $\text{Msc}(p!q(m_1)^B(q!p(m_2)p?q(m_2))^Bq?p(m_1)^B) \in \mathcal{X}_1 \parallel \mathcal{X}_2$, but it is not \exists - $(B - 1)$ -bounded.

Definition 6. Given an MSC $X = (E, \lambda, \mu, (\prec_p)_{p \in \mathcal{P}})$ and a non-negative integer B , let Rev_B be the binary relation on E such that $e \text{Rev}_B e'$ if and only if, for some channel c , e is the i -th reception on channel c and e' is the $i + B$ -th emission on channel c . We also define $\text{Rev}_{\geq B} = \cup_{B' \geq B} \text{Rev}_{B'}$.

Proposition 3 (lemma 2 in [11]). A MSC X is \exists - B -bounded if and only if the relation $\prec \cup \text{Rev}_B$ is acyclic, if and only if the relation $\prec \cup \text{Rev}_{\geq B}$ is acyclic.

If X is \exists - B -bounded then X is \exists - B' -bounded for all $B' \geq B$, because Rev_B is included in the least order relation containing Rev_B and $\bigcup_{p \in \mathcal{P}} <_p$. For instance, in $Msc(p!q(m_1)^B (q!p(m_2) p?q(m_2))^B q?p(m_1)^B)$ let (a_i, b_i) denote the i -th pair of events $(p!q(m_1), q?p(m_1))$ and (c_i, d_i) the i -th pair of events $(q!p(m_2), p?q(m_2))$, then $a_B <_p d_1 Rev_{(B-1)} c_B <_q b_1 Rev_{(B-1)} a_B$ is a cycle.

5 Monitored Product of MSC-Languages

It is important to analyze formally MSC-languages, since following paths in MSC-graphs does not help grasping all the generated scenarios. Most often, in decidable cases [7,16], the analysis of an MSC-language \mathcal{X} amounts to check either the membership of a given MSC X , or whether $Lin(\mathcal{X})$ has an empty intersection with a regular language L (representing the complement of a desired property). In the case of a product language $\mathcal{X}_1 \parallel \mathcal{X}_2$, membership can be checked using the projections, since $X \in \mathcal{X}_1 \parallel \mathcal{X}_2$ if and only if $\pi_i(X) \in \mathcal{X}_i$ for $i = 1, 2$. However, in order to analyse regular properties of $\mathcal{L}(G_1) \parallel \mathcal{L}(G_2)$, one often needs computing a safe CMSC-graph G such that $\mathcal{L}(G) = \mathcal{L}(G_1) \parallel \mathcal{L}(G_2)$. In particular, one needs an existential bound B for the product. Unfortunately, the theorem below shows that one cannot decide whether such G exists when G_1 and G_2 share events on two processes or more.

Theorem 1. *Let G_1, G_2 be two (safe C)MSC-graphs. It is undecidable whether $\mathcal{L}(G_1) \parallel \mathcal{L}(G_2)$ is existentially bounded, in both weak and strong FIFO semantics.*

Proof. We show that the Post correspondence problem may be reduced to the above decision problem. Given two finite lists of words u_1, \dots, u_n and w_1, \dots, w_n on some alphabet Σ with at least two symbols, the problem is to decide whether $u_{i_1}u_{i_2} \dots u_{i_k} = w_{i_1}w_{i_2} \dots w_{i_k}$ for some non-empty sequence of indices $i_1 \dots i_k$. This problem is known to be undecidable for $n > 7$. Given an instance of the Post correspondence problem, *i.e.*, two lists of words u_1, \dots, u_n and w_1, \dots, w_n on Σ , consider the two MSC-graphs $G_1 = (V, \rightarrow, A_1, V^0, V^f)$ and $G_2 = (V, \rightarrow, A_2, V^0, V^f)$, with the same underlying graph $(V, \rightarrow, V^0, V^f)$, constructed as follows (G_1 is partially shown in Fig. 3).

Define $V = \{v_0, v_1, \dots, v_n, v_{n+1}\}$ with $V^0 = \{v_0\}$ and $V^f = \{v_{n+1}\}$. Let $v_0 \rightarrow v_i, v_i \rightarrow v_j$, and $v_i \rightarrow v_{n+1}$ for all $i, j \in \{1, \dots, n\}$ (where possibly $i = j$). Finally let $v_{n+1} \rightarrow v_{n+1}$.

For each $v \in V$, $A_1(v)$ is a finite MSC over $\mathcal{P}_1 = \{p, q\}$, $\mathcal{A}_1 = \{1, \dots, n\} \cup \Sigma$, $\mathcal{M}_1 = \{m_1, m'_1\}$. Actions $i \in \{1, \dots, n\}$ represent indices of pairs of words (u_i, v_i) and they occur on process p . Actions $\sigma \in \Sigma$ represent letters of words u_i and they occur on process q . Let $A_1(v_0)$ be the empty MSC. For $i \in \{1, \dots, n\}$, let $A_1(v_i)$ be the MSC with $p!q(m_1)$ followed by $p(i)$ on process p and with $q?p(m_1)$ followed by the sequence $q(\sigma_{i,1})q(\sigma_{i,2}) \dots q(\sigma_{i,l_i})$, representing $u_i = \sigma_{i,1} \sigma_{i,2} \dots \sigma_{i,l_i}$, on process q . Finally let $A_1(v_{n+1})$ be the MSC with the events $p!q(m'_1)$ and $q?p(m'_1)$ on processes p and q , respectively.

For each $v \in V$, $A_2(v)$ is a finite MSC over $\mathcal{P}_2 = \{p, r, q\}$, $\mathcal{A}_2 = \{1, \dots, n\} \cup \Sigma$, $\mathcal{M}_2 = \{m_2, m''_2, m'_2\}$. For $i = 0, \dots, n$, $A_2(v_i)$ is defined alike $A_1(v_i)$ but now

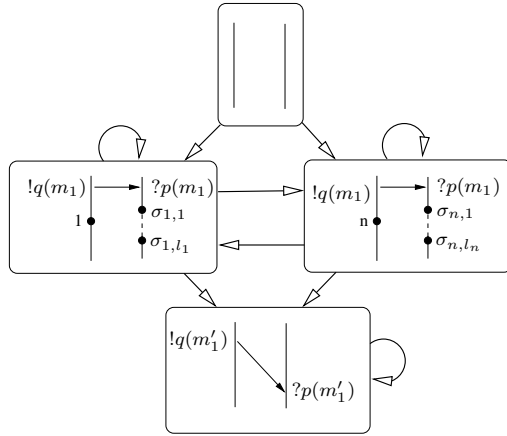


Fig. 3.

replacing the message $p!q(m_1), q?p(m_1)$ with two messages $p!r(m_2), r?p(m_2), r!q(m'_2), q?r(m'_2)$ and u_i with w_i . $A_2(v_{n+1})$ is the MSC with the events $p?q(m'_2)$ and $q!p(m'_2)$ on processes p and q , respectively.

For $i = 1, 2$ let $\mathcal{X}_i = \mathcal{L}(G_i)$, then $\text{Lin}^1(\mathcal{X}_i)$ is a regular representative set for \mathcal{X}_i . If the Post correspondence problem has no solution, then $\mathcal{X}_1 \parallel \mathcal{X}_2$ is empty, hence it is existentially bounded. In the converse case, $\mathcal{X}_1 \parallel \mathcal{X}_2$ contains for all B some MSC including a crossing of B messages m'_1 by B messages m'_2 , hence it is not existentially bounded. \square

The proof of Theorem 1 is inspired by the proof that $\mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset$ is undecidable for generic MSC-graphs G_1, G_2 [15]. Theorem 1 motivates the introduction of a *monitor* process mp and a *monitored product* in which all synchronizations are (internal) events located on the monitor process. The set of synchronizations is denoted by \mathcal{SE} . The *monitored product* $\mathcal{X}_1 \parallel_{mp} \mathcal{X}_2$ of sets \mathcal{X}_1 and \mathcal{X}_2 on monitor process $mp \in \mathcal{P}$ is defined only if $\mathcal{SE} \subseteq \{mp(a) \mid a \in \mathcal{A}\}$. In that case, we set $\mathcal{X}_1 \parallel_{mp} \mathcal{X}_2 = \mathcal{X}_1 \parallel \mathcal{X}_2$. For instance, in the monitored product $\mathcal{L}(G_1) \parallel_{mp} \mathcal{L}(G_2)$ of the CMSC-graphs of Fig. 1 and Fig. 2, we can choose $mp = server$ and $\mathcal{SE} = \{mp(sync)\}$. The adequacy of the monitored product to weave aspects of a distributed system is confirmed by the following theorem, which holds for both strong and weak FIFO semantics. We conjecture that the problem is PSPACE-complete in the strong FIFO case.

Theorem 2. *Given two safe CMSC-graphs G_1, G_2 , one can decide whether the monitored product of $\mathcal{L}(G_1)$ and $\mathcal{L}(G_2)$ is \exists -bounded. The problem is co-NP-complete and in PSPACE for weak and strong FIFO semantics respectively.*

The next section sketches a proof for this theorem. Notice that the proof is trivial in the case where G_1, G_2 have disjoint sets of processes except for mp . Then, $\mathcal{L}(G_1) \parallel_{mp} \mathcal{L}(G_2)$ is existentially bounded (with the bound given by the maximum of the minimal existential bounds of $\mathcal{L}(G_1)$ and $\mathcal{L}(G_2)$).

6 Checking Existential Boundedness

We prove Theorem 2 in two stages. First, we show that if the monitored product $\mathcal{L}(G_1) \parallel_{mp} \mathcal{L}(G_2)$ is existentially bounded, then this property holds for a 'small' bound with respect to the size of G_1 and G_2 .

Proposition 4. *Given two safe CMSC-graphs G_1 and G_2 , the MSC-language $\mathcal{L}(G_1) \parallel_{mp} \mathcal{L}(G_2)$ is existentially bounded if and only if it is existentially B^w -bounded (resp. B^s -bounded) for weak (resp. strong) FIFO semantics, where $B^w = 2K_1B'$, $B^s = 2K_2K_3B'$, and K_1, K_2, B' (resp. K_3) are polynomial (resp. exponential) in the size of P, G_1, G_2 .*

Then we show that one can check whether the monitored product of $\mathcal{L}(G_1)$ and $\mathcal{L}(G_2)$ is \exists - B -bounded, using the bounds B^w, B^s of Prop. 4. Notice that B^s written in binary is of size polynomial in $|G_1| + |G_2|$.

Proposition 5. *Given two safe CMSC-graphs G_1, G_2 and an integer B , it is co-NP-complete (resp. PSPACE) to decide whether $\mathcal{L}(G_1) \parallel_{mp} \mathcal{L}(G_2)$ is \exists - B -bounded, for weak (resp. strong) FIFO semantics. The PSPACE result holds also when B is written in binary.*

★ *Graph representation of monitored products*

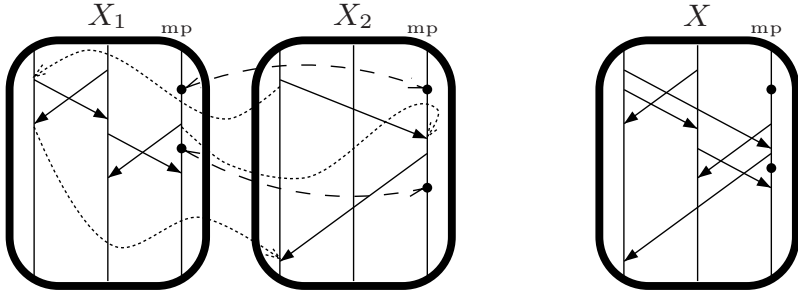


Fig. 4. $X \in X^1 \parallel_{mp} X^2$ and the corresponding relations $\rightarrow^1 \cup \rightarrow^2, \leftrightarrow$

These two results are obtained using special representations for MSCs constructed by monitored product. Let $X \in X_1 \parallel_{mp} X_2$ then $\exists w \in \mathcal{L}in: X = Msc(w)$ and $\pi_i(w) = w_i \in \mathcal{L}in(X_i)$. The MSC X is determined up to isomorphism by its projections on processes, because of FIFO. More precisely, for each $p \in \mathcal{P}$, $\pi_p(w) \in \pi_p(w_1) \parallel \pi_p(w_2)$. Moreover, for $p = mp$, $\pi_p(w_1)$ and $\pi_p(w_2)$ have the same projection on \mathcal{SE} . Therefore the projection $(E_p, <_p)$ of X on each process p may be seen as an interleaving of $(E_p^1, <_p^1)$ and $(E_p^2, <_p^2)$ where the synchronized pairs of events $e_1 \in E_{mp}^1$ and $e_2 \in E_{mp}^2$ with labels in \mathcal{SE} are coalesced. Let $\longleftrightarrow \subseteq E_{mp}^1 \times E_{mp}^2$ be the relation comprising synchronized pairs of events. For each $p \in \mathcal{P}$, let $\rightarrow_p^1 \subseteq E_p^2 \times E_p^1$ (resp. $\rightarrow_p^2 \subseteq E_p^1 \times E_p^2$) be the relation comprising ordered pairs of events $e_2 e_1$ (resp. $e_1 e_2$) switching from E_p^2 to E_p^1

(resp. E_p^1 to E_p^2) in the interleaved sequence $(E_p, <_p)$. The MSC X may now be represented by the juxtaposition of X_1 and X_2 interlinked with \longleftrightarrow and with the relations \rightarrow_p^1 and \rightarrow_p^2 for all $p \in \mathcal{P}$. The result is a *graph*, that we denote $X_{1\parallel_{mp}2}$, with set of nodes $E^1 \cup E^2$. Conversely, any acyclic graph connecting X_1 and X_2 with relations \rightarrow_p^i and \longleftrightarrow represents a non-empty set of weak FIFO MSCs \mathcal{X} . We say that the transitive closure $<_{\parallel_{mp}}$ of $<_i^p, \rightarrow_p^i$ and \longleftrightarrow is *compatible with strong FIFO* if there do not exist two messages $(s, r), (s', r')$ on the same channel c such that $s < s'$ and $r' < r$. There may be several such MSCs if for some p the relation $\rightarrow_p^1 \cup \rightarrow_p^2 \cup <_p^1 \cup <_p^2$ is not a total order on E_p . Otherwise, the original MSC X may be reconstructed from $X_{1\parallel_{mp}2}$ as follows: E is the quotient of $E^1 \cup E^2$ by the equivalence relation \longleftrightarrow and $<_{\parallel_{mp}} \upharpoonright_E$. For an illustration, see Fig. 4 where the edges of the graph represent the relations $<_p^i, \mu_i, \longleftrightarrow$ (dashed) and \rightarrow_p^i (dotted). The graph is compatible with strong FIFO. A unique MSC X can be reconstructed from it, depicted on the right of the figure. More formally, we can state the following lemma:

Lemma 2. *Let G_1 and G_2 be safe and atomic CMSC-graphs and B an integer. Then $\mathcal{L}(G_1) \parallel_{mp} \mathcal{L}(G_2)$ is \exists - B -bounded if and only if, for any synchronized pair of MSCs $X_1 \in \mathcal{L}(G_1)$ and $X_2 \in \mathcal{L}(G_2)$ with respective sets of events E^1 and E^2 , there is no subset $\{e_1, \dots, e_n\} \subseteq E^1 \cup E^2$ with at most two events in $E_p^1 \cup E_p^2$ for each process $p \in \mathcal{P}$ such that:*

1. *for all j , $(e_j, e_{(j+1) \bmod n})$ belongs to one of the relations $<^i, Rev_B$, or $E_p^i \times E_p^{3-i}$ for $i = 1$ or 2 and $p \in \mathcal{P}$,*
2. *there is no proper cycle in $\{e_1, \dots, e_n\}$ w.r.t. the transitive closure $<_{\parallel_{mp}}$ of the relation $<^1 \cup <^2 \cup \longleftrightarrow \cup \rightarrow$ where \longleftrightarrow is the synchronizing relation among coalesced events, and $e \rightarrow e'$ if $e = e_j \in E_p^i$ and $e' = e_{(j+1) \bmod n} \in E_p^{3-i}$ for some $j \in \{1, \dots, n\}$, $i \in \{1, 2\}$ and $p \in \mathcal{P}$,*
3. *in the strong FIFO case, $<_{\parallel_{mp}}$ is compatible with strong FIFO.*

The proofs of Prop. 4 and 5 are based on synchronized paths and Lemma 3. A *synchronized path* $\alpha\beta_1 \dots \beta_n\gamma$ of G_1, G_2 is a sequence of pairs of paths $\alpha = (\alpha^1, \alpha^2), \beta_i = (\beta_i^1, \beta_i^2), \gamma = (\gamma^1, \gamma^2)$, where $\alpha^k\beta_1^k \dots \beta_n^k\gamma^k$ is a path of $G_k, \pi_{SE}(\alpha^1) = \pi_{SE}(\alpha^2), \pi_{SE}(\beta_i^1) = \pi_{SE}(\beta_i^2)$ and $\pi_{SE}(\gamma^1) = \pi_{SE}(\gamma^2)$. Furthermore, β_i^k is a loop of G_k for all i, k . Lemma 3 claims that if n is sufficiently large, there exists a synchronized loop of ρ_2 which has no contribution to the ordering between events in ρ_1 and ρ_2 . This loop can thus be removed or iterated without compromising acyclicity, and is compatible with strong FIFO if needed.

Lemma 3. *Let G_1 and G_2 be safe and atomic CMSC-graphs, K be an integer and $(\alpha^1, \alpha^2)(\beta_1^1, \beta_1^2) \dots (\beta_K^1, \beta_K^2)(\gamma^1, \gamma^2)$ be a synchronized path of G_1, G_2 . Let \rightarrow be a partial order on a set E of $n \leq 2|\mathcal{P}|$ events of $\alpha^1 \cup \alpha^2 \cup \gamma^1 \cup \gamma^2$ compatible with the order of the synchronized path. For all $j \geq i \geq 1, \ell \geq 0$, we denote by $<_{i,j}^\ell$ the relation on $(\alpha^1, \alpha^2)(\beta_1^1, \beta_1^2) \dots [(\beta_i^1, \beta_i^2) \dots (\beta_j^1, \beta_j^2)]^\ell \dots (\beta_K^1, \beta_K^2)(\gamma^1, \gamma^2)$ generated by the synchronizations and the relation \rightarrow .*

- For all $i, j, i', j', <^1_{i,j} = <^1_{i',j'}$, denoted $<$, and this relation is a partial order.
- Let K_1, K_2, K_3 be the constants of Prop. 4.
- If $K > K_1$, then there exists i such that for all $x, y \in \alpha^1 \cup \alpha^2 \cup \gamma^1 \cup \gamma^2$ and $l \geq 0$, we have $x <^l_{i,i} y$ iff $x < y$ (in particular, $<^l_{i,i}$ is a partial order).
- If $K > K_2 K_3$ and $<$ is compatible with strong FIFO, then there exist i, j such that $<^l_{i,j}$ is an order compatible with strong FIFO, for all $l \geq 0$.

★ *General outline of the proof for Prop. 4*

Let $X \in \mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)$, thus X may be represented in product form by $X_{1 \parallel_{\text{mp}} 2} = (X_1, X_2, \longleftrightarrow, (\rightarrow^i_p)_{p \in \mathcal{P}}^{i=1,2})$. Suppose that X is not \exists - B -bounded for some $B = 2KB'$. By Prop. 3, $< \cup \text{Rev}_{\geq B}$ has a cycle in X . We have $\text{Rev}_{\geq B}^1 \cup \text{Rev}_{\geq B}^2 \subseteq \text{Rev}_{\geq B} \subseteq \text{Rev}_{\geq B/2}^1 \cup \text{Rev}_{\geq B/2}^2$. Therefore, the union of \longleftrightarrow and the relations $<^i$, $\text{Rev}_{\geq KB'}^i$, and \rightarrow^i_p for $i = 1, 2$ has a cycle $e_1 e_2 \dots e_m$ with $e_j \neq e_k$ for $j \neq k$. We let $e_{m+1} = e_1$. One can assume that $e_1 e_2 \dots e_m$ contains no synchronization event with shared label and at most two events on each process p (Lemma 5.5 in [4]), hence $m \leq 2|\mathcal{P}|$. Furthermore, there is at least one pair of events (e_j, e_{j+1}) in $\text{Rev}_{B_j}^i$, w.l.o.g. $e_1 \text{Rev}_{B_1}^1 e_2$, with $B_1 \geq KB'$. Notice that $(e_1 \dots e_m)$ is also a cycle for the union of \longleftrightarrow , $<^i$, $\text{Rev}_{\geq KB'}^i$, and $\rightarrow^i_p \cap (e_j, e_{j+1})_{j \leq m}$ for $i = 1, 2$, that is we need to consider only a linear number of pairs in \rightarrow^i_p . We construct MSCs $X'_1 \in \mathcal{L}(G_1), X'_2 \in \mathcal{L}(G_2)$ embedding X_1, X_2 via $\phi : X_i \hookrightarrow X'_i$ such that $\phi(e_1) \phi(e_2) \dots \phi(e_m)$ is a cycle for $\phi(\text{Rev}_{\geq KB'} \cup <_{X'})$, where $(X', <_{X'})$ is the oriented graph obtained by connecting X'_1 and X'_2 with \longleftrightarrow and $\phi(\rightarrow^i_p) \cap (\phi(e_j), \phi(e_{j+1}))_{j \leq m}$. More precisely, X'_1, X'_2 are such that $\phi(e_1) \text{Rev}_{\geq 2B_1+1}^1 \phi(e_2)$ and $e_j \text{Rev}_{B_j}^i e_{j+1} \Rightarrow \phi(e_j) \text{Rev}_{\geq B_j}^i \phi(e_{j+1})$ for $j \neq 1$ and $B_j \geq KB'$. As soon as $<_{X'}$ is a partial order (compatible with strong FIFO if needed), Prop. 4 follows by induction and by applying Lemma 2.

★ *General outline of the proof for Prop. 5*

In order to conclude that $\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)$ is not \exists - B -bounded, one should search for MSCs $X_1 \in \mathcal{L}(G_1), X_2 \in \mathcal{L}(G_2)$, and $X \in (X_1 \parallel_{\text{mp}} X_2)$ such that $<_X \cup \text{Rev}_B$ contains a cycle. In the weak FIFO setting, we use a small model property. Assume that the product of $\mathcal{L}(G_1)$ and $\mathcal{L}(G_2)$ is not existentially B^w bounded. We apply Lemma 2 to obtain a synchronized pair of paths ρ_1, ρ_2 of G_1, G_2 , with a set E of at most $2|\mathcal{P}|$ events, and a relation $\rightarrow \in E \times E$ which creates a cycle with $<^i \cup \text{Rev}_{B^w}^i$. By contradiction, assume that the minimal size of such a synchronized path (ρ_1, ρ_2) (that is its number of transitions) is larger than $((4|\mathcal{P}|B^w + 1)K_1 B')$, then it contains $(4|\mathcal{P}|B^w + 1)K_1$ synchronized pairs of loops. Applying Lemma 3 with $\ell = 0$, we know that there are $4|\mathcal{P}|B^w + 1$ loops which can be individually deleted without changing the order on E . There are at most $2|\mathcal{P}|B^w$ messages which can affect the $\text{Rev}_{B^w}^i$ relation, hence $4|\mathcal{P}|B^w$ loops which contain some emission or reception of such messages. Therefore, one synchronized pair of loops can be deleted without changing the order on E nor the $\text{Rev}_{B^w}^i$ relations, which contradicts the minimality of ρ_1, ρ_2 . To obtain

a co-NP algorithm, it suffices to guess a path of G_1 and a path of G_2 of size polynomial, to guess $2|\mathcal{P}|$ events, and to check in polynomial time that there is no cycle in $\langle^1 \cup \langle^2 \cup \leftrightarrow \cup \rightarrow$, whereas there is a cycle in $\langle^1 \cup \langle^2 \cup \leftrightarrow \cup \rightarrow \cup Rev_{B^s}^i$. Notice that we cannot do the same in the strong FIFO setting, since the exponential bound B^s would lead to a co-NEXPTIME algorithm. Instead, we construct a finite automaton, whose language is empty iff the product is existentially B^s bounded. Each state can be described in polynomial space w.r.t. $|G_1|, |G_2|$ and $\|B^s\| = \log_2(B^s)$ written in binary.

★ *General outline of the co-NP-completeness reduction*

We prove the co-NP hardness of the problem of deciding either the existential-boundedness or the existential- B -boundedness of the product of languages of two MSC-graphs. We do not use the contents of the messages, hence the reduction holds for both weak and strong FIFO semantics. Let ϕ be a 3-CNF-SAT instance, with n variables and m clauses. This formula is true iff for each clause, one can choose a literal of the clause to be true, and no conflict occurs on a variable (one cannot choose a literal and its opposite being true). Let $B > m + 1$. We build two MSC-graphs G_1 and G_2 on processes $\{p, q, r, p_i, p'_i \mid 1 \leq i \leq n\}$ such that $G_1 \parallel_{mp} G_2$ is \exists - B -bounded iff ϕ is non satisfiable. We let $mp = p$.

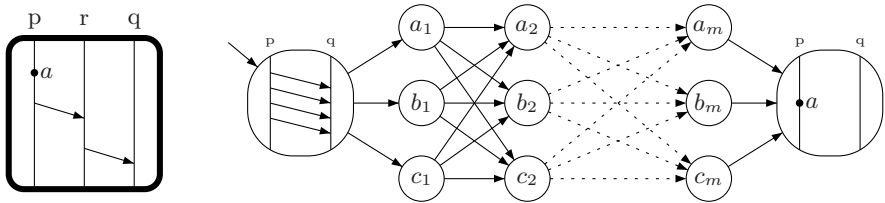


Fig. 5. MSC M_1 and MSC-graph G_2

Graph G_1 is made of one node, both initial and final. The node is labeled by MSC M_1 , which is a synchronization action a on process p , then a message from p to r , then a message from r to q . For graph G_2 , the initial node is labeled with $B + 1$ messages from p to q . Then G_2 has a succession of m choices between three nodes $a_i, b_i, c_i, i \leq m$. Then the final node of G_2 is labeled by the synchronization event a on process p . Informally, the m choices correspond to the m clauses, and a_i, b_i, c_i correspond to the choice of the first, second and third literal true in the i -th clause. That is, if the first literal in the i -th clause is v_j , then a_i is labeled by a message from q to p_j and a message from p'_j to p . If the first literal in the i -th clause is $\neg v_j$, then a_i is labeled by a message from q to p'_j and a message from p_j to p . Any MSC from G_2 corresponds to some choice of literal true in each of the clauses and vice versa. Now, a conflict occurs on one variable iff the receptions on q from p (in G_2) are before the synchronization event a , iff for the corresponding MSC M_2 of G_2 , all MSCs in $M_2 \parallel_{mp} M_1$ are \exists - B -bounded.

7 CMSC-Graph Representation of a Monitored Product

In the case where $\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)$ is \exists -bounded, one may wish to compute a safe CMSC-graph representation of this MSC-language, which can be input to existing tools for analyzing MSC-graphs (MSCan, SOFAT...). For this purpose, we use the results from [4], where a syntax-semantics correspondence is established between *globally cooperative* CMSC-graphs [7], and MSC-languages \mathcal{X} with regular representative sets $\text{Lin}^B(\mathcal{X})$ for some $B > 0$.

Definition 7. $G = (V, \rightarrow, \Lambda, V^0, V^f)$ is a globally cooperative CMSC-graph if

- G is a safe CMSC-graph, and
- for any circuit $v_1 \dots v_n$ in G , all CMSCs in the set $\Lambda(v_1) \cdot \dots \cdot \Lambda(v_n)$ have connected communication graphs.

The communication graph induced by $X = (E, \lambda, \mu, (\langle_p \rangle_{p \in \mathcal{P}})$ is the undirected graph (Q, E) with the set of vertices $Q = \{p \in \mathcal{P} \mid (\exists e \in E) \lambda(e) \in \mathcal{S}_p \cup \mathcal{R}_p\}$ and with the set of edges $E = \{\{p, q\} \mid (\exists e_1, e_2 \in E) (\exists m \in \mathcal{M}) \lambda(e_1) = p!q(m) \wedge \lambda(e_2) = q?p(m)\}$.

Notice that the MSC-graph from Fig. 3 is globally cooperative. Thus, boundedness of the product of $\mathcal{L}(G_1)$ and $\mathcal{L}(G_2)$ stays undecidable even when both G_1, G_2 are globally cooperative (Theorem 1). Quite remarkably, $\mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset$ is decidable as soon as G_1 or G_2 is globally cooperative [7].

Theorem 3. Let \mathcal{X} be a set of MSCs. The following are equivalent:

- $\mathcal{X} = \mathcal{L}(G)$ for some globally cooperative CMSC-graph G ,
- $\text{Lin}^B(\mathcal{X})$ is a regular representative set for \mathcal{X} for sufficiently large $B > 0$. Moreover, B and a finite automaton recognizing $\text{Lin}^B(\mathcal{X})$ can be computed effectively from G . Conversely, G can be computed effectively from $\text{Lin}^B(\mathcal{X})$.

The statement of Theorem 3 is the same as (a fragment of) the main theorem of [4]. However, we consider in this paper messages with contents, while [4] does not. Instead of proving Theorem 3 from scratch, we derive it from [4]. The strong FIFO case comes directly from the proof of [4]. For weak FIFO, we use a translation from sets of weak FIFO MSCs to sets of FIFO MSCs with exactly one (type of) message m (hence they embed in weak FIFO MSCs). In few words, the translation adds as many processes as types of messages per channel, and it preserves the existential boundedness of sets of MSCs, although the bound B may grow to $3B$. Once this translation is defined, the proof of Theorem 3 is almost immediate.

Now let G_1, G_2 be two globally cooperative CMSC-graphs. If $\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)$ is \exists -bounded, then this MSC-language is \exists - B -bounded, for $B \in \{B^s, B^w\}$ as defined in Prop. 4. Therefore, $\text{Lin}^B(\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2))$ is a representative set for $\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)$. By Prop. 2, $\text{Lin}^B(\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)) = \text{Lin}^B(\mathcal{L}(G_1)) \parallel_{\text{mp}} \text{Lin}^B(\mathcal{L}(G_2)) \cap \text{Lin}^B$. Since both G_1, G_2 are globally cooperative, we get that both $\text{Lin}^B(\mathcal{L}(G_1))$ and $\text{Lin}^B(\mathcal{L}(G_2))$ are regular and effectively computable. Since the shuffle of regular language is regular, we get the following.

Theorem 4. *Let G_1, G_2 be two globally cooperative CMSC-graphs such that $\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)$ is \exists -bounded. Then one can effectively compute a globally cooperative CMSC-graph G with $\mathcal{L}(G) = \mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)$. Moreover, G is of size at most exponential and doubly exponential in the size of $|G_1|, |G_2|$, respectively with weak and strong FIFO.*

8 Conclusion

We presented a framework to work with the controlled products of distributed components, granted that synchronizations are operated on a single monitor process, and components are given as globally cooperative CMSC-graphs. Namely, one can test whether the monitored product of components can be represented as a globally cooperative CMSC-graph. In that case, a complete analysis of the product system can be performed with existing tools. We analyze the problem in both weak and strong FIFO contexts. Weak FIFO enjoys a better complexity, while strong FIFO allows us to use *non-synchronized* actions with common names on different components (it suffices to rename the actions according to components, perform the product, and then rename the actions back). A direction for future work is to propose guidelines and tools for modeling product systems with one monitor process.

References

1. Alur, R., Etessami, K., Yannakakis, M.: Realizability and Verification of MSC Graphs. In: Orejas, F., Spirakis, P.G., van Leeuwen, J. (eds.) ICALP 2001. LNCS, vol. 2076, pp. 797–808. Springer, Heidelberg (2001)
2. Caillaud, B., Darondeau, P., Hélouët, L., Lesventes, G.: HMSCs as Partial Specifications.. with PNs as Completions. In: Cassez, F., Jard, C., Rozoy, B., Dermot, M. (eds.) MOVEP 2000. LNCS, vol. 2067, pp. 125–152. Springer, Heidelberg (2001)
3. Duboc, C.: Mixed Product and Asynchronous Automata. *Theoretical Computer Science* 48(3), 183–199 (1986)
4. Genest, B., Kuske, D., Muscholl, A.: A Kleene Theorem and Model Checking for a Class of Communicating Automata. *Inf. Comput.* 204(6), 920–956 (2006)
5. Gunter, E., Muscholl, A., Peled, D.: Compositional Message Sequence Charts. *STTT* 5(1), 78–89, (2003); In: Margaria, T., Yi, W. (eds.) TACAS 2001. LNCS, vol. 2031, pp. 496–511. Springer, Heidelberg (2001)
6. Genest, B., Muscholl, A., Peled, D.: Message Sequence Charts. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) ACPN 2003. LNCS, vol. 3098, pp. 537–558. Springer, Heidelberg (2004)
7. Genest, B., Muscholl, A., Seidl, H., Zeitoun, M.: Infinite-state High-level MSCs: Model-checking and Realizability. *JCSS* 72(4), 617–647 (2006); Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.): ICALP 2002. LNCS, vol. 2380, pp. 617–647. Springer, Heidelberg (2002)
8. Hélouët, L., Jard, C.: Conditions for synthesis of communicating automata from HMSCs. In: FMICS 2000, pp. 203–224 (2000)
9. Henriksen, J.G., Mukund, M., Kumar, K.N., Sohoni, M.A., Thiagarajan, P.S.: A theory of regular MSC languages. *Inf. Comput.* 202(1), 1–38 (2005)

10. Klein, J., Caillaud, B., H elou et, L.: Merging scenarios. In: FMICS 2004, pp. 209–226 (2004)
11. Lohrey, M., Muscholl, A.: Bounded MSC communication. *Inf. Comput.* 189(2), 160–181 (2004)
12. Madhusudan, P., Meenakshi, B.: Beyond Message Sequence Graphs. In: Hariharan, R., Mukund, M., Vinay, V. (eds.) FSTTCS 2001. LNCS, vol. 2245, pp. 256–267. Springer, Heidelberg (2001)
13. Morin, R.: Recognizable Sets of Message Sequence Charts. In: Alt, H., Ferreira, A. (eds.) STACS 2002. LNCS, vol. 2285, pp. 523–534. Springer, Heidelberg (2002)
14. Mukund, M., Kumar, K.N., Sohoni, M.A.: Bounded time-stamping in message-passing systems. *TCS* 290(1), 221–239 (2003)
15. Muscholl, A., Peled, D., Su, Z.: Deciding properties of Message Sequence Charts. In: Nivat, M. (ed.) FOSSACS 1998. LNCS, vol. 1378, pp. 226–242. Springer, Heidelberg (1998)
16. Muscholl, A., Peled, D.: Message Sequence Graphs and Decision Problems on Mazurkiewicz Traces. In: Kutylowski, M., Wierzbicki, T., Pacholski, L. (eds.) MFCS 1999. LNCS, vol. 1672, pp. 81–91. Springer, Heidelberg (1999)