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 processes, 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 linearizations, 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.

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 $\mathcal{S} = \{p\,!\,q\,(m) \,|\, p, q \in \mathcal{P}, p \neq q, m \in \mathcal{M}\}$,$\mathcal{R} = \{p\,?\,q\,(m) \,|\, p, q \in \mathcal{P}, p \neq q, m \in \mathcal{M}\}$, and $\mathcal{I} = \{p\,(a) \,|\, p \in \mathcal{P}, a \in \mathcal{A}\}$. 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 $\lambda : E \to \mathcal{E}$,*
– *for each $p \in \mathcal{P}$, $<_p$ is a total order on $E_p = \lambda^{-1}(\mathcal{E}_p)$,*
– *$\mu : E \to E$ 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 $\lambda : E \to \mathcal{E}$ (hence it is a word of $\mathcal{E}^*$). Let $\mathcal{L}in(X)$ denote the set of linearizations of X. For a set $\mathcal{X}$ of CMSCs, let $\mathcal{L}in(\mathcal{X})$ denote the union of $\mathcal{L}in(X)$ for all $X \in \mathcal{X}$. Linearizations can be defined more abstractly as follows:

Definition 2. *Let $\mathcal{L}in \subseteq \mathcal{E}^*$ 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 $\mathcal{L}in$ (it may not be the case for a CMSC). Conversely, because of weak or strong FIFO, a word $w = \epsilon_1 \dots \epsilon_n \in \mathcal{L}in$ 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 $w, w' \in \mathcal{L}in$ are equivalent (notation $w \equiv w'$) if $Msc(w)$ and $Msc(w')$ are isomorphic. For any language $\mathcal{L} \subseteq \mathcal{L}in$, we write $[\mathcal{L}] = \{w \mid w \equiv w', w' \in \mathcal{L}\}$. A language $\mathcal{L} \subseteq \mathcal{L}in(\mathcal{X})$ is a* representative set *for a set $\mathcal{X}$ if $\mathcal{L} \cap \mathcal{L}in(X) \neq \emptyset$ for all $X \in \mathcal{X}$, or equivalently, if $[\mathcal{L}] = \mathcal{L}in(\mathcal{X})$.*

We deduce the following properties. For any MSC X, $\mathcal{L}in(X)$ is an equivalence class in $\mathcal{L}in$. For any MSC X and for any $w \in \mathcal{L}in$, $w \in \mathcal{L}in(X)$ 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 $\mathcal{L}in(X)$ for two different CMSCs X, where the emission is matched by μ either with the first or with the second reception.

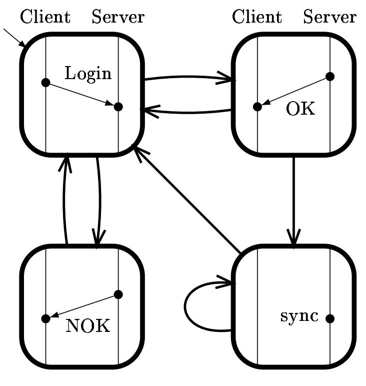

Figure1. Identification Scenario G_1.

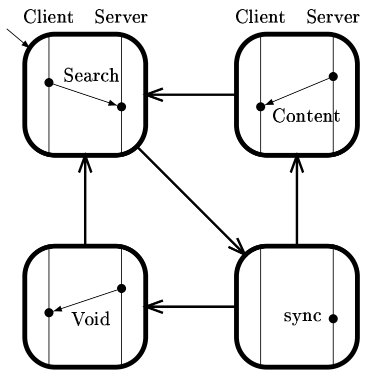

Figure2. Searching Scenario G_2.

We define the concatenation $X_1 \cdot X_2$ of two CMSCs $X_i = (E^i, \lambda^i, \mu^i, (<_p^i)_{p \in \mathcal{P}})$ as the set of CMSCs $X = (E^1 \uplus E^2, \lambda^1 \uplus \lambda^2, \mu, (<_p)_{p \in \mathcal{P}})$ such that:
– $\mu \cap (E^i \times E^i) = \mu^i$ and $<_p \cap (E^i \times E^i) = <_p^i$ 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 $\Lambda(v_0) \cdot \Lambda(v_1) \cdot \ldots \cdot \Lambda(v_n)$ where $v_0, v_1 \ldots, v_n$ 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 $\Lambda(v_0) \cdot \ldots \cdot \Lambda(v_n)$ 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 $\Lambda(v_0) \cdot \ldots \cdot \Lambda(v_n)$ 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 XCMSG [12]. The reason why we do not allow XCMSGs is that safe XCMSGs 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 $\pi_i : \Sigma^* \to \Sigma_i^*$ 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. *$\mathcal{L}in(\mathcal{X}_1 \parallel \mathcal{X}_2) = (\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2)) \cap \mathcal{L}in$.*

Proof. By definition, $\mathcal{X}_1 \parallel \mathcal{X}_2 = Msc(\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2)) = Msc(\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2) \cap \mathcal{L}in)$. Notice that $Msc(w)$ is undefined for $w \notin \mathcal{L}in$. Moreover, $\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2) \cap \mathcal{L}in \subseteq \mathcal{L}in(\mathcal{X}_1 \parallel \mathcal{X}_2)$, because $\mathcal{L}in(X)$ is an equivalence class. Now let $w \in \mathcal{L}in(X)$ and $X \in Msc\,(\mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2))$ for some $X_i \in \mathcal{X}_i$ $(i = 1, 2)$. Again, using the properties of $\mathcal{L}in(X)$, we know that $X = Msc\,(w)$. Therefore, $w \in \mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2)$ by Lemma 1 (see below). □

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

Proof. Let $w \in \mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2) \cap \mathcal{L}in$. We want to show that for any w' in $\mathcal{L}in$ (Def. 2), if $Msc\,(w)$ and $Msc\,(w')$ are isomorphic, then $w' \in \mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2)$. Let $w = \epsilon_1 \ldots \epsilon_n$. ¿From Def. 2, $w' = \epsilon'_1 \ldots \epsilon'_n$ and there exists a bijection $f : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $\epsilon_i = \epsilon'_{f(i)}$. For $j = 1, 2$ let $E^j = \{i \,|\, 1 \leq i \leq n \wedge \epsilon_i \in \mathcal{E}^j\}$ and $E'^j = \{i \,|\, 1 \leq i \leq n \wedge \epsilon'_i \in \mathcal{E}^j\}$, then f restricts and co-restricts to bijections $f_j : E^j \to E'^j$, hence $Msc\,(\pi_j(w))$ and $Msc\,(\pi_j(w'))$ are isomorphic for $j = 1, 2$ (where $\pi_j(w)$ and $\pi_j(w')$ are the respective projections of w and w' on ${\mathcal{E}^j}^*$). Therefore, $\pi_j(w') \in \mathcal{L}in(X_j)$ for $j = 1, 2$ and $w' \in \mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2)$. □

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

$w_1 = (p\,!\,q\,(m_1))\,(q\,?\,p\,(m_1))\,(p\,!\,q\,(m_1))\,(q\,?\,p\,(m_1))$,

$w'_1 = (p\,!\,q\,(m_1))^2\,(q\,?\,p\,(m_1))^2$,

$w_2 = (q\,!\,p\,(m_2))\,(p\,?\,q\,(m_2))\,(q\,!\,p\,(m_2))\,(p\,?\,q\,(m_2))$,
$w'_2 = (q\,!\,p\,(m_2))^2\,(p\,?\,q\,(m_2))^2$,
$w_3 = (p\,!\,q\,(m_1))^2\,(q\,!\,p\,(m_2))^2\,(p\,?\,q\,(m_2))^2\,(q\,?\,p\,(m_1))^2$.
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 $\mathcal{L} \subseteq \mathcal{L}in$ 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 $w \in \mathcal{L}in(X)$ is B-bounded. A MSC X is *$\exists$-B-bounded* if some linearization $w \in \mathcal{L}in(X)$ 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 $\mathcal{L}in^B(\mathcal{X})$ denote the set of B-bounded words w in $\mathcal{L}in(\mathcal{X})$. 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, $\mathcal{L}in^B(\mathcal{X})$ is a representative set for $\mathcal{X}$, but it is not necessarily a regular language.

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

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 $\mathcal{L}in(\mathcal{X}_i) = \mathcal{L}in^B(\mathcal{X}_i)$, 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 $(p!q(m_1)q?p(m_1))^j \in \mathcal{L}in(X_1^j)$ 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 $Msc(p!q(m_1)^B(q!p(m_2)p?q(m_2))^B q?p(m_1)^B) \in \mathcal{X}_1 \parallel \mathcal{X}_2$, but it is not $\exists$-$(B-1)$-bounded.

Definition 6. *Given an MSC $X = (E, \lambda, \mu, (<_p)_{p\in\mathcal{P}})$ and a non-negative integer B, let Rev_B be the binary relation on E such that $e\, Rev_B\, e'$ 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 $Rev_{\geq B} = \cup_{B'\geq B} Rev_{B'}$.*

Proposition 3 (lemma 2 in [11]). *A MSC X is $\exists$-B-bounded if and only if the relation $< \cup\, Rev_B$ is acyclic, if and only if the relation $< \cup\, Rev_{\geq B}$ 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 $\mathcal{L}in(\mathcal{X})$ 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 $u_1, \ldots, u_n$ and $w_1, \ldots, w_n$ on some alphabet Σ with at least two symbols, the problem is to decide whether $u_{i_1}u_{i_2}\ldots u_{i_k} = w_{i_1}w_{i_2}\ldots w_{i_k}$ for some non-empty sequence of indices $i_1 \cdots i_k$. This problem is known to be undecidable for $n > 7$. Given an instance of the Post correspondence problem, *i.e.*, two lists of words $u_1, \ldots, u_n$ and $w_1, \ldots, w_n$ on Σ, consider the two MSC-graphs $G_1 = (V, \to, \Lambda_1, V^0, V^f)$ and $G_2 = (V, \to, \Lambda_2, V^0, V^f)$, with the same underlying graph $(V, \to, V^0, V^f)$, constructed as follows (G_1 is partially shown in Fig. 3).

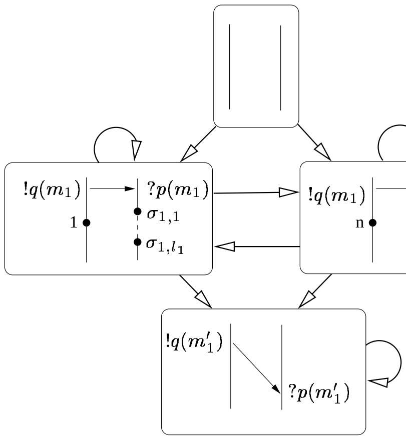

Figure3.

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

For each $v \in V$, $\Lambda_1(v)$ is a finite MSC over $\mathcal{P}_1 = \{p, q\}$, $\mathcal{A}_1 = \{1, \ldots, n\} \cup \Sigma$, $\mathcal{M}_1 = \{m_1, m_1'\}$. Actions $i \in \{1, \ldots, n\}$ 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 $\Lambda_1(v_0)$ be the empty MSC. For $i \in \{1, \ldots, n\}$, let $\Lambda_1(v_i)$ 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 $q(\sigma_{i,1})\, q(\sigma_{i,2}) \ldots q(\sigma_{i,l_i})$, representing $u_i = \sigma_{i,1}\, \sigma_{i,2} \ldots \sigma_{i,l_i}$, on process q. Finally let $\Lambda_1(v_{n+1})$ be the MSC with the events $p!q(m_1')$ and $q?p(m_1')$ on processes p and q, respectively.

For each $v \in V$, $\Lambda_2(v)$ is a finite MSC over $\mathcal{P}_2 = \{p, r, q\}$, $\mathcal{A}_2 = \{1, \ldots, n\} \cup \Sigma$, $\mathcal{M}_2 = \{m_2, m_2'', m_2'\}$. For $i = 0, \ldots, n$, $\Lambda_2(v_i)$ is defined alike $\Lambda_1(v_i)$ but now 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. $\Lambda_2(v_{n+1})$ 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 $\mathcal{L}in^1(\mathcal{X}_i)$ 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. □

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 *monitored product* $\mathcal{X}_1 \parallel_{\mathrm{mp}} \mathcal{X}_2$ of sets $\mathcal{X}_1$ and $\mathcal{X}_2$ on monitor process $mp \in \mathcal{P}$ is defined only if $\mathcal{SE} = \{mp(a) \in \mathcal{E}^1 \cap \mathcal{E}^2\}$. For instance, in the monitored product $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ 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, $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ 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 $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ 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 $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ is existentially bounded if and only if it is existentially B^w-bounded (resp. B^s-bounded) for weak (resp. strong) FIFO semantics, where $B^w = 2K_1B'$, $B^s = 2K_2K_3B'$, $|G_i|$ is the number of events in G_i, $B' = (2\,|\mathcal{P}| + 2)^2 \times (|G_1|+1) \times (|G_2|+1)$, $K_1 = (2\,|\mathcal{P}| + (|\mathcal{P}|)^2/2 \times (\,|\mathcal{M}_1| \times |G_1| + |\mathcal{M}_2| \times |G_2|))^2$, $K_2 = 4|E||\mathcal{P}|^2(|G_1| + |G_2||) + (4|E||\mathcal{P}|^2 + 2)(2|\mathcal{P}| + |\mathcal{P}|^2/2(|G_1| + |G_2|))^2$ and $K_3 = (|G_1| + |G_2|)^{|\mathcal{P}|} \times 3^{4(|G_1|+|G_2|+2)^2 \times (|\mathcal{P}|)^6}$.*

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 $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ 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*

These two results are obtained using special representations for MSCs constructed by monitored product. Let $X \in X_1 \parallel_{\mathrm{mp}} X_2$ 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

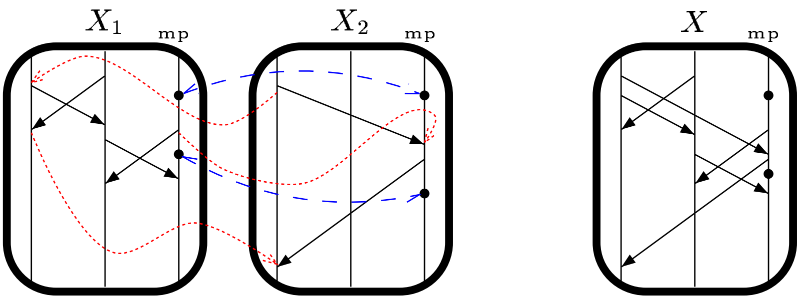

Figure4. $X \in X^1 \parallel_{\mathrm{mp}} X^2$ and the corresponding relations $\rightarrow^1 \cup \rightarrow^2$, $\leftrightarrow$

synchronized pairs of events $e_1 \in E^1_{mp}$ and $e_2 \in E^2_{mp}$ with labels in $\mathcal{SE}$ are coalesced. Let $\longleftrightarrow \subseteq E^1_{mp} \times E^2_{mp}$ be the relation comprising synchronized pairs of events. For each $p \in \mathcal{P}$, let $\rightarrow^1_p \subseteq E^2_p \times E^1_p$ (resp. $\rightarrow^2_p \subseteq E^1_p \times E^2_p$) be the relation comprising ordered pairs of events $e_2\, e_1$ (resp. $e_1\, e_2$) switching from E^2_p to E^1_p (resp. E^1_p to E^2_p) 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 $\rightarrow^1_p$ and $\rightarrow^2_p$ for all $p \in \mathcal{P}$. The result is a *graph*, that we denote $X_{1\|_{\mathrm{mp}}2}$, with set of nodes $E^1 \cup E^2$. Conversely, any acyclic graph connecting X_1 and X_2 with relations $\rightarrow^i_p$ and $\longleftrightarrow$ represents a non-empty set of weak FIFO MSCs $\mathcal{X}$. We say that the transitive closure $<_{\|_{\mathrm{mp}}}$ of $<^p_i$, $\rightarrow^i_p$ 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 $\rightarrow^1_p \cup \rightarrow^2_p \cup <^1_p \cup <^2_p$ is not a total order on E_p. Otherwise, the original MSC X may be reconstructed from $X_{1\|_{\mathrm{mp}}2}$ as follows: E is the quotient of $E^1 \cup E^2$ by the equivalence relation $\longleftrightarrow$ and $< = <_{\|_{\mathrm{mp}}} |_E$. For an illustration, see Fig. 4 where the edges of the graph represent the relations $<^i_p$, μ_i, $\longleftrightarrow$ (dashed) and $\rightarrow^i_p$ (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 $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ 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 $\{e_1, \ldots, e_n\} \subseteq E^1 \cup E^2$ with at most two events in $E^1_p \cup E^2_p$ 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 $E^i_p \times E^{3-i}_p$ for $i = 1$ or 2 and $p \in \mathcal{P}$,*
2. *there is no proper cycle in $\{e_1, \ldots, e_n\}$ w.r.t. the transitive closure $<_{\|_{\mathrm{mp}}}$ of the relation $<^1 \cup <^2 \cup \longleftrightarrow \cup \rightarrow$ where $\longleftrightarrow$ is the synchronizing relation among coalesced events, and $e \rightarrow e'$ if $e = e_j \in E^i_p$ and $e' = e_{(j+1) \bmod n} \in E^{3-i}_p$ for some $j \in \{1, \ldots n\}$, $i \in \{1, 2\}$ and $p \in \mathcal{P}$,*

3. in the strong FIFO case, $<_{\|_{\mathrm{mp}}}$ is compatible with strong FIFO.

The proofs of Prop. 4 and 5 are based on synchronized paths and Lemma 3. A *synchronized path* $\alpha\beta_1\cdots\beta_n\gamma$ 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 $\alpha^k\beta_1^k\cdots\beta_n^k\gamma^k$ is a path of G_k, $\pi_{\mathcal{SE}}(\alpha^1) = \pi_{\mathcal{SE}}(\alpha^2)$, $\pi_{\mathcal{SE}}(\beta_i^1) = \pi_{\mathcal{SE}}(\beta_i^2)$ and $\pi_{\mathcal{SE}}(\gamma^1) = \pi_{\mathcal{SE}}(\gamma^2)$. 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 $(\alpha^1, \alpha^2)(\beta_1^1, \beta_1^2)\cdots(\beta_K^1, \beta_K^2)(\gamma^1, \gamma^2)$ 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 $(\alpha^1, \alpha^2)(\beta_1^1, \beta_1^2)\cdots[(\beta_i^1, \beta_i^2)\cdots(\beta_j^1, \beta_j^2)]^{\ell}\cdots(\beta_K^1, \beta_K^2)(\gamma^1, \gamma^2)$ generated by the synchronizations and the relation $\rightarrow$.*

- *For all i, j, i', j', $<_{i,j}^1 = <_{i',j'}^1$, 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 $x <_{i,i}^{\ell} y$ iff $x < y$ (in particular, $<_{i,i}^{\ell}$ is a partial order).*
- *If $K > K_2K_3$ and $<$ is compatible with strong FIFO, then there exist i, j such that $<_{i,j}^{\ell}$ is an order compatible with strong FIFO, for all $l \geq 0$.*

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

Let $X \in \mathcal{L}(G_1) \|_{\mathrm{mp}} \mathcal{L}(G_2)$, thus X may be represented in product form by $X_{1\|_{\mathrm{mp}}2} = (X_1, X_2, \longleftrightarrow, (\rightarrow_p^i)_{p\in\mathcal{P}}^{i=1,2})$. Suppose that X is not $\exists$-B-bounded for some $B = 2KB'$. By Prop. 3, $< \cup Rev_{\geq B}$ has a cycle in X.We have $Rev_{\geq B}^1 \cup Rev_{\geq B}^2 \subseteq Rev_{\geq B} \subseteq Rev_{\geq B/2}^1 \cup Rev_{\geq B/2}^2$. Therefore, the union of $\longleftrightarrow$ and the relations $<^i$, $Rev_{\geq KB'}^i$, and $\rightarrow_p^i$ for $i = 1, 2$ has a cycle $e_1\, e_2 \ldots e_m$ with $e_j \neq e_k$ for $j \neq k$. We let $e_{m+1} = e_1$. One can assume that $e_1\, e_2 \ldots e_m$ 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 $Rev_{B_j}^i$, w.l.o.g. $e_1\, Rev_{B_1}^1\, e_2$, with $B_1 \geq KB'$. Notice that $(e_1\cdots e_m)$ is also a cycle for the union of $\longleftrightarrow$, $<^i$, $Rev_{\geq KB'}^i$, and $\rightarrow_p^i \cap (e_j, e_{j+1})_{j\leq m}$ for $i = 1, 2$, that is we need to consider only a linear number of pairs in $\rightarrow_p^i$. 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 $\phi(e_1)\,\phi(e_2)\ldots\phi(e_m)$ is a cycle for $\phi(Rev_{\geq KB'}) \cup <_{X'}$, where $(X', <_{X'})$ is the oriented graph obtained by connecting X_1' and X_2' with $\longleftrightarrow$ and $\phi(\rightarrow_p^i) \cap (\phi(e_j), \phi(e_{j+1}))_{j\leq m}$. More precisely, X_1', X_2' are such that $\phi(e_1)\, Rev_{\geq 2B_1+1}^1\, \phi(e_2)$ and $e_j\, Rev_{B_j}^i\, e_{j+1} \Rightarrow \phi(e_j)\, Rev_{\geq B_j}^i\, \phi(e_{j+1})$ 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.

★ *Sketch of the induction step*

For $i = 1, 2$ let ρ_i be the generating path for X_i in the (safe and) atomic CMSC-graph G_i. Let $e_1\, Rev^1_{B_1}\, e_2$ with $B_1 \geq KB'$, where $e_1\, e_2 \ldots e_m$ is the considered cycle in the union of the relations $<^i$, $Rev^i_{\geq KB'}$, $\rightarrow^i_p$ $(i = 1, 2)$, with $m \leq (2|\mathcal{P}|)$. Thus there exists a channel c, such that e_1 is a reception event on c and e_2 is an emission event on c. As $B_1 \geq KB'$, at least $KB' - 1$ emission events on c preceding e_2 in X_1 are matched by μ^1 with reception events on c following e_1 in X_1. At least $KB' - 1/\,|G_1|$ of these emission events originate from the same vertex v of G_1. As $(KB' - 1)/\,|G_1| > (2\,|\mathcal{P}| + 1)^2 \times (|G_2| + 1) \times K$ and $m \leq 2|\mathcal{P}|$, the path ρ_1 may be written as UVW such that:

- any event from X_1 in the cycle $e_1 \ldots e_m$ originates from the path prefix U or from the path suffix W,
- $V = vV_1vV_2v \ldots vV_kv$ for some $k \geq (2\,|\mathcal{P}| + 1) \times (|G_2| + 1) \times K$,
- e_1 and e_2 originate from occurrences of vertices in U and W, respectively. The reason is that we can always expand a safe CMSC graph G into an atomic CMSC graph, whose paths give a set of regular representatives bounded by some $B \leq |G|/2$. This also applies to the safe CMSC-graph G_1, and $k > K > |G_1|$: e_1 cannot be generated in the path after V).

Since the shared events are on only one process mp, path ρ_2 may be written in a similar form $U'v_1V'_1V'_2 \ldots V'_kW'$ such that Uv and $U'v_1$, resp. V_jv and V'_j, resp. W and W' synchronize on shared events (notice that if X, Y have no shared events, then they indeed synchronize). Define inductively for $i > 1, v_i$ is the last node of path V'_i if the path is nonempty, else $v_i = v_{i-1}$ ($V'_i = \epsilon$ acts as an (empty) self loop on v_{i-1}). As $k > (2\,|\mathcal{P}| + 1)\,|G_2| \times K$ and $m \leq (2|\mathcal{P}|)$, there must exist a strictly increasing sequence of indices $j_1 \ldots j_K$ such that $v_{j_1} = v_{j_2} = \ldots = v_{j_K}$ is the same vertex (of G_2) and no event (from X_2) in the cycle $e_1 \ldots e_m$ originates from $V'_{j_1} \ldots V'_{j_K}$. Then we let:

- $\alpha^1 = UvV_1v \ldots V_{j_1-1}v$, $\alpha^2 = U'v_1V'_1 \ldots V'_{j_1-1}$
- $\beta^1_h = V_{j_h}v \ldots V_{j_{h+1}-1}v$ and $\beta^2_h = V'_{j_h} \ldots V'_{j_{h+1}-1}$ for $1 \leq h \leq K$,
- $\gamma^1 = V_{j_K+1}v \ldots W$, and $\gamma^2 = V'_{j_K+1} \ldots W'$

Choose a fixed $h \in \{1, \ldots, K\}$. As G_1 is a safe CMSC-graph, the path $\alpha^1\beta^1_1 \cdots (\beta^1_h)^{B_1+2} \cdots \beta^1_K\gamma^1$, defines an MSC X'_1. The MSC X_1 embeds into X'_1 with the following $\phi : X_1 \hookrightarrow X'_1$ mapping events of X_1 generated from $\alpha^1\beta^1_1 \cdots \beta^1_{h-1}\beta^1_h$, respectively from $\beta^1_{h+1} \ldots \beta^1_K\gamma^1$, to similar events of X'_1, then $e <^1 e'$ in X_1 entails $\phi(e) <'^1 \phi(e')$ in X'_1 (whereas the converse implication needs not hold). Since $e_1 \in \alpha_1, e_2 \in \gamma_1$, and there is the same positive number of emission and reception event on every channel in the loop β^1_h of the safe CMSC-graph G_1, we have $\phi(e_1)\, Rev^1_{>2B_1}\, \phi(e_2)$. Let $e_j\, Rev^i_{B_j}\, e_{j+1}$. As $e_1, \ldots, e_m$ do not occur in V, e_j and e_{j+1} occur in α^1 or γ^1. As $B_j \geq B/2 \geq |G_1|$, and ρ_1 is a path in the safe and atomic CMSC-graph G_1, e_j must occur before e_{j+1}. Therefore, we cannot have e_j in γ^1 and e_{j+1} in α^1. That is, either e_j, e_{j+1} are both in α_1 or γ_1, or e_j is in α_1 and e_{j+1} in γ_1. In the two first cases, we easily have $\phi(e_j)\, Rev^i_{\geq B_j}\, \phi(e_{j+1})$

in X'_1. It is also the case in the latter case, in view of requirements of the definition of concatenation, which ensures that if s is the matching emission of e_j and r is the matching reception of e_{j+1}, then $\phi(s), \phi(r)$ are the matching event of $\phi(e_j), \phi(e_{j+1})$. The situation is analogous for the second component MSC X_2, and we let $\phi : X_2 \hookrightarrow X'_2$ be the map that embeds X_2 into the MSC generated from the path $\alpha^2 \beta^2_1 \cdots (\beta^2_h)^{B_1+2} \cdots \beta^2_K \gamma^2$.

Let $(X', <_{X'})$ be the directed graph formed by connecting X'_1 and X'_2 with relation $\longleftrightarrow$ plus edges $(\phi(e_j), \phi(e_{j+1}))$ for all $j \leq m$ such that $e_j \rightarrow^i_p e_{j+1}$ for some $i \in \{1, 2\}$ and $p \in \mathcal{P}$. By construction, $\phi(e_1)\phi(e_2) \ldots \phi(e_m)$ is a cycle of $<_{X'} \cup Rev^i_{\geq B}$, with $\phi(e_1)\, Rev^1_{\geq B_1+1}\, \phi(e_2)$ and $e_j\, Rev^i_{B_j}\, e_{j+1} \Rightarrow \phi(e_j)\, Rev_{\geq B_j}\, \phi(e_{j+1})$ for pairs (e_j, e_{j+1}) in Rev_{B_j} with $j \neq 1$. In order to validate the inductive proof of Prop. 4, it remains to show that one can choose $h \in \{1, \ldots, K\}$ such that the associated graph X' is acyclic (up to short circuits $e \longleftrightarrow e'$), and if needed is compatible with strong FIFO.

The graph formed by connecting X'_1 and X'_2 with $\longleftrightarrow$ is acyclic because $\longleftrightarrow$ concerns only one process mp. Hence without loss of generality, we can write any cycle in X' in the form $\phi(e'_1)\phi(e'_2) \ldots \phi(e'_{2l})$ with $\{e'_1, \cdots, e'_{2l}\} \subseteq E = \{e_1 \cdots e_m\}$, and where $\forall j$, $e'_{2j} \rightarrow e'_{2j+1}$ and $(\phi(e'_{2j+1}), \phi(e'_{2j+2}))$ belongs to the reflexive and transitive closure of the union of $<_{X'_1}$, $<_{X'_2}$ and $\longleftrightarrow$. In particular, any e'_i belongs either to $\alpha^1, \alpha^2, \gamma^1$ or γ^2. Lemma 3 (that we prove below) claims that one can choose $h \in \{1, \ldots, K\}$ such that $\rightarrow \cap (e_j, e_{j+1})_{j \leq m}$ does not conflict with the reflexive and transitive closure of the union of $<_{X'_1}$, $<_{X'_2}$ and $\longleftrightarrow$ (that is we cannot have (e, e') in this transitive closure and $e' \rightarrow e$). Therefore, for the considered h, $(X', <_{X'})$ is an acyclic graph, compatible with strong FIFO if needed.

★ *Proof of Lemma 3 in the weak FIFO setting.*

Let $(\alpha^1, \alpha^2)(\beta^1_1, \beta^2_1) \cdots (\beta^1_{K_1}, \beta^2_{K_1})(\gamma^1, \gamma^2)$ be a synchronized path of G_1, G_2. We are interested in the order relation between events in $\alpha^1, \alpha^2, \gamma^1$ and γ^2. As G_1 is safe and atomic, and all factors β^1_j are loops in G_1, for all $p, q \in \mathcal{P}$ and $m \in \mathcal{M}_1$, the number of emission events $p\,!\,q\,(m)$ in excess over reception events $q\,?\,p\,(m)$ in $\alpha^1 \cdot \beta^1_1 \ldots \beta^1_j$ does not depend on j for $0 \leq j \leq K_1$. Let $N(p, q, m)$ denote this number. Analogously, the number of reception events $q\,?\,p\,(m)$ in excess over emission events $p\,!\,q\,(m)$ in $\beta^1_j \cdot \beta^1_{j+1} \ldots \gamma^1$ does not depend on j for $1 \leq j \leq K_1 + 1$, and it is equal to $N(p, q, m)$, since G_1 is safe (hence X_1 is an MSC). Moreover $N(p, q, m) \leq (|G_1|/2)$. Similar remarks apply to G_2.

For all $0 < j \leq n \leq K_1$, the dependence relation $\mathcal{O}(j, n)$ between the events from $(\alpha^1, \alpha^2)(\beta^1_1, \beta^2_1) \cdots (\beta^1_{j-1}, \beta^2_{j-1})$ and from $(\beta^1_{n+1}, \beta^2_{n+1}) \cdots (\beta^1_{K_1}, \beta^2_{K_1})(\gamma^1, \gamma^2)$ induced from $(\beta^1_j, \beta^2_j) \cdots (\beta^1_n, \beta^2_n)$ may therefore be represented as a relation on partially ordered sets of generic events Δ and Δ' as follows. Let Δ be the set of events (p, i) and (p, q, m, l) for all $p, q \in \mathcal{P}$, $i \in \{1, 2\}$, $m \in \mathcal{M}_1 \cup \mathcal{M}_2$, and $l \leq N(p, q, m)$. The event (p, i) stands for the last event on process p in $\alpha^i \beta^i_1 \cdots \beta^i_{j-1}$, and the events $(p, q, m, 1) < \cdots < (p, q, m, N(p, q, m))$ stand for the $N(p, q, m)$ last emissions $p!q(m)$. We also let $(p, q, m, l) < (p, i)$ for all p, q, m, l with m a message type from component i, but (p, q, m, l) and (p, q', m', l') are unordered as soon as $q \neq q'$ or $m \neq m'$. Similarly, let Δ' be the partially ordered set of events $(p,i)'$ and $(p,q,m,l)'$, where $(p,i)'$ stands for the first event on process p in $\alpha^i\beta_1^i\cdots\beta_{j-1}^i$, and the events $(p,q,m,1)' < \cdots < (p,q,m,N(p,q,m))'$ stand for the $N(p,q,m)$ first receptions $q?p(m)$. In the same way, we let $(p,i)' < (p,q,m,l)'$ for all p,q,m,l with m a message type from component i.

For $0 < j \leq n \leq K_1$, let $\mathcal{O}(j,n)$ be the binary relation on $\Delta \times \Delta'$ such that $(\delta,\delta') \in \mathcal{O}(j,n)$ if the event δ from Δ is smaller than the event δ' from Δ' in $\Delta(\beta_j^1,\beta_j^2)\cdot\ldots\cdot(\beta_n^1,\beta_n^2)\Delta'$. In order to complete establish the proof of Lemma. 3, it suffices now to show that:

- $\mathcal{O}(1,h-1) = \mathcal{O}(1,h)$ for some $h < K_1$,
- $\mathcal{O}(j,n) = \mathcal{O}(j,l)\circ\mathcal{O}(l+1,n)$ for $j < l < n$, where $\mathcal{O}(j,l)\circ\mathcal{O}(l+1,n)$ means $\mathcal{O}(j,l)\circ Id'\circ\mathcal{O}(l+1,n)$ with $Id' : \Delta' \to \Delta : Id'(e') = e$.

These relations entail $\mathcal{O}(1,K_1) = \mathcal{O}(1,h-1)\circ\mathcal{O}(h,K_1) = \mathcal{O}(1,h)\circ\mathcal{O}(h,K_1)$. Therefore, for any pair of two component CMSCs Y,Z, the order between the events from Y and Z is the same in $Y\cdot(\beta_1^1,\beta_1^2)\cdots(\beta_{K_1}^1,\beta_{K_1}^2)\cdot Z$ as in $Y\cdot(\beta_1^1,\beta_1^2)\cdots(\beta_{h-1}^1,\beta_{h-1}^2)\cdot(\beta_h^1,\beta_h^2)\cdot(\beta_h^1,\beta_h^2)\cdot(\beta_{h+1}^1,\beta_{h+1}^2)\cdots(\beta_{K_1}^1,\beta_{K_1}^2)\cdot Z$. This establishes the claim at the end of the induction step in the proof of Prop. 4.

The relation $\mathcal{O}(j,n) = \mathcal{O}(j,l)\circ\mathcal{O}(l+1,n)$ is obvious, once it has been noticed that an event $p\,!\,q(m)$ emitted but not received in $(\beta_j^1,\beta_j^2)\ldots(\beta_l^1,\beta_l^2)$ is linked with a matching $q\,?\,p(m)$ received but not emitted in $(\beta_{l+1}^1,\beta_{l+1}^2)\ldots(\beta_n^1,\beta_n^2)$ by $p\,!\,q(m) < (p,q,m,i)$ and $(p,q,m,i) < q\,?\,p(m)$ for some $i \leq N(p,q,m)$. Finally, for all $j < k$, $\mathcal{O}(1,j) \subseteq \mathcal{O}(1,k)$, and the maximal length of an increasing chain of binary relations on $\Delta\times\Delta'$ containing Id' is strictly bounded by $K_1 = |\Delta||\Delta'| \leq [2\,|\mathcal{P}| + (|\mathcal{P}|)^2/2\times(|M_1|\times|G_1|+|M_2|\times|G_2|)]^2$. Therefore, $\mathcal{O}(1,h-1) = \mathcal{O}(1,h)$ for some $h \leq K_1$.

★ *Proof of Lemma 3 in the Strong FIFO case.*

Let X_1, X_2 be the MSCs generated from the paths $\alpha^1\beta_1^1\cdots\beta_{K_2K_3}^1\gamma^1$ and $\alpha^2\beta_1^2\cdots\beta_{K_2K_3}^2\gamma^2$ respectively. We make packets of K_3 consecutive loops $\beta_i'^j = \beta_{K_3(i-1)+1}^j\cdots\beta_{K_3i}^j$, $j = 1,2$, and denote by $\beta_i' = (\beta_i'^1,\beta_i'^2)$ the sequence of synchronized loops $(\beta_{K_3(i-1)+1}^1,\beta_{K_3(i-1)+1}^2)\cdots(\beta_{K_3i}^1,\beta_{K_3i}^2)$. Adapting the proof of the weak FIFO case, there exists at most $(2\,|\mathcal{P}| + (|\mathcal{P}|)^2/2\times(\,|G_1| + |G_2|))^2$ synchronized loops $(\beta_i')_{i\in I}$ such that $\mathcal{O}(1,i-1) \neq \mathcal{O}(1,i)$. By monotonicity of $\mathcal{O}(1,i)$, we can iterate (or delete) any subsequence of synchronized loops $(\beta_s^1,\beta_s^2)\cdots(\beta_t^1,\beta_t^2)$ of β_i' without affecting $\mathcal{O}(1,i$, whenever $\mathcal{O}(1,i-1) = \emptyset(1,i)$.

Let $\ell \geq 0$ and $i \geq 1$, and X_1', X_2' be the MSCs generated from $\alpha^1\beta_1^1\cdots(\beta_i^1)^\ell\cdots\beta_{K_2K_3}^1\gamma^1$ and $\alpha^2\beta_1^2\cdots(\beta_i^2)^\ell\cdots\beta_{K_2K_3}^2\gamma^2$ respectively. Assume that the resulting order $<$ on (X_1,X_2) is not compatible with strong FIFO. As X_1' and X_2' are strong FIFO, there exists $\mu(s) = r$ on some channel c in X_1' and $\mu(s') = r'$ on the same channel c in X_2', such that e.g. $s < s'$ and $r' < r$. There may or not exist $e,e' \in E$ with $s < e < e' < s'$ or $r' < e < e' < r$. E.g., assume that there exists $e_i, e_j \in E$ with $s < e_i < e_j < s'$. Let s_2' be the first emission of X_2' on c such that $e_j < s_2'$, and s_2 be the last emission of X_1' on c such that $s_2 < e_i$, and denote by r_2, r'_2 their matching receptions. We have $s_2 < e_i < e_j < s'_2$. We also have $s < s_2$ and $s'_2 < s'$, and since X'_1, X'_2 are strong FIFO, we have $r'_2 < r' < r < r_2$. That is, the messages $(s_2, r_2); (s'_2, r'_2)$ show that X' is not strong FIFO. We call f a *critical event* of $X'_1 \|_{\text{mp}} X'_2$ if f is the first (resp. or last) emission or reception events of its component on its channel after (resp. before) every $e_i \in E$. That is, if there exist matching messages $(s, r), (s', r')$ and events $e, e' \in E$ with $r' < r$ and $s < e < e' < s'$; or $s < s'$ and $r' < e < e' < r$, then we can choose s, r, s', r' in the set of critical events and in the set of events matching critical events. We call (s, r) a critical message if $e \in \{s, r\}$ is critical, and we call it the critical message associated with e. If we iterate or delete a synchronized loop while keeping the same critical messages, and their events are ordered in the same way, then since the original order (without loop iteration) is compatible with strong FIFO, there does not exist critical messages $(s, r), (s', r')$ and events $e, e' \in E$ with $r' < r$ and $s < e < e' < s'$; or $s < s'$ and $r' < e < e' < r$; that is there does not exist messages $(s, r), (s', r')$ and events $e, e' \in E$ with $r' < r$ and $s < e < e' < s'$; or $s < s'$ and $r' < e < e' < r$. There are at most $2|E| \times |\mathcal{P}|^2$ critical events. If $\mathcal{O}(1, i-1) = \mathcal{O}(1, i)$ (that is β_i does not create new order with respect to events in α^1, α^2), iterating the synchronized loop β'_i does not change the first event of each type after any distinguished event $e \in E$. In the same way, if $\mathcal{O}(i, K_2K_3) = \mathcal{O}(i+1, K_2K_3)$, then iterating the synchronized loop β'_i does not change the last event of each type before any distinguished event $e \in E$. Adapting the proof for weak FIFO, there may be at most $2(2\,|\mathcal{P}| + (|\mathcal{P}|)^2/2 \times (\,|G_1| + |G_2|))^2$ loops with either $\mathcal{O}(i, K_2K_3) \neq \mathcal{O}(i+1, K_2K_3)$ or $\mathcal{O}(1, i-1) \neq \mathcal{O}(1, i)$. If one wants that iterating β'_i does not change the critical message associated with f, seeing that there are at most $(|G_1| + |G_2|)$ loops containing a send and a receive on the channel of f between f and its matching event, one should exclude at most $2|E||\mathcal{P}|^2(|G_1| + |G_2|)$ other loops. At last, if one wants that iterating β'_i does not change the ordering between the critical events or their matchings, one should exclude at most $4|E| \times |\mathcal{P}|^2(2\,|\mathcal{P}| + (|\mathcal{P}|)^2/2 \times (\,|G_1| + |G_2|))^2$ other loops. On the whole, one excludes less than K_2 loops.

Thus, if $K = K_2K_3$, we can find a synchronized loop β'_i made of K_3 loops $\beta_{K_3(i-1)+1} \cdots \beta_{iK_3}$ that can be iterated (or deleted) without affecting the order between the events of E and such that there does not exist two messages $(s, r), (s', r')$ and events $e, e' \in E$ with $r' < r$ and $s < e < e' < s'$; or $s < s'$ and $r' < e < e' < r$. We assume without loss of generality that $i = 1$, that is β'_1 can be iterated in such a way. The above reasoning does not suffice to establish the lemma since synchronization on shared events taken alone may create incompatibility of $<$ with strong FIFO. In order to establish Lemma 3, it suffices now to show that for any synchronized path $(\alpha^1, \alpha^2)(\beta^1_1, \beta^2_1) \cdots (\beta^1_{K_3}, \beta^2_{K_3})(\gamma^1, \gamma^2)$ and relation $\rightarrow$ on $(\alpha^1 \cup \gamma^1) \times (\alpha^2 \cup \gamma^2)$ compatible with strong FIFO, if $C(< 2|\mathcal{P}|^2)$ channels (p, q, i) are used in the loops $(\beta^1_1, \beta^2_1) \cdots (\beta^1_{K_3}, \beta^2_{K_3})$ and $D(C) = ((|G_1| + |G_2|)^C \times 3^{1/3(|G_1|+|G_2|)^2C^3 + |\mathcal{P}|(|G_1|+|G_2|)|C|^2} \leq K_3$, then there exists $i < j \leq K_3$ such that for any $\ell \geq 0$, the synchronized path $(\alpha^1, \alpha^2)(\beta^1_1, \beta^2_1) \cdots [(\beta^1_i, \beta^2_i) \cdots (\beta^1_j, \beta^2_j)]^\ell \cdots (\beta^1_{K_3}, \beta^2_{K_3})(\gamma^1, \gamma^2)$ together with the relation $\rightarrow$ is compatible with strong FIFO. We establish this fact by induction on C. Remember that incompatibility with strong FIFO may result only from synchronizations on shared events of $[(\beta_i^1, \beta_i^2) \cdots (\beta_j^1, \beta_j^2)]$.

Assume that no channel is used in β_1', then $D(0) = 1$, and any choice of i, j is right. Else, assume that $C+1$ channels are used. If we can find $D(C)$ consecutive synchronized loops $(\beta_i^1, \beta_i^2) \cdots (\beta_{i+D(C)}^1, \beta_{i+D(C)}^2)$ which do not use one of the channels, then we can apply the inductive step. Otherwise, seeing that $D(C+1) \geq D(C) \times (|G_1| + |G_2|) \times 3^{2|\mathcal{P}|(|G_1|+|G_2|)(C+1)+(|G_1|+|G_2|)^2(C+1)^2}$, we consider the $3^{2|\mathcal{P}|(|G_1|+|G_2|)(C+1)+(|G_1|+|G_2|)^2(C+1)^2}$ consecutive synchronized loops $\beta_i'' = (\beta^1_{i\times D(C)\times(|G_1|+|G_2|)}, \beta^2_{i\times D(C)\times(|G_1|+|G_2|)}) \cdots (\beta^1_{(i+1)\times D(C)\times(|G_1|+|G_2|)-1}, \beta^2_{(i+1)\times D(C)\times(|G_1|+|G_2|)-1})$. In each loop $\beta"_i$, we thus find at least $(|G_1| + |G_2|)$ messages sent and received on every channel used. Since $N(p,q) \leq |G_1| + |G_2|$, no message sent before $\beta"_i$ is received after $\beta"_{i+1}$.

Let $j \leq K_3$. Every unmatched reception r of the synchronized loop $(\beta"^1_j, \beta"^2_j)$ is represented as (p,q,k,l) if it is the l-th unmatched reception on channel (p,q) and component $k \in \{1,2\}$. Let $G = \{(p,q,k,l),(p,k) \mid p,q \in \mathcal{P}, k \in \{1,2\}, l < (|G_1| + |G_2|)\}$. For $g \in G$ and x an event of $(\beta"^1_j, \beta"^2_j)$, we let $g < x$ if either

- $g = (p,k)$ and there exists y in $(\beta"^1_j, \beta"^2_j)$ on process p and component k such that $y < x$,
- Or $g = (p,q,k,l)$ and x is after (for $<$) the l-th unmatched reception on channel (p,q) and component k.

In the same way, we define $x = (p,q,k,l)$ if x is this unmatched reception, and $x < (p,q,k,l)$ if x is before (for $<$) this unmatched reception. We define two other relations OS_j, OS_j', on $G \times \{(p,q,k,l)\}$. Let $x\, OS_j\, (p,q,k,l)$ iff $x < (p,q,k,l)$. Moreover, let $x\, OS_j'\, (p,q,k,l)$ iff there exists a message (s,r) sent and received in $(\beta"^1_j, \beta"^2_j)$ on channel (p,q) and component $3-k$ such that $x < s$ and $r < (p,q,k,l)$. Notice that $OS_j' \subseteq OS_j$. There are less than $3^{2|\mathcal{P}|(|G_1|+|G_2|)(C+1)+(|G_1|+|G_2|)^2(C+1)^2}$ possible relations (OS, OS'), so among the $3^{2|\mathcal{P}|(|G_1|+|G_2|)(C+1)+(|G_1|+|G_2|)^2(C+1)^2}$ loops, there are two loops $\beta"_i, \beta"_{j+1}$ with $(OS_i, OS_i') = (OS_{j+1}, OS_{j+1}')$. Now, the synchronized sequence $(\alpha^1, \alpha^2)(\beta"^1_1, \beta"^2_1) \ldots [(\beta"^1_i, \beta"^2_i) \cdots (\beta"^1_j, \beta"^2_j)]^\ell \cdots (\beta"^1_{D(C+1)}, \beta"^2_{D(C+1)})(\gamma^1, \gamma^2)$ with relation $\rightarrow$, is compatible with strong FIFO.

If it was otherwise, there would exist a pair of messages $(s_1, r_1), (s_2, r_2)$ on the same channel p, q such that $s <' s'$ and $r' <' r$, where we denote by $<'$ the order induced from $<^1, <^2, \rightarrow, \leftrightarrow$ when the loop $(\beta"^1_i, \beta"^2_i) \cdots (\beta"^1_j, \beta"^2_j)$ is iterated $\ell \geq 0$ times. Let $<$ denote the order in the original path, without iteration. Since by construction the non compatibility with strong FIFO can come only from synchronizations, it means that the orders $s <' s', r' <' r$ follow the paths, that is r is in a copy of $\beta"_i$, and s in the copy of $\beta"_j$ immediately before this copy of $\beta"_i$ (else the two messages $(s,r), (s',r')$ would belong to the same loop β_k'', and $<$ would not be compatible with strong FIFO). That is, r is an unmatched reception, say $(p,q,1,l)$. The choices for s', r' are either:

- s' is in the copy of $\beta"_j$ and r' is in the copy of $\beta"_i$. Then r' is an unmatched reception, let say $(p,q,2,k)$. Then we have $(p,q,2,k) OS_i (p,q,1,l)$, that is

$(p,q,2,k)OS_{j+1}(p,q,1,l)$, and we reach a contradiction as $s < s'$ and $<$ is not compatible with strong FIFO, a contradiction.

- both s', r' are in the copy of $\beta"_j$, then there exists $r' < x$ in β_j, $y < r$ in β_i with $y \in G$ and either x and y are matching emission and reception or are on the same process, then $yOS_i(p,q,1,l)$, that is $yOS_{j+1}(p,q,1,l)$, that is r' precedes the reception $(p,q,1,l)$ of $(\beta"^1_{j+1}, \beta"^2_{j+1})$ and we have $<$ is not compatible with strong FIFO, a contradiction.
- both s', r' are in the copy of $\beta"_i$, and then there exists $s < x$ in β_j, $y < s'$ in β_i with $y \in G$, and either x and y are matching emission and reception or are on the same process, and $r' < r$ in β_i. Then $yOS'_i(p,q,1,l)$, that is $yOS'_{j+1}(p,q,1,l)$, that is $<$ is not compatible with strong FIFO, a contradiction.

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

In order to conclude that $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ 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 $X \in (X_1 \parallel_{\mathrm{mp}} X_2)$ such that $<_X \cup Rev_B$ contains a cycle. We claim that there cannot exist any function $f : \mathbb{N} \to \mathbb{N}$, possibly depending on G_1 and G_2 but not depending on X_1 nor on X_2, such that X when it exists can be found in the set $Msc(\mathcal{L}in^{f(B)}(X_1 \parallel_{\mathrm{mp}} X_2))$. An illustration is given in Fig. 5: for all n, X_1^n and X_2 are $\exists$-1-bounded, $X^n \in (X_1^n \parallel_{\mathrm{mp}} X_2)$ is not $\exists$-1-bounded, and $Y^n \in Msc(\mathcal{L}in^{n-1}(X_1^n \parallel_{\mathrm{mp}} X_2))$ is $\exists$-1-bounded. Linearizations of products of MSCs are therefore of little help: $X_1 \parallel_{\mathrm{mp}} X_2$ must be analyzed as a set of graphs even though X_1 and X_2 are defined by paths ρ_1 and ρ_2 in G_1 and G_2, hence by linearizations.

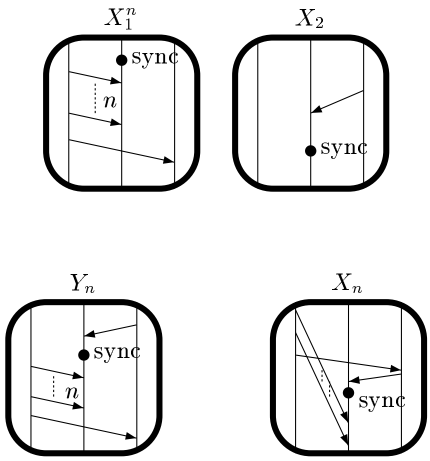

Figure5. Two MSCs Y^n, X^n in the monitored product of X_1^n and X_2

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 $<^i \cup Rev^i_{B^w}$. 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 $Rev^i_{B^w}$ 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 $Rev^i_{B^w}$ 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 $<^1 \cup <^2 \cup \leftrightarrow \cup \rightarrow$, where there is a cycle in $<^1 \cup <^2 \cup \leftrightarrow \cup \rightarrow \cup Rev^i_{B^w}$. 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. We present the construction first in the weak FIFO case since it gives a deterministic algorithm.

★ *Construction of the automaton for Prop. 5 in the weak FIFO setting.*

Since G_1 and G_2 are safe and atomic, the difference between the number of events $p\,!\,q\,(m)$ and $q\,?\,p\,(m)$ varies between 0 and $|G_i|/2$ along any path of G_i. Relying on this crucial property, we construct a *finite* non deterministic state machine that explores all synchronized pairs of paths ρ_1, ρ_2 in G_1 and G_2, selects on the fly a set E of at most $2 \times |\mathcal{P}|$ events $e_1 \dots e_n$, constructs on E a binary relation $\rightarrow \subseteq \cup_i \cup_p (E^i_p \times E^{3-i}_p)$ such that the transitive closure $<$ of $<^1 \cup <^2 \cup \longleftrightarrow \cup \rightarrow$ is acyclic, and keeps in each state the relations $<$ and Rev^i_B on the current set E. Notice that if $Rev_B \neq Rev^i_B \cup Rev^i_B$ as is the case in strong FIFO, then we may not compute Rev_B precisely. However, computing Rev^1_B and Rev^2_B suffices because $Rev^i_B \cup Rev^i_B \subseteq Rev_B \subseteq Rev^i_{B/2} \cup Rev^i_{B/2}$. A state is final if and only if a cycle is found in $(E, < \cup Rev^i_B)$ at this state. By Lemma 2 and Prop. 4, $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ is not $\exists$-bounded if and only if some reachable state is final. The statement of Theorem 2 follows therefore from the finiteness of the construction.

Each state s of the non-deterministic automaton should contain at least:

1. a pair of vertices (v_1, v_2) reached from the initial vertices of G_1, G_2 by synchronized paths ρ_1, ρ_2 such that final vertices of G_1, G_2 may be reached by synchronized continuations of these paths,
2. the set $E = \{e_1, \dots, e_n\}$ of distinguished events that have been selected among all those generated by ρ_1, ρ_2,
3. the restrictions on E of the relations $<$ and Rev^i_B.

The states of the machine should provide enough information to update $(E, <, Rev^i_B)$ when constructing new states from existing states. Assume that a pair of vertices (v_1, v_2) has been reached by synchronized path (ρ_1, ρ_2) and the current state is $(v_1, v_2, E, <, Rev^i_B, \ldots)$ where $E = \{e_1, \ldots, e_n\}$. A new state may result from taking an edge $v_1 \to v_1'$ in G_1, or an edge $v_2 \to v_2'$ in G_2, or two edges $v_1 \to v_1'$ and $v_2 \to v_2'$ if v_1' and v_2' have the same label in $\mathcal{SE}$. In the last case, two synchronized events are generated from v_1' and v_2' on the two copies of the monitor process. Emissions and receptions generated from G_1 or G_2 may be inserted or not in the set of distinguished events $E = \{e_1, \ldots, e_n\}$. Local events, and in particular synchronized events, will never be inserted in this set: they cannot belong to a cycle $\{e_1, \ldots, e_n\}$ since the conditions in Lemma 2 forbid to have three events in E on the same process.

Recall that $<$ is the transitive closure of the union of $<^1$, $<^2$, $\longleftrightarrow$, and $\to$ where relation $\to$ is defined on E while relations $<^1$, $<^2$, and $\longleftrightarrow$ are defined on supersets of E, namely E^1, E^2 and $E^1 \cup E^2$ where E^i denotes the collection of all events generated from path ρ_i in G_i. Also note that $<^1$, $<^2$, $\longleftrightarrow$ are totally determined by paths ρ_1, ρ_2 whereas $\to$ is not. Suppose e.g. that an edge $v_1 \to v_1'$ has been taken in G_1 and one wants to insert the new event e' generated from v_1' into the set of distinguished events $\{e_1, \ldots, e_n\}$. Possibly $e < e'$ for some e in $\{e_1, \ldots, e_n\}$, because $e = f_0 R_1 f_1 \ldots R_k f_k = e'$ for some events $f_1 \ldots f_{k-1}$ outside $\{e_1, \ldots, e_n\}$ and corresponding relations R_j in $\{<^1, <^2, \longleftrightarrow\}$. For each distinguished event $e \in E$, the current state of the non-deterministic machine should therefore display the set $\mathcal{O}_P(e)$ of all pairs (p, k) such that $e < e'$ will hold whenever a new event e' is generated from G_i on process p.

Now, these sets $\mathcal{O}_P(e)$ must in turn be updated whenever new events are generated by moving from v_1 to v_1' or from v_2 to v_2' or both. There may be three reasons for an update: *i)* relation $\longleftrightarrow$ increases as a result of a synchronized move, *ii)* relation $\to$ increases because the event generated from v_i' is selected for insertion into E, and there is already in E a distinguished event generated on the same process p from G_{3-i}, *iii)* the newly generated event is a reception.

Consider case *(iii)*. For any event e generated from G_i on process p, as soon as some message $m \in M_i$ sent after e from process p to process q is received, (q, i) should be inserted in $\mathcal{O}_P(e)$. In order to update $\mathcal{O}_P(e)$ at the right time, one should know from the current machine state, for each process q and for each message $m \in \mathcal{M}_i$, how many events $p\,!\,q\,(m)$, up to and including the first instance after e, have not yet been matched by reception $q\,?\,p\,(m)$. For each *channel* $c = (p, m, q)$, let $\mathcal{O}_S(e)(c)$ denote this number (this notation for channels is not ambiguous since $\mathcal{M}_1$ and $\mathcal{M}_2$ are disjoint sets). Things work as follows. When a distinguished event e generated from G_i on process p is inserted in E, $\mathcal{O}_P(e)$ is initialized with (p, k). At the first time an emission $p\,!\,q\,(m)$ with $m \in \mathcal{M}_i$ is generated after e in ρ_i, the counter $\mathcal{O}_S(e)(c)$ is set to the number of messages stored in the channel $c = (p, m, q)$, including the message produced by this emission. $\mathcal{O}_S(e)(c)$ is decreased by one each time q receives m from p. When $\mathcal{O}_S(e)(p, m, q)$ reaches 0, (q, i) is inserted into $\mathcal{O}_P(e)$. Similar counters

$\mathcal{O}_S(e)(q', m', r')$ are maintained for all processes q' and content $m' \in M_j$ ($j = 1$ or 2) such that $(q', j) \in \mathcal{O}_P(e)$.

For case *(i)*, the update is simple: $(mp, 1)$ is inserted into sets $\mathcal{O}_P(e)$ that contained only $(mp, 2)$ and conversely. Case *(ii)* is a little more delicate. Let $e' = e_{n+1}$ be the new event inserted into E, and let e be the event already present in E such that e resp. e' are on the same process p of G_i resp. G_{3-i}. Both orientations $e \to e'$ or $e' \to e$ are a priori possible. Updating $<$ with $e \to e'$ cannot ever introduce circularity in $<$. In contrast, updating $<$ with $e' \to e$ might result in circularity. *Circularity of relation $<$ must be avoided by explicit checking.*

It remains to consider the updating of relation Rev_B^i on E. Assume that e and e' are two distinguished events in E and $eRev_B^i e'$, hence e and e' are labeled with $q\,?\,p\,(m)$ and with $p\,!\,q\,(m)$, respectively. The event e may have been generated before e' but the converse is also possible. Therefore, for detecting at run time that $eRev_B^i e'$, one must anticipate on these two events. The right time for predicting $eRev_B^i e'$ is when generating the emission f to be matched by the reception e, i.e., the B-th event $p\,!\,q\,(m)$ before e' (the event f needs not be selected for insertion into E). One needs two counters $R_r(c)$ and $R_s(c)$ for channel $c = (p, m, q)$, initialized just after generating f. $R_r(c)$ is initialized with the number of messages stored in channel c immediately after f. $R_s(c)$ is initialized with the value B. $R_r(c)$ and $R_s(c)$ are decreased by one each time a reception $q\,?\,p\,(m)$, resp. an emission $p\,!\,q\,(m)$ is generated. The event e, resp. the event e', is inserted into E when $R_r(c)$, resp. $R_s(c)$, reaches the value 0. The relation $eRev_B^i e'$ is recorded in the current state when e and e' have been both inserted into E.

Whenever Rev_B^i or $<$ is updated, if some cycle appears in $< \cup Rev_B^i$, the current state is declared final, and the construction of new states is stopped, with the diagnostic that $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ is not $\exists$-B-bounded. Otherwise, the construction is pursued until no new state can appear, which must occur sooner or later since the information contained in $(v_1, v_2, E, <, Rev_B, \mathcal{O}_P, \mathcal{O}_S, R_r, R_s)$ is bounded (for all channels c, $\mathcal{O}_S(e)(c)$ and $R_r(c)$ are uniformly bounded by $K = max(|G_1|, |G_2|)/2)$.

A pseudo-algorithmic description of the construction is given hereafter. Let Ch denote the set of channels (p, m, q), where $p, q \in \mathcal{P}$ and $m \in \mathcal{M}_1 \cup \mathcal{M}_2$. For each channel $c = (p, m, q)$, let $head(c) = p$ and $tail(c) = q$. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two disjoint copies of $\mathcal{P}$, with $(p, 1) \in \mathcal{P}_1$ and $(p, 2) \in \mathcal{P}_2$ for all $p \in \mathcal{P}$. $E = \{1, \ldots, n\}$ stands for the set of distinguished events, hence $n' := n + 1$ means the insertion of a new event in E, and for convenience, we let n' denote this new event. Finally, $K = max(|G_1|, |G_2|)/2$.

A *state* is a tuple $s = (v_1, v_2, count, n, P, rev, <, \mathcal{O}, \mathcal{R})$ as follows:

- v_i is a vertex of the CMSC-graph G_i $(i = 1, 2)$,
- $count : Ch \to \{0, \cdots, K\}$ counts the messages stored in each channel,
- $n \leq 2.|\mathcal{P}|$ counts the distinguished events,

- $P : E \to \mathcal{P}_1 \cup \mathcal{P}_2$ indicates for each distinguished event on what process and from which component G_1 or G_2 it was generated,
- $<\subseteq E \times E$ is an order relation,
- $rev : E \cup Ch \to E \cup Ch$ is a partial function interpreted as follows: $rev(c) = c$ for $c = (p, m, q)$ means that two events e and e' such that $eRev_B e'$ are expected at both ends of channel c, $rev(e) = c$ means that an event e' such that $eRev_B e'$ is expected on p, etc... ,
- $\mathcal{O} : e \to (\mathcal{O}_P(e), \mathcal{O}_S(e))$ where $e \in E$, $\mathcal{O}_P(e) \subseteq \mathcal{P}_1 \cup \mathcal{P}_2$ and $\mathcal{O}_S(e) : Ch \to \{0, 1, \cdots, K, +\infty\}$,
- $\mathcal{R} : c \to (\mathcal{R}_r(c), \mathcal{R}_s(c))$ where $c \in Ch$, $\mathcal{R}_r(c) \in \{0, \cdots, K, +\infty\}$ and $\mathcal{O}_s(c) \in \{0, \cdots, B, +\infty\}$,

The components of the initial state are the vertices v_1^0, v_2^0 and $+\infty$, $\mathbf{0}$ or $\emptyset$ for all the rest. A new state $s' = (v_1', v_2', count', n', P', rev', <', \mathcal{O}', \mathcal{R}')$ is constructed from $s = (v_1, v_2, count, n, P, rev, <, \mathcal{O}, \mathcal{R})$ if and only if it may be produced by the following pseudo-algorithm (by default $x' = x$ for all state components x):

1. *choose edges of CMSC-graphs*
   $v_i \to_i v_i'$ in G_i for $i = 1$ or $i = 2$.
   If the label of v_i does not belong to $\mathcal{SE}$, $v_{3-i}' = v_{3-i}$.
   Otherwise, $v_{3-i} \to_{3-i} v_{3-i}'$ such that v_i' and v_{3-i}' have the same label; for each $e \in E$ such that $(mp, 1)$ or $(mp, 2)$ belongs to $\mathcal{O}_P(e)$, $\mathcal{O}_P'(e) = \mathcal{O}_P(e) \cup \{(mp, 1), (mp, 2)\}$.
   If no synchronized paths from v_1', v_2' can reach final vertices in G_1, G_2, no new state s' is produced.

2. *if v_i' is an emission on channel $c = (p, m, q)$*
   $count'(c) = count(c) + 1$, $n' = n$ or $n' = n + 1$.
   $\mathcal{O}_S'(e)(c) = min(\mathcal{O}_S(e)(c), count'(c))$ for every $e \in E$ such that $(head(c), j) \in \mathcal{O}_P(e)$ for $j = 1$ or 2 (with $head(p, m, q) = p$ and $tail(p, m, q) = q$).
   Let $\mathcal{R}(c) = (R_r, R_s)$. $\mathcal{R}'(c) = (R_r, R_s - 1)$ (with 0-1=0).
   If $\mathcal{R}(c) = (+\infty, +\infty)$ then $\mathcal{R}'(c) = (+\infty, +\infty)$ or $\mathcal{R}'(c) = (count'(c), B)$ and $rev'(c) = c$.
   If $R_s = 1$ then $n' = n + 1$ and according to the case: if $rev(c) = c$ then $rev'(c) = n'$, else $rev'(e) = n'$ for the (unique) $e \in E$ such that $rev(e) = c$.
   Finally, if $n' = n + 1$ then $P'(n') = i$ and $\mathcal{O}_P'(n') = \{(p, i)\}$.

3. *if v_i' is a reception on channel $c(p, m, q)$*
   $count'(c) = count(c) - 1$, $n' = n$ or $n' = n + 1$.
   $\mathcal{O}_S'(e)(c) = \mathcal{O}_S(e)(c) - 1$ for every $e \in E$ such that $(head(c), j) \in \mathcal{O}_P(e)$ for $j = 1$ or 2.
   Let $\mathcal{R}(c) = (R_r, R_s)$. $\mathcal{R}'(c) = (R_r - 1, R_s)$ (with 0-1=0).
   If $R_r = 1$ then $n' = n + 1$ and according to the case: if $rev(c) = c$ then $rev'(n') = c$, else $rev'(n') = e$ for the (unique) $e \in E$ such that $rev(c) = e$.
   For every $e \in E$, if $\mathcal{O}_S(e)(c) = 1$ and $\mathcal{O}_S'(e)(c) = 0$ then
   $\mathcal{O}_P'(e) = \mathcal{O}_P(e) \cup \{tail(c), i\}$

Finally, if $n' = n + 1$ then $P'(n') = i$ and $\mathcal{O}'_P(n') = \{(q,i)\}$.

4. *if* v'_i *or* v'_i *and* v'_{3-i} *are internal events.* In this case, $s' = s$.

5. *update* $<$
If $n' = n$ there is nothing to do. Assume $n' = n + 1$ and $P'(n') = (p,i)$. If there are already two events e in E with $P(e) = (p,i)$ or $(p, 3-i)$, no new state s' is generated. Otherwise, let $e <' n'$ for all events $e \in E$ such that $(p,i) \in \mathcal{O}_P(e)$. If there is one event e in E such that $P(e) = (p, 3-i)$ and $(p,i) \notin \mathcal{O}_P(e)$ then let
   - either $e <' n'$
   - or $n' <' e$ *if this does not create circularity*

   Close transitively $<'$. Add $\mathcal{O}'_P(e)$ to $\mathcal{O}'_P(e')$ whenever $e <' e'$. For every channel c and event $e \in E$, replace $\mathcal{O}'_S(e)(c)$ with the least defined value of $\mathcal{O}'_S(e')(c)$ for $e \leq' e'$.
   Check $<' \cup rev'$ for non circularity before generating s'.

★ *Proof of Prop. 5 in the Strong FIFO settings*

In the Strong FIFO case, as we compute Rev^i_B and not Rev_B, we consider $\{(p,q,i)/p,q \in \mathcal{P}, i \in \{1,2\}\}$ as the set of channels. We also consider (p,i) as the set of processes. Notice that $count(c)$ is still a number and not a word, since the next reception on q from p on any path of a safe CMSC-graph always matches the first unmatched emission on p to q. Thus, the number $rev(p,q,i)$ that we compute is exactly $Rev^i_B(p,q)$.

Compared to weak FIFO, we have to check that the product MSC which is generated and the set of distinguished events E are compatible with strong FIFO. We now show that it suffices to consider a polynomial number of events F and the order between them, such that if the restriction of the order to F is compatible with strong FIFO, then the order is compatible with strong FIFO. By contradiction, since both G_1 and G_2 are assumed to be strong FIFO, assume that there are two messages, (s,r) on channel $(p,q,1)$ and (s',r') on channel $(p,q,2)$, with $s < s', r' < r$ imposed by $\leftrightarrow \cup \rightarrow$.

Assume that at the time when r is seen on the path, we have the two relations $s < s', r' < r$. It means that at that time, $s_2 = s$ was an unmatched emission, and we denote $r_2 = r$. We also have $r' \leq x \leq r_2$ with $P(x) = P(r_2)$. Taking r'_2 the last receive of type $\lambda(r')$ such that there exists y with $r'_2 \leq y \leq r_2$ and $P(y) = P(r_2)$, and s'_2 its matching emission, we have $s_2 = s < s' \leq s'_2$ since G_2 is strong FIFO and $r' \leq r'_2$. That is, $(s_2, r_2), (s'_2, r'_2)$ is a proof that the order is not compatible with strong FIFO.

There must be two event $e_i, e_j \in E$ with $e_i \rightarrow e_j$ such that for instance $s < e_i$ and $e_j < s'$. Now, consider the last emission s_2 before e_i of type $\lambda(s)$ and the first emission s'_2 after e_j of type $\lambda(s')$. Thus, $s < s_2$ and $s'_2 < s'$. Since G_1 and G_2 are strong FIFO, the receptions r_2, r'_2 matching s_2, s'_2 satisfy $r < r_2$ and $r'_2 < r'$, hence (s_2, r_2) and (s'_2, r'_2) contradicts strong FIFO.

Therefore, it suffices to consider at each state the set F of events that are either the first of their type after some $e_i \in E$, or the last of their type before some $e_i \in E$ or before any event of some process p, or their matching events of the above events, or unmatched emissions. This set is of polynomial size. The automaton guesses whether an event belong to F at its creation. Keeping a polynomial size information, the automaton can check that no guess was wrong on the current path.

★ *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 $G_1 \|_{\mathrm{mp}} G_2$ is $\exists$-B-bounded iff ϕ is non satisfiable. We let $mp = p$.

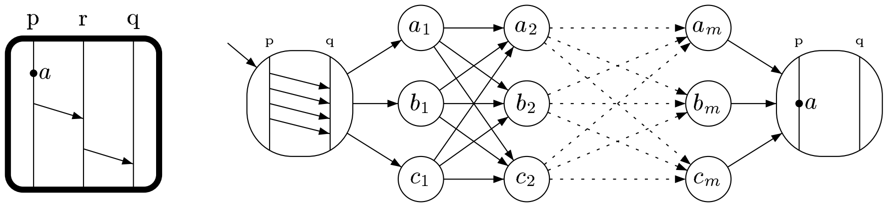

Figure6. MSC M_1 and MSC-graph G_2

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. Assume that two choices of literals are conflicting, that is we choose v_j for clause i and $\neg v_j$ for clause i'. If $i < i'$, then all the receptions on q from p (in G_2) are before a, through the sequence of messages q to p_j and p_j to p. If $i' < i$, then we have a sequence of messages q to p'_j and p'_j to p, implying the same dependency. If it is the case, the two messages in G_1

cannot mix with the B messages from p to q in G_2, and the MSC obtained is existentially K-bounded, for every $K \geq 1$.

On the other hand, if no two choices conflict (ϕ is satisfiable), there is no dependency between the receive on q from p and a, hence the first message from p to q (in G_2) can be received after the reception of the message from r to q (in G_1), hence after the synchronization a, hence after the emission of the last message from p to q (in G_2), and the MSC is not existentially B bounded. In order to show the hardness of checking existential-boundedness, it suffices to replace the MSCs labeling the initial node of G_2 by a self loop with a message from p to q.

7 CMSC-graph representation of a Monitored Product

In the case where $\mathcal{L}(G_1) \parallel_{\mathrm{mp}} \mathcal{L}(G_2)$ 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 $\mathcal{L}in^B(\mathcal{X})$ 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 $v_1 \ldots v_n$ in G, all CMSCs in the set $\Lambda(v_1) \cdot \ldots \cdot \Lambda(v_n)$ have connected communication graphs.

The communication graph induced by $X = (E, \lambda, \mu, (<_p)_{p \in \mathcal{P}})$ 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,
– $\mathcal{L}in^B(\mathcal{X})$ is a regular representative set for $\mathcal{X}$ for sufficiently large $B > 0$. Moreover, B and a finite automaton recognizing $\mathcal{L}in^B(\mathcal{X})$ can be computed effectively from G. Conversely, G can be computed effectively from $\mathcal{L}in^B(\mathcal{X})$.

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.

Theorem 4 ([4]). *Let $\mathcal{X}$ be a set of MSCs. Provided that $\mathcal{M}$ contains exactly one message and there are no internal events, the following assertions are equivalent:*
– $\mathcal{X} = \mathcal{L}(G)$ for some globally cooperative CMSC-graph G,
– $\mathcal{L}in^B(\mathcal{X})$ is a regular representative set for $\mathcal{X}$ for sufficiently large $B > 0$.
Moreover, B and a finite automaton recognizing $\mathcal{L}in^B(\mathcal{X})$ can be computed effectively from G. Conversely, G can be computed effectively from $\mathcal{L}in^B(\mathcal{X})$.

We show that Theorem 4 extends to the case where $\mathcal{M}$ is a finite set of messages, and that it also stays valid when internal events are added.

Given finite sets $\mathcal{P}$, $\mathcal{M}$, and $\mathcal{A}$ (of processes, messages, and internal actions, respectively), let $X = (E, \lambda, \mu, (<_p)_{p\in\mathcal{P}})$ be a CMSC. We will transform X into a CMSC X' over a larger set of processes $\mathcal{P}'$ such that X' is a pure CMSC according to the following definition.

Definition 8. *A CMSC is* pure *if it has no internal events and all messages have an empty content (that can therefore be omitted).*

Definition 9. *Let $\mathcal{P}'$ be the union of $\mathcal{P}$ and the sets $\{p\,(a) \mid p \in \mathcal{P},\, a \in \mathcal{A}\}$, $\{p\,!\,q\,(m) \mid p, q \in \mathcal{P},\, p \neq q,\, m \in \mathcal{M}\}$ and $\{q\,?\,p\,(m) \mid p, q \in \mathcal{P},\, p \neq q,\, m \in \mathcal{M}\}$. Define $X' = (E', \lambda', \mu', (<'_{p'})_{p'\in\mathcal{P}'})$ as the (pure) CMSC with components as follows (see Fig. 7).*
– Each internal event $e \in E$ with label $p(a)$ is replaced in E' by four events e, e', e'', $\overline{e}$. The events e and $\overline{e}$ belong to the process p, and e is the immediate predecessor of $\overline{e}$ according to $<'_p$. The events e' and e'' belong to the process $p\,(a)$, and e' is the immediate predecessor of e'' according to $<'_{p(a)}$. Moreover, we let $\mu'(e) = e'$ and $\mu'(e'') = \overline{e}$, hence the labels of these events are respectively $\lambda'(e) = p\,!\,(p\,(a))$, $\lambda'(e') = (p\,(a))\,?\,p$, $\lambda'(e'') = (p\,(a))\,!\,p$, $\lambda'(\overline{e}) = p\,?\,(p\,(a))$.
– Each emission $e \in E$ with label $p\,!\,q\,(m)$ is replaced in E' by three events e, e', e''. The event e belongs to the process p. The events e' and e'' belong to the process $p\,!\,q\,(m)$, and e' is the immediate predecessor of e'' according to $<'_{p\,!\,q\,(m)}$. Moreover, we let $\mu'(e) = e'$, hence $\lambda'(e) = p\,!\,(p\,!\,q\,(m))$ and $\lambda'(e') = (p\,!\,q\,(m))\,?\,p)$. We let $\lambda'(e'') = (p\,!\,q\,(m))\,!\,(q\,?\,p\,(m))$.
– Each reception $e \in E$ with label $p?q\,(m)$ is replaced in E' by three events e, e', e''. The event e belongs to the process p. The events e' and e'' belong to the process $p?q\,(m)$, and e' is the immediate predecessor of e'' according to $<'_{p\,?q\,(m)}$. Moreover, we let $\mu'(e'') = e$, hence $\lambda'(e'') = (p\,?\,q\,(m))\,!\,p$ and $\lambda'(e) = p\,?\,(p\,?\,q\,(m))$. We let $\lambda'(e') = (p\,?\,q\,(m))\,?\,(p\,!\,q\,(m))$.
– For any $e_1, e_2 \in E$, we let $\mu'(e''_1) = e'_2$ if $\mu(e_1) = e_2$ in X.
– Finally, for any $p' \in \mathcal{P}'$, two events of the process p' are in the relation $<'_{p'}$ if they have been derived respectively from two events in E in the relation $<_p$ for some $p \in \{\mathcal{P}\}$.

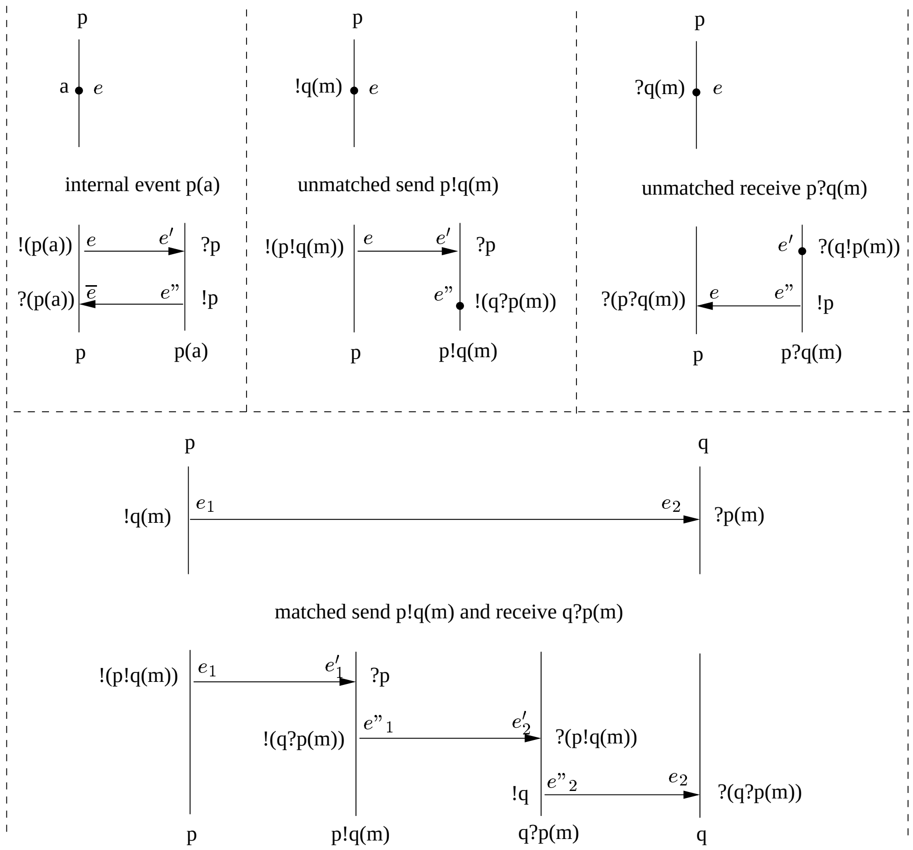

Figure7. Transforming CMSCs into pure CMSCs

Any potential circuit in the relation $<'$ (induced from μ' and $<_{p'}$ for all $p' \in \mathcal{P}'$) must result from some circuit in the similar relation $<$ in X, hence X' is a CMSC. Clearly, X' is a MSC if and only if X is an MSC. Moreover, in this case, the bounded representations of X and X' may be set in correspondence as follows.
– $\mathcal{L}in^B(X')$ is representative of $\{X'\}$ if and only if $\mathcal{L}in^{3B}(X)$ is representative of $\{X\}$.
– the B-bounded representations of X' rewrite onto the $3B$-bounded representations of X through the following *simplification rules:*

1. $p\,!\,(p(a)) \rightarrow p(a)$
2. $p\,!\,(p\,!\,q(m)) \rightarrow p\,!\,q(m)$
3. $p\,?\,(p\,?\,q(m)) \rightarrow p\,?\,q(m)$
4. all other labels are rewritten to ε.

Given a CMSC-graph $G = (V, \rightarrow, \Lambda, V^0, V^f)$, define now $G' = (V, \rightarrow, \Lambda', V^0, V^f)$ with $\lambda'(v) = (\lambda(v))'$ for all vertices v. Then G' is a CMSC-graph, and clearly, G' is globally cooperative if and only if G is globally cooperative. We are ready to prove that Theorem 4 extends to sets $\mathcal{X}$ of possibly impure MSCs.

The rest of the section is the proof of Theorem 3.

★ *Suppose $\mathcal{X} = \mathcal{L}(G)$ for some globally cooperative CMSC-graph G.*

Let G' be defined as above, and let $\mathcal{X}' = \mathcal{L}(G')$. As G' is globally cooperative, by Theorem 4, for some $B > 0$, $\mathcal{L}in^B(\mathcal{X}')$ is a regular representative set for $\mathcal{X}'$. The image of $\mathcal{L}in^B(\mathcal{X}')$ under the simplification rules is the set of all $3B$-bounded representations of MSCs X in $\mathcal{X}$. As the image of a regular set under an alphabetic morphism, this set is regular, and it is a representative set for $\mathcal{X}$, since for each $X \in \mathcal{X}$, $\mathcal{L}in^{3B}(X)$ is representative of $\{X\}$.

★ *Suppose $\mathcal{L}in^{3B}(\mathcal{X})$ is a regular representative set for $\mathcal{X}$ for some $B > 0$.*

Consider the MSC language $\mathcal{X}' = \{X' \mid X \in \mathcal{X}\}$.

Lemma 4. *$\mathcal{L}in^B(\mathcal{X}')$ is a representative set for $\mathcal{X}'$.*

Proof. Suppose for contradiction that some MSC $X' \in \mathcal{X}'$ has no B-bounded representation. Let $X' = (E', \lambda', \mu', (<'_{p'})_{p' \in \mathcal{P}'})$. By Prop. 3, there is a cycle in the relation $<' \cup Rev'_B$. Consider a minimal cycle. As X' has been produced from $X = (E, \lambda, \mu, (<_p)_{p \in \mathcal{P}})$ in $\mathcal{X}$ as defined in Def. 9, this cycle may be decomposed into the two types of segments (of length 6) which are depicted in Fig. 8 (the fat right-to-left arrows are occurrences of Rev'_2).
It should be clear that whenever e'_1 to e'_6 are joined by a cascade of Rev'_B as shown in the right part of this figure, the inverse images of e'_1 and e'_6 in E (under the embedding of E into E') are joined by $Rev_{B'}$ (in X) for some $B' \geq 3B$. Therefore, the cycle in the relation $<' \cup Rev'_B$ induces a cycle in the similar relation $< \cup Rev_{B'}$ (in X) for some $B' \geq 3B$. This enters in contradiction with the assumption that $\mathcal{L}in^{3B}(\mathcal{X})$ is a regular representative set for $\mathcal{X}$. □

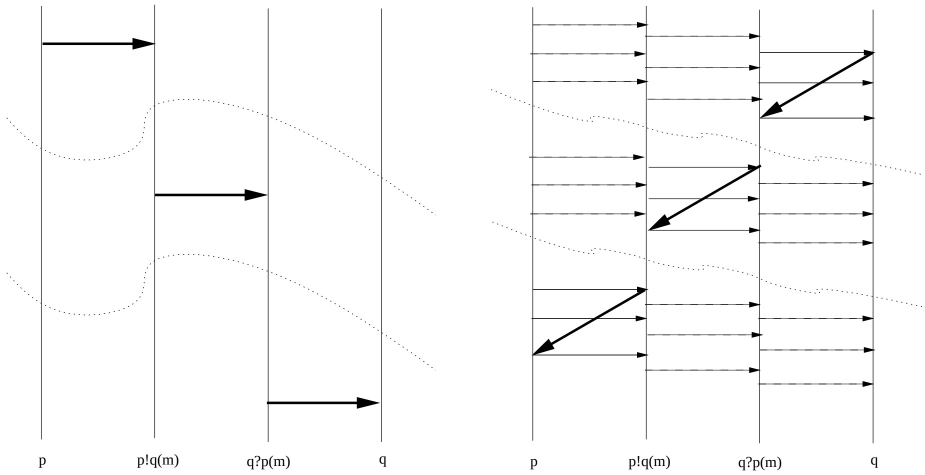

Figure8.

Lemma 5. *$\mathcal{L}in^B(\mathcal{X}')$ is a regular language.*

Proof. Let $\mathcal{E}' = \lambda'(E')$, then $\mathcal{L}in^B(\mathcal{X}')$ is the set of all words $w' \in (\mathcal{E}')^*$ for which the following requirements are fulfilled:
– the inverse image of w' under the simplification rules belongs to $\mathcal{L}in^{3B}(\mathcal{X})$,
– for any $p', q' \in \mathcal{P}'$, the projection of w' on $\{(p' \,!\, q'), (q' \,?\, p')\}^*$ is a B-bounded MSC representation,
– for any $p \in \mathcal{P}$ and $a \in \mathcal{A}$, the projection of w' along the process p (resp. $p(a)$) belongs to the language $(\alpha\beta + \mathcal{E}' \setminus \{\alpha, \beta\})^*$ where $\alpha = (p \,!\, p(a))$ and $\beta = (p \,?\, p(a))$ (resp. $\alpha = (p(a) \,?\, p)$ and $\beta = (p(a) \,!\, p)$,
– for any $p, q \in \mathcal{P}$ and $m \in \mathcal{M}$, the projection of w' along the process $p \,!\, q\,(m)$ (resp. $p \,?\, q\,(m)$) is in the language $(\alpha\beta + \mathcal{E}' \setminus \{\alpha, \beta\})^*$ where $\alpha = (p \,!\, q\,(m)) \,?\, p$ and $\beta = (p \,!\, q\,(m)) \,!\, (q \,?\, p\,(m))$ (respectively, where $\alpha = (p \,?\, q\,(m)) \,?\, (q \,!\, p\,(m))$ and $\beta = (p \,?\, q\,(m)) \,!\, p$.
As an intersection of regular languages, $\mathcal{L}in^B(\mathcal{X}')$ is therefore regular. □

In view of Lemmas 4 and 5, and by Theorem 4, $\mathcal{X}' = \mathcal{L}(H)$ for some globally cooperative CMSC-graph $H = (V, \rightarrow, \Lambda', V^0, V^f)$. One can easily produce from H a CMSC-graph $G = (V, \rightarrow, \Lambda, V^0, V^f)$ such that $\mathcal{X}' = \{X' \mid X \in \mathcal{L}(G)\}$. It suffices, for each vertex v, to let $\Lambda(v) = U(\Lambda'(v))$ where $U(X')$ is constructed from X' as follows:
– suppress all processes not in $\mathcal{P}$,
– suppress all events labeled $p \,?\, (p(a))$,
– apply the simplification rules to the remaining events,
– set $\mu(e_1) = e_2$ for each ordered pair of events e_1 and e_2 matching the pattern at bottom of Fig. 7.
Note that for any MSC X, $U(X') = X$ up to isomorphism of MSCs, hence

$(\cdot)' : X \to X'$ is injective on MSCs. Therefore, $\mathcal{X} = \mathcal{L}(G)$.

It remains to show that G is globally cooperative. Let $v_0, v_1 \dots v_n$ be a path in G from some initial vertex $v_0 \in V^0$ to some final vertex $v_n \in V^f$. As H is globally cooperative, there is one MSC Y in $\Lambda'(v_0) \cdot \ldots \cdot \Lambda'(v_n)$. It is readily verified that $U(Y)$ is an MSC and that $U(Y)$ belongs to $U(\Lambda'(v_0)) \cdot \ldots \cdot U(\Lambda'(v_n))$. Therefore, G is safe. Now let $v_1 \dots v_n$ be a circuit in G (hence in H) and let Y be a CMSC in the set $\Lambda'(v_1) \cdot \ldots \cdot \Lambda'(v_n)$.
– As $\mathcal{L}(H) = \mathcal{X}'$, the label $(p!q(m))?p$, resp. $(q?p(m))!q$, occurs in Y if and only if the label $(p!q(m))!(q?p(m))$, resp. $(q?p(m)?(p!q(m))$, occurs in Y.
– As H is globally cooperative and from the first condition in Def. 7, whenever some label $p!(p!q(m))$, or $(p!q(m))!(q?p(m))$, or $(q?p(m))!q$ occurs in Y, some corresponding label $(p!q(m))?p$, or $(q?p(m)?(p!q(m))$, or $q?(q?p(m))$ occurs in Y, and vice versa.
– Therefore, whenever $p!(p!q(m))$ or $q?(q?p(m))$ occurs in Y, both occur in Y and $\{p, p!q(m)\}$, $\{p!q(m), q?p(m)\}$, and $\{q?p(m), q\}$ are edges of the communication graph of Y.
Thus for any CMSC X in the set $U\Lambda'(v_1) \cdot \ldots \cdot U\Lambda'(v_n)$, $p!q(m)$ occurs in X if and only $q?p(m)$ occurs in X, and since the communication graph of Y is connected, the communication graph of X is connected. Therefore, G is globally cooperative.

★ *Completion of the proof of Theorem 3*

We have established heretofore the main part of Theorem 3, namely the equivalence of the two assertions. Now given a globally cooperative CMSC-graph G, if B and $\mathcal{L}in^B(\mathcal{X}')$ are an existential bound and a regular representative set for $\mathcal{X}' = \mathcal{L}(G')$, then $3B$ and the image of $\mathcal{L}in^B(\mathcal{X}')$ under the simplification rules are respectively an existential bound and a regular representative set for $\mathcal{X} = \mathcal{L}(G)$. Finally, the proof that G can be computed from $\mathcal{L}in^{3B}(\mathcal{X})$ is a remake of a similar proof for pure CMSCs and CMSC-graphs. □

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, $\mathcal{L}in^B(\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2))$ is a representative set for $\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)$. By Prop. 2, $\mathcal{L}in^B(\mathcal{L}(G_1) \parallel_{\text{mp}} \mathcal{L}(G_2)) = \mathcal{L}in^B(\mathcal{L}(G_1)) \parallel_{\text{mp}} \mathcal{L}in^B(\mathcal{L}(G_2)) \cap \mathcal{L}in^B$. Since both G_1, G_2 are globally cooperative, both $\mathcal{L}in^B(\mathcal{L}(G_1))$ and $\mathcal{L}in^B(\mathcal{L}(G_2))$ are regular and effectively computable. Since the shuffle of regular language is regular, we get the following.

Theorem 5. *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. R. Alur, K. Etessami, and M. Yannakakis. Realizability and Verification of MSC Graphs. In *ICALP'01*, LNCS **2076**, 797-808, 2001.
2. B. Caillaud, P. Darondeau, L. Hélouët, and G. Lesventes. HMSCs as Partial Specifications ... with PNs as Completions. In *Modeling and Verification of Parallel Processes*, LNCS **2067**, 125-152, 2001.
3. C. Duboc. Mixed Product and Asynchronous Automata. *Theoretical Computer Science* **48**(3): 183-199, 1986.
4. B. Genest, D. Kuske, and A. Muscholl. A Kleene Theorem and Model Checking for a Class of Communicating Automata. *Inf. Comput.*, **204**(6):920-956, 2006.
5. E. Gunter, A. Muscholl, and D. Peled. Compositional Message Sequence Charts. *STTT* **5**(1): 78-89, 2003 and in *TACAS'01*, LNCS **2031**, 496-511, 2001.
6. B. Genest, A. Muscholl, and D. Peled. Message Sequence Charts. In *Lectures on Concurrency and Petri Nets*, LNCS **3098**, 537-558, 2004.
7. B. Genest, A. Muscholl, H. Seidl, and M. Zeitoun. Infinite-state High-level MSCs: Model-checking and Realizability. *JCSS* **72**(4): 617-647, 2006 and in *ICALP'02*, LNCS **2380**, 657-668, 2002.
8. L. Hélouët and C. Jard. Conditions for synthesis of communicating automata from HMSCs. In *FMICS'00*, 203-224, 2000.
9. J.G. Henriksen, M. Mukund, K.N. Kumar, M.A. Sohoni, and P.S. Thiagarajan. A theory of regular MSC languages. *Inf. Comput.* **202**(1): 1-38, 2005.
10. J. Klein, B. Caillaud, and L. Hélouët. Merging scenarios. In *FMICS'04*, 209-226, 2004.
11. M. Lohrey and A. Muscholl. Bounded MSC communication. *Inf. Comput.* **189**(2): 160-181, 2004.
12. P. Madhusudan and B. Meenakshi. Beyond Message Sequence Graphs. In *FSTTCS'01*, LNCS **2245**, 256-267, 2001.
13. R. Morin. Recognizable Sets of Message Sequence Charts. In *STACS'02*, LNCS **2285**, 523-534, 2002.
14. M. Mukund, K.N. Kumar, and M.A. Sohoni. Bounded time-stamping in message-passing systems. *TCS* **290**(1): 221-239, 2003.
15. A. Muscholl, D. Peled, and Z. Su. Deciding properties of Message Sequence Charts. In *FOSSACS'98*, LNCS **1378**, 226-242, 1998.
16. A. Muscholl and D. Peled. Message Sequence Graphs and Decision Problems on Mazurkiewicz Traces. In *MFCS'99*, LNCS **1672**, 81-91, 1999.