Products of Message Sequence Charts

Philippe Darondeau, Blaise Genest, Loïc Hélouët

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Products of Message Sequence Charts

Philippe Darondeau , Blaise Genest , Loïc Hélouët[†]

N°6258

Septembre 2007

Systèmes communicants

Rapport de recherche

ISSN 0249-6399

Products of Message Sequence Charts *

Philippe Darondeau[†] , Blaise Genest[‡] , Loïc Hélouët[†]

Systèmes communicants
Projet S4 and DistribCom

Rapport de recherche n ° 6258 — Septembre 2007 — 27 pages

Abstract: An effective way to assemble partial views of a distributed system is to compute their product. Given two languages of message sequence charts generated by message sequence graphs, we address the problem of computing a message sequence graph that generates their product. Since all MSCs generated by a message sequence graph may be run within fixed bounds on the message channels, a subproblem is to decide whether the considered product is existentially bounded. We show that this question is undecidable but turns decidable in the restricted case where all shared events belong to the same process. For this case, we propose sufficient conditions under which a message sequence graph representing the product can be constructed.

Key-words: Scenarios, product, partial orders, composition, distributed systems

(Résumé : tsvp)

* Work supported by France Telecom R&D (CRE CO2)
[†] INRIA
[‡] CNRS

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Produits de Message Sequence Charts

Résumé : Les produits sont un moyen effectif de calculer des vues partuielles de comportements de systèmes distribués. Etant donnés deux langages de Message Sequence Charts engendrés par des MSC graphes, nous étudions la question du calcul d'un MSC graphe engendrant leur produit. Puisque tous les MSCs engendrés par un MSC graphe peuvent être exécutés sans dépasser une certaine borne sur le contenu des canaux de communication (on parle de borne existentielle) un sous-problème est de décider si le produit considéré est existentiellement borné. Nous montrons que cette question est indécidable dans le cas général, mais devient décidable dès que l'on restreint le produit à des MSC graphes dont les événements communs sont localisés sur un unique processus. Pour ce cas, nous proposons des conditions suffisantes pour que le MSC graphe engendrant le produit de deux langages de MSC graphes puisse être construit.

Mots-clé : Scénarios, produit, ordres partiels, composition, systèmes distribués

Contents

1 Introduction

Scenario languages, and in particular Message Sequence Charts (MSCs) have met a considerable interest over the last decade. MSCs allow for the description of distributed systems executions, and their visual aspect made them popular in the engineering community. Several difficult problems such as races, confluence, model checking have been addressed for MSCs and their variants. For a recent survey of Message Sequence Charts (MSCs) and Message Sequence Graphs (MSC-graphs), we refer the reader to [4, 7]. We feel that two problems about MSCs and MSC-graphs need to be addressed before design engineers can envisage serious applications. One question is how to implement MSC-languages given by MSC-graphs. Apart from the restricted case of Local Choice (C)MSC-graphs [6, 5], this problem has received no satisfactory solution, since either deadlocks arise from the implementation [11, 3], or implementation may exhibit unspecified behaviors [1]. An even more basic issue is how to model a system using an MSC-graph. For instance, how to describe the common behavior of several MSC-graph modules reflecting different aspects of a distributed system. A first attempt in this direction [8] defines the amalgamation of MSC graphs as a lifted version of the merge of their basic MSC blocks (or bMSCs).

We propose here a different, more flexible operation of product of MSC-languages, assuming they do not share any message (which is allowed in the amalgamation of MSCs considered in [8]), but they can share internal events. The product we choose is the 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. The objective of the work started here is to be able to compute the aggregation of MSC-graph descriptions of subsystems, thus yielding support to the modular design of distributed systems.

An important feature of MSC-graphs (or more generally safe CMSC-graphs) is existential boundedness [9], that is there exists an upper bound on the contents of the message channels, within which all MSCs in the language can be run. In order to have an MSC-graph representation, the product must be existentially bounded. This is the main problem solved in this paper. We show that knowing whether the product of two MSC-graphs is existentially bounded is undecidable. However, if the shared events belong to only one process, then this question becomes decidable. Once a product is known to be existentially bounded, results on representative linearizations of [3] can be used. In brief, languages of MSCs defined by the globally cooperative subclass of safe CMSC-graphs have regular sets of linear representatives, where the regular representations can be computed from the CMSC-graphs and conversely. 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 [3] assume FIFO communication between processes and they ignore the contents of messages in the definition of MSCs. The FIFO assumption would force a complicated definition of the product of MSCs (because mixing two different flows of messages between two processes needs not preserve the FIFO condition). Therefore, we impose here the weaker requirement that messages with identical contents do not overtake one another -as was assumed in [1]. We adapt to this different framework the correspondence established in [3].

The note is organized as follows. Section 2 recalls the background of MSCs and MSC-graphs. Section 3 introduces the mixed 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 mixed product of two existentially bounded MSC-languages, whereas a decision is possible whenever the shared events belong to a single process. Section 7 defines for that special case an operation of mixed product on CMSC-graphs.

2 Background

To begin with, we recall here the usual definition of *compositional Message Sequence Charts* (*CMSCs* for short), which describe executions of communicating protocols, and of CMSC-graphs, which are generators of CMSCs 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.,* $p\,?\,q\,(m) \in \mathcal{S}_p$).

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 labelling 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,
– *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* $\lambda^{-1}(\mathcal{R})$.

Definition 1 extends the original definition of [3] mainly in that the communications between an ordered pair of processes are not necessarily FIFO, but weakly FIFO. That is, there are as many FIFO channels from p to q as there are types of events $p\,!\,q\,(m)$. The choice of weak FIFO over FIFO is for ease of use mainly, since the mixed product could mess the FIFO restriction. However, we do believe that results similar to those presented here would apply with the usual FIFO restriction, albeit with more technicalities.

Given a MSC $X = (E, \lambda, \mu, (<_p)_{p\in\mathcal{P}})$, 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 (linear) *representation* 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 linear representations of X. For a set $\mathcal{X}$ of MSCs, let $\mathcal{L}in(\mathcal{X})$ denote the union of $\mathcal{L}in(X)$ for all $X \in \mathcal{X}$. Linearizations can be defined irrespective of MSCs 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* $v = w$.

Any linear extension w of an MSC belongs to $\mathcal{L}in$. Conversely, a word $w = \epsilon_1 \dots \epsilon_n \in \mathcal{L}in$ is a linear extension of $Msc(w) = (\{1, \dots, n\}, \lambda, \mu, (<_p))$ with:
– $\lambda(i) = \epsilon_i$ and $i <_p j$ if $i < j$ & $\epsilon_i, \epsilon_j \in \mathcal{E}_p$,
– $\mu(i) = j$ if the letter $\epsilon_i = p\,!\,q\,(m)$ occurs n times in $\epsilon_1 \dots \epsilon_i$ and the letter $\epsilon_j = q\,?\,p\,(m)$ occurs n times in $\epsilon_1 \dots \epsilon_j$ for some p, q, m, n.

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* $\mathcal{X}$ *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 send event is matched by μ either with the first or with the second receive event.

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

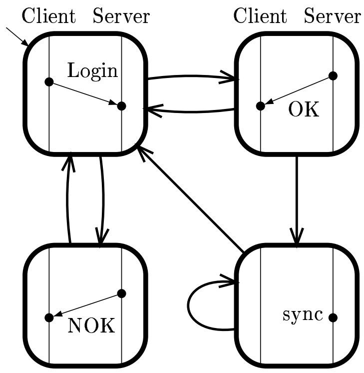

Figure 1: Identification Scenario G_1.

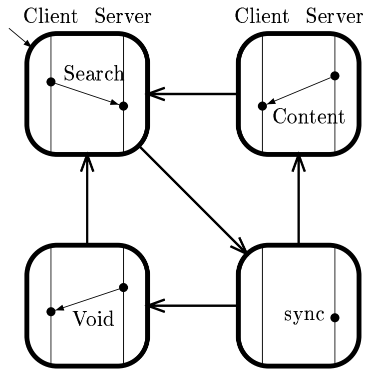

Figure 2: Searching Scenario G_2.

– $\mu \cap (E^i \times E^i) = \mu^i$ 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 graph, $V^0, V^f \subseteq V$ are the subsets of initial or 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 composition 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 labelled 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 $\Lambda(v_0) \cdot \Lambda(v_f)$. Figures 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 Mixed 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 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 is needed: we propose here to synchronize shared events and shuffle non-shared events per process. All shared events are internal events (messages are never shared). The intersection with a regular language could be used in place of the shared events to control the shuffle, but this would not change significantly the results of this paper, except for heavier proofs.

First, we recall the definition of the *mixed product* $L_1 \parallel L_2$ of two languages L_1, L_2 of words (see [2]), 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., $\{ab\} \parallel \{cad\} = \{cabd, cadb\}$.

Definition 5 *For $i = \{1,2\}$, let $\mathcal{X}_i$ be a MSC-language over $\mathcal{E}_i$, 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 $Msc(\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2))$ and it is a MSC-language over $\mathcal{E}_1 \cup \mathcal{E}_2$.*

The mixed product operation may serve 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 Figures 1,2. The synchronization on the shared events *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. Note that even though X_1 and X_2 are MSCs, $\{X_1\} \parallel \{X_2\}$ may contain more than one MSC. Also note that $w_1 \parallel w_2 \subseteq \mathcal{L}in$ for $w_1 \in \mathcal{L}in$ and $w_2 \in \mathcal{L}in$. Mixing all linearizations pairwise yields all linearizations of a product:

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

Proof We know that for any MSC X, $\mathcal{L}in(\mathcal{X})$ is an equivalence class, hence $\mathcal{L}in(\mathcal{X}_1 \parallel \mathcal{X}_2) = [\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2)]$ is immediate.

$\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2) \subseteq \mathcal{L}in(\mathcal{X}_1 \parallel \mathcal{X}_2)$ also follows from the properties of $\mathcal{L}in(\mathcal{X})$, because $\mathcal{X}_1 \parallel \mathcal{X}_2 = Msc(\mathcal{L}in(\mathcal{X}_1) \parallel \mathcal{L}in(\mathcal{X}_2))$. 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 $\mathcal{L}in(\mathcal{X})$, 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 $\mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2)$ *is closed under the equivalence $\equiv$ (see Def. 3).*

Proof Let $w \in \mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2)$. 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 $E^j = \{i \mid 1 \leq i \leq n \wedge \epsilon_i \in \mathcal{E}^j\}$ and $E'^j = \{i \mid 1 \leq i \leq n \wedge \epsilon'_i \in \mathcal{E}^j\}$, 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 $w' \in \mathcal{L}in(X_1) \parallel \mathcal{L}in(X_2)$. $\square$

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 $X_1 = Msc(w_1) = Msc(w_1')$, $X_2 = Msc(w_2) = Msc(w_2')$, $X_3 = Msc(w_3)$. 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 [10], it follows from a pumping lemma that whenever $\mathcal{L} \subseteq \mathcal{L}in$ is a regular representative set for $\mathcal{X}$, the words in $\mathcal{L}$ are uniformly B-bounded, for some $B > 0$, according to the following definition. A word $w \in \mathcal{E}^*$ is *B-bounded* if, for any prefix v of w, for any $m \in \mathcal{M}$ and $p, q \in \mathcal{P}$, the number of occurrences of $p\,!\,q\,(m)$ in v exceeds that of $q\,?\,p\,(m)$ by at most B.

A MSC X is *$\forall$-B-bounded* if every linear representation $w \in \mathcal{L}in(X)$ is B-bounded. A MSC X is *$\exists$-B-bounded* if some linear representation $w \in \mathcal{L}in(X)$ 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 be $\forall$-B-bounded for no 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 not necessarily a regular language.

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

Proof Proposition 2 is an immediate corollary of proposition 1.

The above result shows that 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 proposition 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 let for all j 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. Both MSCs are $\exists$-1-bounded since $(p!q(m_1)q?p(m_1))^j$ 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 is not $\exists$-$(B-1)$-bounded.

Definition 6 *Given a 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* p *and* q *in* $\mathcal{P}$ *and* $m \in \mathcal{M}$*,* e *is the* i*-th event on process* p *with the label* $\lambda(e) = p\,?\,q\,(m)$ *and* e' *is the* $i+B$*-th event on process* q *with the label* $\lambda(e') = q\,!\,p\,(m)$*. We also define* $Rev_{\geq B} = \cup_{B'\geq B} Rev_{B'}$*.*

Proposition 3 (lemma 2 in [9]) *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

An important question regarding MSC-languages and their products is verification. Most often, in decidable cases [5, 13], verifications performed on a MSC-language $\mathcal{X}$ amount to check the membership of a given MSC X, or to check that $\mathcal{L}in(\mathcal{X})$ 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 check regular properties of $\mathcal{L}(G_1) \parallel \mathcal{L}(G_2)$, one often needs computing a safe CMSC-graph G such that $\mathcal{L}(G) = \mathcal{L}(G_1) \parallel \mathcal{L}(G_2)$, and in particular 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 MSC-graphs. It is undecidable whether* $\mathcal{L}(\mathcal{G}_1) \parallel \mathcal{L}(G_2)$ *is existentially bounded.*

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_j. 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 $G_1 = (V, \rightarrow, \Lambda_1, V^0, V^f)$ and

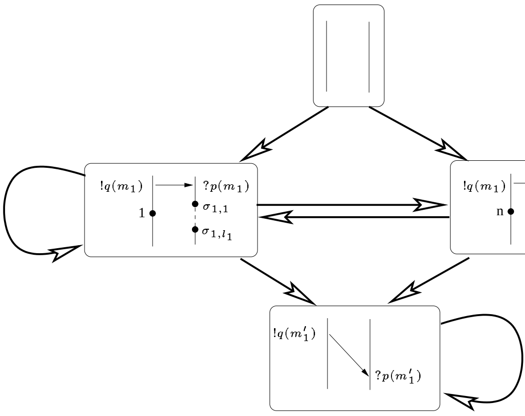

Figure 3:

$G_2 = (V, \rightarrow, \Lambda_2, V^0, V^f)$, with the same underlying graph $(V, \rightarrow, V^0, V^f)$, constructed as follows (G_1 is partially shown in Fig. 3).

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

For each $v \in V$, $\Lambda_1(v)$ is a finite MSC over $\mathcal{P}_1 = \{p, q\}$, $\mathcal{A}_1 = \{1, \dots, n\} \cup \Sigma$, $\mathcal{M}_1 = \{m_1, m'_1\}$. Actions $i \in \{1, \dots, n\}$ represent indices of pairs of words (u_i, v_i) and they occur on process p. Actions $\sigma \in \Sigma$ represent letters of words u_i and they occur on process q. Let $\Lambda_1(v_0)$ be the empty MSC. For $i \in \{1, \dots, n\}$, 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 $q(\sigma_{i,1})\, q(\sigma_{i,2}) \dots q(\sigma_{i,l_i})$, representing $u_i = \sigma_{i,1}\, \sigma_{i,2} \dots \sigma_{i,l_i}$, on process q. Finally let $\Lambda_1(v_{n+1})$ 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 $\mathcal{P}_2 = \{p, q\}$, $\mathcal{A}_2 = \{1, \dots, n\} \cup \Sigma$, $\mathcal{M}_2 = \{m_2, m'_2\}$. For $i = 0, \dots, n$, $\Lambda_2(v_i)$ is defined alike $\Lambda_1(v_i)$ but now replacing m_1 with 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. $\Box$

The proof of Theorem 1 is similar to the proof that $\mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset$ is undecidable for generic MSC-graphs G_1, G_2 [12]. Theorem 1 motivates the introduction of a *monitor* process mp and a *monitored product* in which all shared events are internal events located on the monitor process. The monitored product $\mathcal{X}_1 \,|||\, \mathcal{X}_2$ with monitor process mp is defined as the mixed product $\mathcal{X}_1 \parallel \mathcal{X}_2$ after all internal events $p(a)$ ocurring in both $\mathcal{X}_1$ and $\mathcal{X}_2$ with $p \neq mp$ have been renamed, such that the set of the shared events is included in

$\mathcal{SE} = \{mp(a) \mid a \in \mathcal{A}_1 \cap \mathcal{A}_2\}$. For instance, with the CMSC-graphs of Fig. 1 and Fig. 2, we can chose $mp = server$ and $\mathcal{SE} = \{mp(sync)\}$ in $\mathcal{L}(G_1) \,|||\, \mathcal{L}(G_2)$. The adequacy of the monitored product to weave aspects of a distributed system is confirmed by the following theorem.

Theorem 2 *Given two safe CMSC-Graphs G_1, G_2, one can decide in co-NP whether the monitored product of $\mathcal{L}(G_1) \,|||\, \mathcal{L}(G_2)$ is $\exists$-bounded.*

The next section is a proof sketch for this theorem. The proof would be easier if we had defined $\mathcal{X}_1 \,|||\, \mathcal{X}_2$ by renaming all processes p occurring in both $\mathcal{X}_1$ and $\mathcal{X}_2$ with $p \neq mp$, i.e. if the final set of shared processes is $\{mp\}$. Then, $\mathcal{L}(G_1) \,|||\, \mathcal{L}(G_2)$ would always be existentially bounded (with the bound given by the maximum of the minimal existential bounds of $\mathcal{L}(G_1)$ and $\mathcal{L}(G_2)$). However, this would cause a serious loss of expressivity. E.g., for the CMSC-graphs of figures 1 and 2, $\mathcal{L}(G_1) \,|||\, \mathcal{L}(G_2)$ would no longer mean that search requests are not answered unless the client is logged in.

6 Checking Existential Boundedness

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

Proposition 4 *Given two safe CMSC-graphs G_1 and G_2, the MSC-language $\mathcal{L}(G_1)|||\mathcal{L}(G_2)$ is existentially bounded if and only if it is existentially B-bounded for $B = (2\,|\mathcal{P}| + 2)^2 \times (|G_1| + 1) \times (|G_2| + 1) \times K$, where $|G_i|$ is the number of events in G_i and K is the square of the sum $2\,|\mathcal{P}| + (|\mathcal{P}|)^2/2 \times (\,|\mathcal{M}_1| \times |G_1| + |\mathcal{M}_2| \times |G_2|)$.*

Then we show that we can check whether the monitored product of $\mathcal{L}(G_1), \mathcal{L}(G_2)$ is $\exists$-B-bounded, using the bound of proposition 4.

Proposition 5 *Given two safe CMSGs G_1, G_2 and an integer B, one can decide in co-NP whether $\mathcal{L}(G_1) \,|||\, \mathcal{L}(G_2)$ is $\exists$-B-bounded.*

These two results are obtained using special representations for MSCs constructed by monitored product.

★ *Graph representation of monitored products*
Let $X \in X_1 \,|||\, X_2$ then $\exists w$: $X = Msc(w)$ and $\pi_i(w) = w_i \in \mathcal{L}in(X_i)$. The MSC X is determined up to isomorphism by its projection on processes. More precisely, for each $p \in \mathcal{P}$, $\pi_p(w) \in \pi_p(w_1) \,|||\, \pi_p(w_2)$. Moreover, for $p = pm$, $\pi_p(w_1)$ 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 $(E^1_p, <^1_p)$ and $(E^2_p, <^2_p)$ where the synchronized pairs of events $e_1 \in E^1_{pm}$ and $e_2 \in E^2_{pm}$ with labels in $\mathcal{SE}$ are coalesced. Let $\longleftrightarrow \subseteq E^1_{pm} \times E^2_{pm}$ be the relation comprising synchronized pairs of events. For each $p \in \mathcal{P}$, including $p = pm$, let

$\to_p^1 \subseteq E_p^2 \times E_p^1$ (resp. $\to_p^2 \subseteq E_p^1 \times E_p^2$) be the relation comprising ordered pairs of events $e_2\, e_1$ (resp. $e_1\, e_2$) switching from E_p^2 to E_p^1 (resp. from E_p^1 to E_p^2) in the interleaved sequence $(E_p, <_p)$.

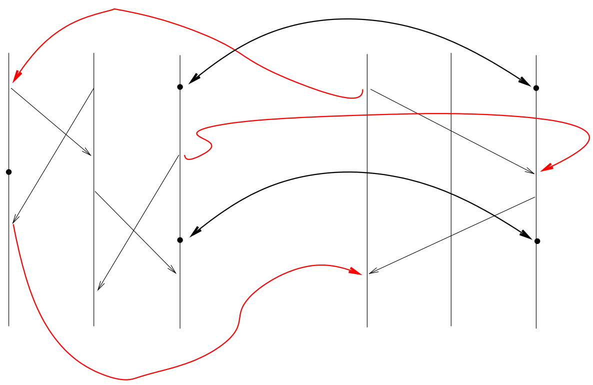

Figure 4: MSC X with two components X_1 and X_2

The MSC X may now be represented by the juxtaposition of X_1 and X_2 interlinked with $\longleftrightarrow$ and with the relations $\to_p^1$ and $\to_p^2$ for all $p \in \mathcal{P}$. The result is a graph $X_{1|||2}$ with set of nodes $E^1 \cup E^2$. For an illustration, see Fig. 4 where the edges of the graph represent the relations μ_i, $\longleftrightarrow$ and the transitive reduction of the relations $<_p^i$ and $\to_p^i$. As X is an MSC, $X_{1|||2}$ should be an acyclic graph except for short circuits $e \longleftrightarrow e'$. The MSC X may be reconstructed from $X_{1|||2}$ as follows. Let E be the quotient of $E^1 \cup E^2$ by the equivalence relation $\longleftrightarrow$. Then $<_p$ is the transitive closure of the union of $<_p^1$, $<_p^2$ and $\to_p^1$, $\to_p^2$.

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

Let $X \in \mathcal{L}(G_1)|||\mathcal{L}(G_2)$, thus X may be represented in product form by $X_{1|||2} = (X_1, X_2, \longleftrightarrow, (\to_p^i)_{\,p\in\mathcal{P}}^{\,i=1,2})$. Suppose that X is not $\exists B$-bounded for the specified bound B. By Prop. 3, $< \cup\, Rev_{\geq B}$ has a cycle in X. As X_1 and X_2 do not share messages, $Rev_{\geq B}$ is the union of the similar relations $Rev^i_{\geq B}$ in X_i for $i = 1, 2$. Therefore, the union of $\longleftrightarrow$ and the relations $<^i$, $Rev^i_{\geq B}$, and $\to_p^i$ for $i = 1, 2$ has a cycle $e_1\, e_2 \dots e_m$ (letting $e_j \neq e_k$ for $j \neq k$ and $e_{m+1} = e_1$). One can assume w.l.o.g. that $e_1\, e_2 \dots e_m$ contains no events with shared label and at most two events on each process p, hence $m \leq 2\,|\mathcal{P}|$. As X is an MSC, the union of relations $<^i$, and $\to_p^i$ for $i = 1, 2$ is free from cycles. Therefore, there is at least one pair of events (e_j, e_{j+1}) in $Rev^i_{\geq B}$ for $i = 1$ or $i = 2$.

Suppose w.l.o.g. that $e_1\, Rev^1_{B_1}\, e_2$ for some bound $B_1 \geq B$. We will construct MSCs $X'_1 \in \mathcal{L}(G_1), X'_2 \in \mathcal{L}(G_2)$ and embeddings $\phi : X_1 \hookrightarrow X'_1, \phi : X_2 \hookrightarrow X'_2$ such that $\phi(e_1)\,\phi(e_2)\dots\phi(e_m)$ is a cycle in $\phi(Rev_{\geq B}) \cup X'$, with $(X', <_{X'})$ the oriented graph obtained by connecting X'_1 and X'_2 with $\longleftrightarrow$ and with edges $(\phi(e_j), \phi(e_{j+1}))$ for all $j \leq m$ such that $e_j \to_p^i e_{j+1}$ for some $i \in \{1, 2\}$ and $p \in \mathcal{P}$. More precisely, we will construct embeddings such that $\phi(e_1)\, Rev_{\geq B_1+1}\, \phi(e_2)$ and $e_j\, Rev_{B_j}\, e_{j+1} \Rightarrow \phi(e_j)\, Rev_{\geq B_j}\, \phi(e_{j+1})$ for pairs (e_j, e_{j+1}) in Rev_{B_j} with $j \neq 1$. Proposition 4 then follows by induction, after proving that $X'_{1|||2}$ is acyclic (hence there exists at least an MSC whose order refines X').

★ *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 $B_1 \geq B$, where $e_1\, e_2 \ldots e_m$ is the considered cycle in the union of the relations $<^i$, $Rev^i_{\geq B}$, $\rightarrow^i_p$ $(i = 1, 2)$, with $m \leq (2|\mathcal{P}|)$. Thus $\lambda^1(e_1) = q\,?\,p(m)$ and $\lambda^1(e_2) = p\,!\,q(m)$ for some $p, q \in \mathcal{P}$ and $m \in \mathcal{M}_1$. As $B_1 \geq B$, at least $B - 1$ emission events $p\,!\,q(m)$ preceding e_2 in X_1 are matched by μ^1 with reception events $q\,?\,p(m)$ following e_1 in X_1. At least $(B-1)/\,|G_1|$ of these emission events originate from the same vertex v of G_1. As $(B-1)/\,|G_1| > (2\,|\mathcal{P}|+1)^2 \times (|G_2|+1) \times K$, 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, which paths give a set of regular representatives bounded by some $B \leq |G|/2$. This also apply 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 pm, 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 $i > 1, v_i = last(V'_i)$ if $V'_i \neq \epsilon$, else $v_i = v_{i-1}$ ($V'_i = \epsilon$ acts as an (empty) self loop on v_{i-1}). As $k > (2\,|\mathcal{P}|+1)\,|G_2| \times K$, 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 $\alpha^1\beta^1_1 \ldots \beta^1_{h-1}\beta^1_h\beta^1_h\beta^1_{h+1} \ldots \beta^1_K\gamma^1$, 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 $\alpha^1\beta^1_1 \ldots \beta^1_{h-1}\beta^1_h$, 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

$p!q(m)$ and $p?q(m)$ in the loop β_h^1 of the safe CMSC-graph G_1, we have $\phi(e_1)\, Rev_{>B_1}\, \phi(e_2)$. Let $e_j\, Rev_{B_j}\, e_{j+1}$: The remark on the existential bound for safe CMSC graphs also applies here, 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 $\phi(e_j)\, Rev_{\geq B_j}\, \phi(e_{j+1})$ 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 send of e_j and r is the matching receive 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 $\alpha^2\beta_1^2 \dots \beta_{h-1}^2\beta_h^2\beta_h^2\beta_{h+1}^2 \dots \beta_K^2\gamma^2$.

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 $e_j \rightarrow_p^i e_{j+1}$ for some $i \in \{1, 2\}$ and $p \in \mathcal{P}$. By construction, $\phi(e_1)\phi(e_2)\dots\phi(e_m)$ is a cycle of $<_{X'} \cup Rev_{\geq B}$, with $\phi(e_1)\, Rev_{\geq B_1+1}\, \phi(e_2)$ and $e_j\, Rev_{B_j}\, e_{j+1} \Rightarrow \phi(e_j)\, Rev_{\geq B_j}\, \phi(e_{j+1})$ for pairs (e_j, e_{j+1}) in Rev_{B_j} with $j \neq 1$. In order to validate the inductive proof of Proposition 4, it remains to show that one can choose $h \in \{1, \dots, K\}$ such that the associated graph X' is acyclic (up to short circuits $e \longleftrightarrow e'$).

The graph formed by connecting X_1' and X_2' with $\longleftrightarrow$ is acyclic because $\longleftrightarrow$ concerns only one process pm. Hence without lack of generality, we can write any cycle in X' in the form $\phi(e_1')\phi(e_2')\dots\phi(e_{2l}')$ where $(\forall j)$ (e_{2j}', e_{2j+1}') is an edge of the cycle $e_1 \dots e_m$ 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, $\{e_1', \cdots, e_{2l}'\} \subseteq \{e_1 \cdots e_m\}$, and hence any e_i' belongs either to $\alpha^1, \alpha^2, \gamma^1$ or γ^2.

We claim that one can choose $h \in \{1, \dots, K\}$ such that for all events e, e' in the cycle $e_1 \dots e_m$, if $(\phi(e), \phi(e'))$ is in the reflexive and transitive closure of the union of $<_{X_1'}$, $<_{X_2'}$ and $\longleftrightarrow$, then (e, e') was is in the reflexive and transitive closure of the union of $<_{X_1}$, $<_{X_2}$ and $\longleftrightarrow$. The last part of the section is devoted to establishing this claim. Then we reach a contradiction since we obtain a cycle $e_1'e_2' \dots e_{2l}'$ in $X_{1|||2}$ which is an MSC, hence acyclic. Therefore, for the considered h, $(X', <_{X'})$ is an acyclic graph.

★ *A technical lemma*

Let $X_0^1 \cdots X_{K+1}^1$ and $X_0^2 \cdots X_{K+1}^2$ be the respective decompositions of X_1 and X_2 into MSC factors induced from the generating paths in G_1 and G_2:
$\alpha^1\beta_1^1 \dots \beta_{h-1}^1\beta_h^1\beta_{h+1}^1 \dots \beta_K^1\gamma^1$ and $\alpha^2\beta_1^2 \dots \beta_{h-1}^2\beta_h^2\beta_{h+1}^2 \dots \beta_K^2\gamma^2$. We consider here the graph formed by connecting X_1 and X_2 with $\longleftrightarrow$. This relation, which connects synchronized events on the monitor process, projects (by construction) on connected pairs of MSC factors (X_j^1, X_j^2), and it may be reconstructed from these projections. We are interested in the order relation between some events from α_1, α_2 and γ_1, γ_2, that is from (X_0^1, X_0^2) and (X_{K+1}^1, X_{K+1}^2) in the two component MSC $(X_0^1, X_0^2)\cdot(X_1^1, X_1^2)\cdot\ldots\cdot(X_K^1, X_K^2)\cdot(X_{K+1}^1, X_{K+1}^2)$.

As G_1 is safe and atomic, and all factors β_j^1 are loops in G_1, for all $p, q \in \mathcal{P}$ and $m \in {}_mathcalM_1$, the number of emission events $p\,!\,q\,(m)$ in excess over reception events $q\,?\,p\,(m)$ in $X_0^1 \cdot X_1^1 \dots X_j^1$ does not depend on j for $0 \leq j \leq K$. 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 $X^1_j \cdot X^1_{j+1} \dots X^1_{K+1}$ does not depend on j for $1 \leq j \leq K+1$, and it is equal to $N(p,q,m)$, since X_1 is an MSC. Moreover $N(p,q,m) \leq (|G_1|/2)$. Similar remarks apply to G_2.

For all $0 < j \leq n \leq K$, the dependence relation $\mathcal{O}(j,n)$ between the events from $(X^1_0, X^2_0) \cdots (X^1_{j-1}, X^2_{j-1})$ and from $(X^{n+1}_0, X^2_{n+1}) \cdots (X^1_{K+1}, X^2_{K+1})$ induced from $(X^1_j, X^2_j) \cdots (X^1_n, X^2_n)$ may therefore be represented as a relation on generic sets of events γ and γ' as follows. Let Γ 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 set γ represents an ordered relation of events, where (p,i) stands for the last event on process p in the i-th component, and the events $(p,q,m,1) < \cdots < (p,q,m,N(p,q,m))$ stand for the $N(p,q,m)$ last sends $p!q(m)$. We 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 Γ' be the set of events $(p,i)'$ and $(p,q,m,l)'$, where $(p,i)'$ stands for the first event on process p, and the events $(p,q,m,1)' < \cdots < (p,q,m,N(p,q,m))'$ stand for the $N(p,q,m)$ first receives $q?p(m)$. 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 $0 < j \leq n \leq K$, let $\mathcal{O}(j,n)$ be the binary relation on $\Gamma \times \Gamma'$ such that $(\gamma,\gamma') \in \mathcal{O}(j,n)$ if the event γ from Γ is smaller than the event γ' from Γ' in $\Gamma(X^1_j, X^2_j) \cdot \ldots \cdot (X^1_n, X^2_n)\Gamma'$. In order to complete the proof of Prop. 4, it suffices now to show that:

- $\mathcal{O}(1,h-1) = \mathcal{O}(1,h)$ for some $h < K$,
- $\mathcal{O}(j,n) = \mathcal{O}(j,l) \circ \mathcal{O}(l+1,n)$ 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 $Id' : \Gamma' \to \Gamma : Id'(e') = e$.

These relations entail $\mathcal{O}(1,K) = \mathcal{O}(1,h-1) \circ \mathcal{O}(h,K) = \mathcal{O}(1,h) \circ \mathcal{O}(h,K)$. Therefore, for any pair of two component MSCs Y, Z, the order between the events from Y and Z is the same in $Y \cdot (X^1_1, X^2_1) \cdots (X^1_K, X^2_K) \cdot Z$ as in $Y \cdot (X^1_1, X^2_1) \cdots (X^1_{h-1}, X^2_{h-1}) \cdot (X^1_h, X^2_h) \cdot (X^1_h, X^2_h) \cdot (X^1_{h+1}, X^2_{h+1}) \cdots (X^1_K, X^2_K) \cdot Z$. 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 $(X^1_j, X^2_j) \dots (X^1_l, X^2_l)$ is linked with a matching $q\,?\,p(m)$ received but not emitted in $(X^1_{l+1}, X^2_{l+1}) \dots (X^1_n, X^2_n)$ 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 $\Gamma \times \Gamma'$ containing Id' is strictly bounded by $K = |\Gamma||\Gamma'| \leq [2\,|\mathcal{P}| + (|\mathcal{P}|)^2/2 \times (|M_1| \times |G_1| + |M_2| \times |G_2|)]^2$. Therefore, $\mathcal{O}(1,h-1) = \mathcal{O}(1,h)$ for some $h \leq K$.

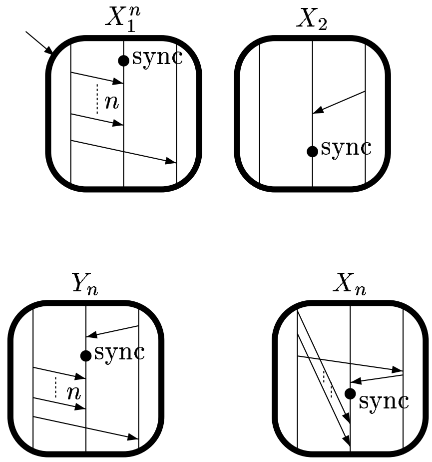

Figure 5: Two MSCs Y^n, X^n in the monitored product of X_1^n and X_2

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

In order to conclude that $\mathcal{L}(G_1) \,|||\, \mathcal{L}(G_2)$ 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 $X \in (X_1 \,|||\, X_2)$ 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 $Msc(\mathcal{L}in^{f(B)}(X_1 \,|||\, X_2))$. An illustration is given in Fig. 5: for all n, X_1^n and X_2 are $\exists 1$-bounded, $X^n \in (X_1^n \,|||\, X_2)$ is not $\exists 1$-bounded, and $Y^n \in Msc(\mathcal{L}in^{n-1}(X_1^n \,|||\, X_2))$ is $\exists 1$-bounded. Linear representations of products of MSCs are therefore of little help: $X_1 \,|||\, X_2$ 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 linear representations.

The constructive proof which we propose for Prop. 5 stems from the following lemma.

Lemma 2 *Let G_1 and G_2 be safe and atomic CMSGs and B an integer. Then $\mathcal{L}(G_1 \,|||\, G_2)$ is $\exists B$-bounded if and only if, for any synchronized pair of MSCs $X_1 \in \mathcal{L}(G_1)$ and $X_2 \in \mathcal{L}(G_2)$ with respective sets of events E^1 and E^2, there is no subset $\{e_1, \dots, e_n\} \subseteq E^1 \cup E^2$ with at most two events in $E^1_p \cup E^2_p$ 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^i_B, or $E^i_p \times E^{3-i}_p$ for $i = 1$ or 2 and $p \in \mathcal{P}$,*
2. *there is a proper cycle in $\{e_1, \dots, e_n\}$ w.r.t. the transitive closure of the relation $<^1 \cup <^2 \cup \longleftrightarrow \cup \dashrightarrow$ where $\longleftrightarrow$ is the synchronizing relation among coalesced events, and $e \dashrightarrow e'$ if $e = e_j \in E^i_p$ and $e' = e_{(j+1) \bmod n} \in E^{3-i}_p$ for some $j \in \{1, \dots n\}$, $i \in \{1, 2\}$ and $p \in \mathcal{P}$.*

Proof immediate from Prop. 3 using the representation of monitor products proposed in section 5. □

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 shall 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_B on the current set E. By lemma 2, $\mathcal{L}(G_1) \mathbin{|||} \mathcal{L}(G_2)$ is not $\exists B$-bounded if and only if, at some state, a cycle is found in $(E, < \cup\, Rev_B)$. Prop. 5 follows therefore from the finiteness of the construction.

Each state s of the non-deterministic machine 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_B.

The states of the machine should provide enough information to update $(E, <, Rev_B)$ when constructing new states from existing states. Assume that a pair of vertices (v_1, v_2) has been reached by synchronized paths ρ_1, ρ_2 and the current state is $(v_1, v_2, E, <, Rev_B, \dots)$ where $E = \{e_1, \dots, e_n\}$. A new state may result from taking an edge $v_1 \rightarrow v_1'$ in G_1, or an edge $v_2 \rightarrow v_2'$ in G_2, or two edges $v_1 \rightarrow v_1'$ and $v_2 \rightarrow v_2'$ 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. Send and receive events generated from G_1 or G_2 may be inserted or not in the set of distinguished events $\{e_1, \dots, e_n\}$. Local events, and in particular synchronized event, will never be inserted in this set: they cannot belong to a cycle $\{e_1, \dots, e_n\}$ since the conditions in lemma 2 forbid to have three events e_i on the same process.

Recall that $<$ is the transitive closure of the union of $<^1$, $<^2$, $\longleftrightarrow$, and $\rightarrow$ where relation $\rightarrow$ 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 $\rightarrow$ is not. Suppose e.g. that an edge $v_1 \rightarrow v_1'$ 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 $\{e_1, \dots, e_n\}$. Possibly $e < e'$ for some e in $\{e_1, \dots, e_n\}$ ($e' <^1 e$ is not possible), because $e = f_0 R_1 f_1 \dots R_k f_k = e'$ for some events $f_1 \dots f_{k-1}$ outside $\{e_1, \dots, e_n\}$ 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, i) 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 $\rightarrow$ 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 receive event.

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 receive events $q\,?\,p\,(m)$. 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, i). At the first time a send event $p\,!\,q\,(m)$ 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 send event. $\mathcal{O}_S(e)(c)$ 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 $\mathcal{O}_P(e)$, and all counters $\mathcal{O}_S(e)(q, m', r)$ with $m' \in \mathcal{M}_i$ are set simultaneously to the current number of messages stored in the channel (q, m', r). Similar counters $\mathcal{O}_S(e)(q', m', r')$ are maintained for all processes q' and messages $m' \in M_j$ ($j = 1$ or 2) such that $(q, j) \in \mathcal{O}_P(e)$.

For case *(i)*, the update is simple: $(pm, 1)$ is inserted into sets $\mathcal{O}_P(e)$ that contained only $(pm, 2)$ 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 $e \rightarrow e'$ or $e' \rightarrow e$ are a priori possible. Updating $<$ with $e \rightarrow e'$ cannot ever introduce circularity in $<$. In contrast, updating $<$ with $e' \rightarrow e$ might result in circularity. *Circularity of relation $<$ must be avoided by explicit checking.*

It remains to consider the updating of relation Rev_B on E. Assume that e and e' are two distinguished events in E and $eRev_Be'$, hence e and e' are labelled 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 $eRev_Be'$, one must anticipate on these two events. The right time for predicting $eRev_Be'$ is when generating the send event f to be matched by the receive event 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 receive event $q\,?\,p\,(m)$, resp. a send event $p\,!\,q\,(m)$ 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 $eRev_Be'$ is recorded in the current state when e and e' have been both inserted into E.

Whenever Rev_B or $<$ is updated, if some cycle appears in $< \cup Rev_B$, one stops the construction of new states with the diagnostic that $\mathcal{L}(G_1) \,|||\, \mathcal{L}(G_2)$ 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 in the rest of the section. 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). 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 CMSG 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 $eRev_Be'$ are expected at both ends of channel c, $rev(e) = c$ means that an event e' such that $eRev_Be'$ 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 $\mathcal{O}_S(e) : Ch \to \{0, 1, \cdots, K\}$,
- $\mathcal{R} : Ch \to \{0, \cdots, K\} \times \{0, \cdots, B\}$, where $\mathcal{R}(c) = (R_r, R_s)$.

The components of the initial state are the vertices v_1^0, v_2^0 and $\mathbf{0}$ 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 CMSG 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 $(pm, 1)$ or $(pm, 2)$ belongs to $\mathcal{O}_P(e)$, $\mathcal{O}_P'(e) = \mathcal{O}_P(e) \cup \{(pm, 1), (pm, 2)\}$.
   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 a send on channel $c = (p, m, q)$*
$count'(c) = count(c) + 1$, $n' = n$ or $n' = n + 1$. Let $\mathcal{R}(c) = (R_r, R_s)$.
If $\mathcal{R}(c) = (0, 0)$ then $\mathcal{R}'(c) = (0, 0)$ or $\mathcal{R}'(c) = (count'(c), B)$ and $rev'(c) = c$.
If $R_s > 0$ then $\mathcal{R}'(c) = (R_r, R_s - 1)$.
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 $rev(e) = c$.
In all cases, $\mathcal{O}'_S(e)(c) = min(\mathcal{O}_S(e)(c), count'(c))$ for every $e \in E$ such that $(head(c), j) \in \mathcal{O}_P(e)$ for $j = 1$ or 2 (with $min(0, a) = a$).
Finally, if $n' = n + 1$ then $P'(n') = i$ and $\mathcal{O}'_P(n') = \{(p, i)\}$.

3. *if v'_i is a receive on channel $c(p, m, q)$*
$count'(c) = count(c) - 1$, $n' = n$ or $n' = n + 1$. Let $\mathcal{R}(c) = (R_r, R_s)$.
If $R_r > 0$ then $\mathcal{R}'(c) = (R_r - 1, R_s)$.
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 $rev(c) = e$.
In all cases, for every $e \in E$ such that $(head(c), j) \in \mathcal{O}_P(e)$ for $j = 1$ or 2, $\mathcal{O}'_S(e)(c) = \mathcal{O}_S(e)(c) - 1$.
For every $e \in E$, if $\mathcal{O}_S(e)(c) = 1$ and $\mathcal{O}'_S(e)(c) = 0$ then
$\mathcal{O}'_P(e) = \mathcal{O}_P(e) \cup \{tail(c)\}$ and $\mathcal{O}'_S(e)(c) = count'(c)$.
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'.

7 CMSC-graph representation of a Monitored Product

In the case where $\mathcal{L}(G_1)|||\mathcal{L}(G_2)$ 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 model-checking and realization. For this purpose, we use the results from [3], where a syntax-semantics correspondence is established between *globally cooperative* CMSC-graphs [5], 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 Figure 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 [5].

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 [3]. However, we deal in this paper with weak FIFO MSCs while [3] considers FIFO MSCs. Instead of proving Theorem 3 from scratch, we derive it from [3], using 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 ([3]) *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 figure 6).*

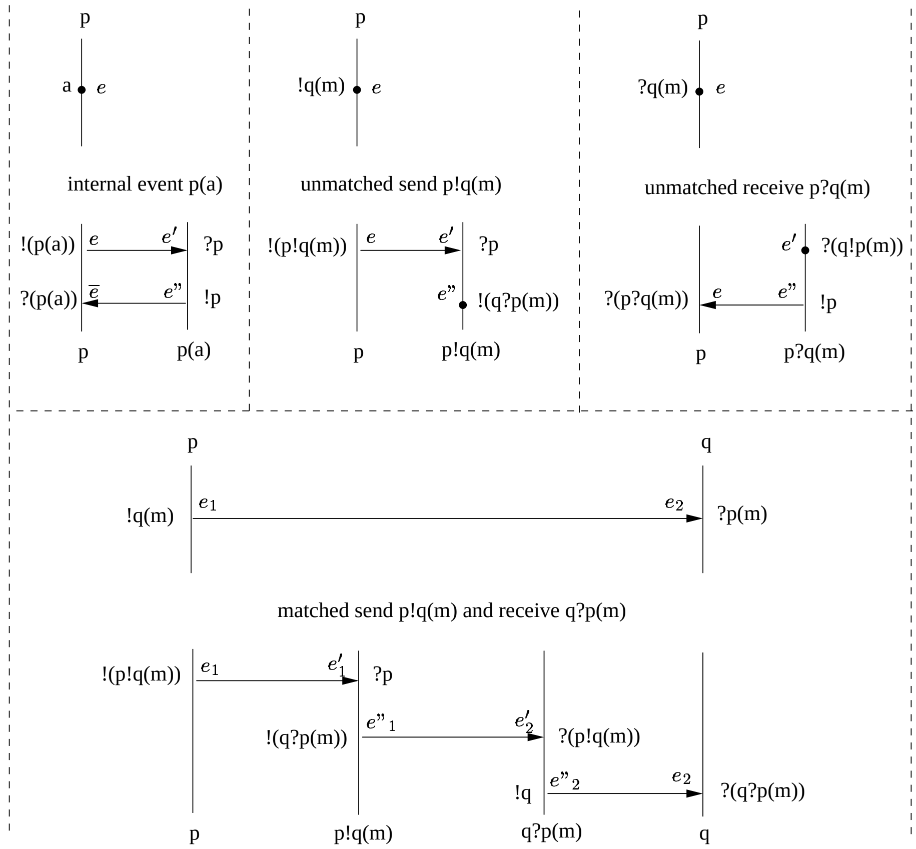

Figure 6: Transforming CMSCs into pure CMSCs

– 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 send event $e \in E$ 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 receive event $e \in E$ 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 $p \in \{\mathcal{P}\}$.

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 a 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 CMSG $G = (V, \rightarrow, \Lambda, V^0, V^f)$, 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 theoremé3.

★ *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 3 *$\mathcal{L}in^B(\mathcal{X}')$ 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 proposition 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 figure 7 (the fat right-to-left arrows are occurrences of Rev'_2).

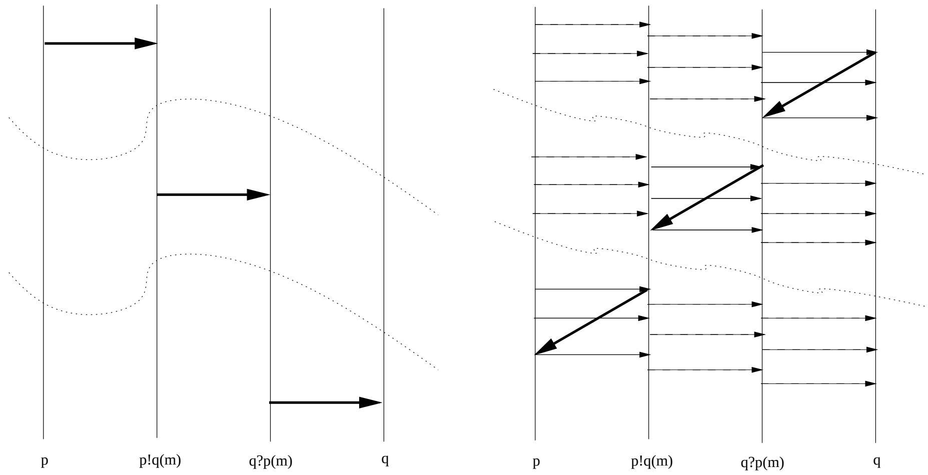

Figure 7:

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 $\mathcal{X}$. $\square$

Lemma 4 *$\mathcal{L}in^B(\mathcal{X}')$ 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 3 and 4, 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 UX' is constructed from X' as follows:
– suppress all processes not in $\mathcal{P}$,
– suppress all events labelled $p\,?\,(p(a))$,
– 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 figure 6.
Note that for any MSC X, $U(X') = X$ up to isomorphism of MSCs, hence $(\cdot)' : X \rightarrow X'$ is injective on MSCs. Therefore, $\mathcal{X} = \mathcal{L}(G)$.

It remains to show that G is globally cooperative. Let $v_0, v_1 \ldots v_n$ 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 UY is a MSC and that UY belongs to $U\Lambda'(v_0) \cdot \ldots \cdot U\Lambda'(v_n)$. Therefore, G is safe. Now let $v_1 \ldots v_n$ 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 definition 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 $\mathcal{L}(G_1) \mid\mid\mid \mathcal{L}(G_2)$ is $\exists$-bounded, then this MSC-language is $\exists$-B-bounded for the bound B defined in Prop. 4. Therefore, $\mathcal{L}in^B(\mathcal{L}(G_1) \mid\mid\mid \mathcal{L}(G_2))$ is a representative set for $\mathcal{L}(G_1) \mid\mid\mid \mathcal{L}(G_2)$. By proposition

2, $\mathcal{L}in^B(\mathcal{L}(G_1) \mid\mid\mid \mathcal{L}(G_2)) = \mathcal{L}in^B(\mathcal{L}(G_1)) \mid\mid\mid \mathcal{L}in^B(\mathcal{L}(G_2))$. 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 $\mathcal{L}(G_1) \mid\mid\mid \mathcal{L}(G_2)$ is $\exists$-bounded. Then one can effectively compute a globally cooperative CMSC-graph G with $\mathcal{L}(G) = \mathcal{L}(G_1) \mid\mid\mid \mathcal{L}(G_2)$. Moreover, G is of size at most exponential in the size of $|G_1|, |G_2|$.*

8 Conclusion

We proposed a general framework to work with the products of views of distributed systems, granted that products are monitored by a single process, and views are given as globally cooperative CMSC-graphs. Namely, one can test whether the monitored product of globally cooperative CMSC-graphs can be represented as another globally cooperative CMSC-graph. In that case, a complete analysis of the system can be performed with existing tools. A direction for future work is to propose guidelines and tools for modeling complex systems with at most one monitor process.

References

[1] Caillaud, B., Darondeau, P., Hélouët, L., and Lesventes L.: HMSCs as Partial Specifications ... with PNs as Completions. in *Modeling and Verification of Parallel Processes*, Springer-Verlag LNCS **2067** (2001) 125-152.

[2] Duboc, C.: Mixed Product and Asynchronous Automata. *Theoretical Computer Science* **48**, 3 (1986) 183-199.

[3] Genest, B., Kuske, D., and Muscholl, A.: A Kleene Theorem and Model Checking for a Class of Communicating Automata. In *IC* **204** (6): 920-956 (2006).

[4] Genest, B., Muscholl, A., and Peled, D.: Message Sequence Charts. in *Lectures on Concurrency and Petri Nets*, Springer-Verlag LNCS **3098** (2004), 537-558.

[5] Genest, B., Muscholl, Seidl, H., and Zeitoun, M.: Infinite-state High-level MSCs: Model-checking and Realizability. Proc. ICALP, Springer-Verlag LNCS **2380** (2002) 657-668. Journal version in JCSS **72** (4): 617-647 (2006).

[6] Hélouët, L., Jard, C.: Conditions for synthesis of communicating automata from HMSCs. Proc of FMICS, Berlin, 3-4 avril 2000.

[7] Henriksen, J., Mukund, M., Narayan Kumar, K., Sohoni, M. and Thiagarajan, P.S.: A theory of regular MSC languages. *IC* **202** (1): 1-38 (2005).

[8] Klein, J., Caillaud, B., and Hélouët, L.: Merging scenarios. Proc. FMICS, Linz, Austria (2004) 209-226.

[9] Lohrey, M., and Muscholl, A.: Bounded MSC communication. *Information and Computation* **189** (2): 160-181 (2004).

[10] Madhusudan, M., and Meenakshi, B.: Beyond Message Sequence Graphs. Proc. F-STTCS, Springer-Verlag LNCS **2245** (2001) 256-267.

[11] Mukund, M., Narayan Kumar, K. and Sohoni, M.: Bounded time-stamping in message-passing systems. *TCS* **290** (1): 221-239 (2003).

[12] Muscholl, A., Peled, D., and Su,Z.: Deciding properties of message sequence charts. Proc. FOSSACS, Springer-Verlag LNCS **1378** (1998) 226-242.

[13] Muscholl, A. and Peled, D.: Message Sequence Graphs and Decision Problems on Mazurkiewicz Traces. Proc. MFCS, Springer-Verlag LNCS **1672** (1999) 81-91.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399