Weakening the Weak Sequential Composition in Scenarios

Loïc Hélouët

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Weakening the Weak Sequential Composition in Scenarios

Loïc Hélouët

N°6262

Septembre 2005

Systèmes communicants

ISSN 0249-6399

Weakening the Weak Sequential Composition in Scenarios

Loïc Hélouët

Systèmes communicants
Projet DISTRIBCOMM

Rapport de recherche n ° 6262 — Septembre 2005 — 28 pages

Abstract: Message Sequence Charts are a popular formalism for the design of distributed systems executions based on pomset composition. However, this formalism in its basic form is not expressive enough to model typical behaviors such as sliding windows executions. A solution is to embed in MSCs the expressive power of communicating automata, as in CMSCs (another extension of Message sequence charts). However, most basic problems become undecidable for CMSCs. This paper proposes an extension to message sequence charts which extends the expressive power of MSCs while preserving the decidability of some properties. This modification extends sequential composition, but still rely on compositions of closed communication patterns (messages are emitted and received in the same pomset). This paper gives a definition of this new formalism called "sliding" message Sequence charts (or SMSCs). SMSCs can be ill-formed, and we provide a decision procedure to detect and transform such SMSCs inti well-formed ones. Then, the expressive power of SMSCs is compared to that of HMSCS, CMSCs, sure CMSCs and HMSC projections.

Key-words: Scenarios, weak sequence

(Résumé : tsvp)

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

Afaiblissement de la composition faible de scénarios

Résumé : Les Message Sequence Charts sont un formalisme bien connu qui permet de décrire des comportements de systèmes distribués au moyen d'ordres partiels. Cependant, dans sa forme initiale, ce formalisme n'est pas suffisamment expressif pour décrire les comportements typiques de protocoles actuels, comme par exemple les phénomènes de fenêtres glissantes. Une solution proposée dans les CMSCs consiste à intégrer aux MSCs la puissance d'expression des automates communicants. cependant, la plupart des propriétés de base qui étaient décidables sur les MSCs deviennent indécidables dans les CMSCs. Cet article propose une extension des MSCs qui étend leur puissance d'expression tout en préservant la décidabilité de certaines propriétés des MSCs. Cette modification étend la composition séquentielle de MSCs, mais s'appuie toujours sur une composition de motifs de communications clos (les messages sont émis et reçus dans le même ordre). Cet article donne une définition de ce nouveau formalisme appelé "sliding" Message Sequence Charts (ou SMSC par la suite). Les SMCs peuvent être incohérents, et nous fournissons une procédure de décision pour transformer des MSCs incohérents en MSCs cohérents. La puissance d'expression des SMCs est ensuite comparée à celle des MSCs, des CMSCs, et des projections de MSCs.

Mots-clé : Scenarios, composition séquentielle

Contents

1 Introduction

Several scenario languages have appeared this last decade: Message Sequence Charts [8], Live Sequence Charts [7], UML's sequence diagrams[5], ... These languages depict interactions between objects of a distributed system, such as a communication protocol. Very often, distributed systems are specified using process algebras or communicating automata. While these models represent systems via parallel composition of communicating machines, scenarios are defined as sequential composition of communication patterns. Despite the syntactical differences between scenario languages, they are all built from the same idea: they provide composition means for pomsets. In this paper, we will mainly focus on High-level Message Sequence Charts (HMSCs) and their variants.

HMSCs can be seen as order automata generating pomset families. The transitions of a HMSC are labeled with basic Message Sequence Charts (bMSC), which are roughly speaking pomsets describing an interaction among components of a distributed system. Pomset composition is supposed closer to human understanding of distributed behaviors, as it explicitly represents communications (a message emission and reception usually appears in a single communication pattern). This is why scenarios are often recommended for requirement capture, or to describe typical executions of distributed system. However, HMSCs are not expressive enough to model behaviors such as executions of sliding windows protocols. This can be considered as a real drawback for a language that must represent typical behaviors of distributed systems.

To solve this problem, several variants of MSCs have been proposed. Compositional Sequence Charts [6] (or CMSCs for short) are an extension of HMSCs, where communications are not always defined within a single basic pattern. Roughly speaking, compositional sequence charts are HMSCs extended with the asynchronous communications of communicating automata. However, this formalism has a bigger expressive power than communicating automata, and some simple problems such as deciding whether a message is sent and received in at least an execution of a CMSC are undecidable. Note that several trace related problems (language inclusion, intersection etc.) are already undecidable for HMSCs([11]), but that some questions regarding the structure of the pomset family generated remain decidable for a large subset of HMSCs ([10] has identified a class of HMSC for which the pomset family generated is MSO definable).

Another drawback of CHMSCs is that some orders generated by the support automaton contain more receptions than emissions. Hence, they cannot be considered as valid executions of CHMSC. This is a real drawback of the language, as the invalid orders are not immediately visible as a structural property of the automaton. A subset of CMSCs called "sure" CMSCs has been proposed. A CMSC is sure iff all orders generated by the support automaton contain the same number of emissions and receptions. Several properties which were undecidable in general for CHMSC become decidable for sure CHMSC [2, 4].

Projection of HMSCs [3] (or pHMSCs for short) is another approach for the definition of sliding windows executions. PHMSCs are given by a HMSC H and a set of preserved events. The pomset family generated by a pHMSC is the projection of the family of H on preserved events. [3] shows that pHMSC (up to certain limitations) have the same expressive power

as sure CHMSCs. However, defining sliding windows as a projection of a larger HMSC is not easy, as it would consist in creating a hidden instance every time two messages cross.

This paper proposes a new extension to High-level Message Sequence Charts, based on a slight modification of sequential composition. So far, sequential composition of two bMSCs is always weak sequential composition, and this does not allow the construction of scenarios where communications cross. In fact, in weak sequential composition, scenarios are piled up one after another. This composition method is somehow too restrictive to allow sliding windows definition. We propose a new sequence for pomsets, that allow a limited overlapping of composed orders.

This paper is organized as follows. Section 2 recalls some basic definitions on message sequence charts. Section3 recalls the definition of pHMSCs, Section 4 recalls the definitions and problems of CHMSCs. Section 5 introduces our new formalism, and Section 6 compares the expressive power of sliding scenarios with that of other formalisms. Section 7 concludes this work.

2 Message Sequence Charts

This section recalls some basic definitions about Message Sequence Charts (or MSC for short). MSC is a standardized language [8] equipped with a formal semantics [13, 12]. A *Basic Message Sequence chart* (or bMSC for short) is a pomset labeled with action names and locations. bMSCs depict simple exchanges among communicating entities called *instances*. Communications are supposed asynchronous, and events along an instance axis are totally ordered. For a complete description of the core language, interested readers may consult [14].

Definition 1 *A* bMSC *is a tuple* $B = (E, \leq, A, I, \alpha, \phi, m)$ *where* E *is a set of events,* $\leq$ *is a partial order relation on* E *(reflexive, transitive, antisymmetric),* A *is a set of action names,* I *is a set of processes,* $\alpha : E \longrightarrow A$ *is a labeling function, and* $\phi : E \longrightarrow I$ *associates a locality to each event,* $m \subseteq E \times E$ *is a relation that pairs message emissions and receptions.*

The set of events E *can be partitioned into a set of sending events* E_S*, a set of receiving events* E_R *and a set of internal actions* E_A*. In addition to the classical definition of bMSCs, one can associate a type to each sending or receiving event. This type is a function* $t : E \longrightarrow MSG \times I \times I$ *which associates a message name from a finite message alphabet* MSG*, a sending and a receiving instance to each communication event.*

Graphically, a bMSC is represented by vertical axes symbolizing instances. Communications are represented by arrows from the emitting instance to the receiving instance. Figure 1-b shows 3 examples of bMSCs called M1, M2 and M3. bMSCs represent closed communication patterns, i.e. all messages sent in a bMSC are received in the same diagram. ($\forall e \in E_S, \exists e' \in E_R$ such that $(e, e') \in m$). For a given bMSC $B = (E, \leq, A, I, \alpha, \phi, m)$, we will denote by $\sharp_i(B)$ the number of events situated on instance $i \in I$ (i.e. $\sharp_i(B) = |\phi^{-1}(i)|$). For a given event $e \in E$, we will denote by $rk(e)$ the *rank* of event e on instance $\phi(e)$, i.e.

$rk(e) = |\{x \in E | \phi(x) = \phi(e) \wedge x \leq e\}|$. Basic MSCs alone are not powerful enough to define complex behaviors, and several bMSC composition operators (sequence choice, loops, ...) have been proposed.

Definition 2 *The* sequential composition *of two bMSCs* $B_1 = (E_1, \leq_1, A_1, I_1, \alpha_1, \phi_1, m_1)$ *and* $B_2 = (E_2, \leq_2, A_2, I_2, \alpha_2, \phi_2, m_2)$ *is the bMSC* $B_1 \circ B_2 = (E_1 \uplus E_2, \leq_{1\circ 2}, A_1 \cup A_2, I_1 \cup I_2, \alpha_1 \uplus \alpha_2, \phi_1 \uplus \phi_2, m_1 \uplus m_2)$, *where* $\leq_{1\circ 2} = \left(\leq_1 \uplus \leq_2 \uplus \{(e_1, e_2) \in E_1 \times E_2 \mid \phi(e_1) = \phi(e_2)\}\right)^*$.

Intuitively, sequential composition glues bMSCs along their common instance axes. A common way to compose bMSCs is to define a kind of partial order automaton, called High-level Message Sequence Chart(or HMSC for short).

Definition 3 *A* HMSC *is a tuple* $H = (N, \longrightarrow, \mathcal{B}, n_0, F)$, *where* N *is a set of nodes,* n_0 *is a specific node called the "initial node",* $\mathcal{B}$ *is a set of bMSCs,* $\longrightarrow \subseteq N \times \mathcal{B} \times N$ *is a set of transitions,* F *is a set of accepting nodes.*

A HMSC defines a set of paths, i.e. sequences of transitions of the kind $p = n_0 \xrightarrow{B_0} n_1 \xrightarrow{B_1} n_2 \ldots \xrightarrow{B_{k-1}} n_k$ from the initial node to an accepting node, that will be denoted by $\mathcal{P}_H$. HMSCs allow for the definition of infinite behaviors composed of closed communication patterns. For a given path $p = n_0 \xrightarrow{B_0} n_1 \xrightarrow{B_1} n_2 \ldots \xrightarrow{B_{k-1}} n_k$, we will denote by O_p the bMSC obtained by successive concatenation of labels of p : $O_p = B_1 \circ B_2 \cdots \circ B_{k-1}$. The set of partial orders generated by a HMSC is denoted $\mathcal{F}_H$ and is defined as follows: $\mathcal{F}_H = \{O_p | p \in \mathcal{P}_H\}$. However, HMSCs do not allow for the definition of very classical behaviors such as sliding windows executions. Figure 1-a shows an example of behavior that can not be modeled by a HMSC (for an arbitrary number of messages). This is a real drawback for scenarios, as this kind of behavior is very usual in distributed systems. This sliding window execution can be finitely generated by hiding some elements of the orders generated using HMSC projection [3]. Another approach has been proposed, and extends MSCs to obtain communicating Message Sequence Charts [6].

3 PHMSCs: HMSC projections

HMSC projection were not initially designed to enhance the expressive power of HMSCs, but rather as an abstraction for verification purposes [3]. However, HMSC projections are expressive enough to design sliding windows. Consider, for example the HMSC Figure 1-b. A projection of this HMSC on instances A and B generates a set of executions similar to the chronograms depicted in Figure 1-a).

Definition 4 *A projection of a bMSC* $B = (E, \leq, A, I, \alpha, \phi, m)$ *on a set of events* $E' \subseteq E$ *is noted* $\Pi_{E'}(B)$, *and is the restriction of* B *to events of* E'. $\Pi'_E(B) = (E', \leq \cap (E' \times E'), A, I, \alpha_{|E'}, \phi_{|E'}, m \cap (E' \times E'))$. *For a given instance* $i \in I$, *we will often denote by* $\pi_i(B)$ *the projection of* B *on events located on instance* i, *that is* $\pi_i(B) = \Pi_{\phi^{-1}(i)}(B)$. *The*

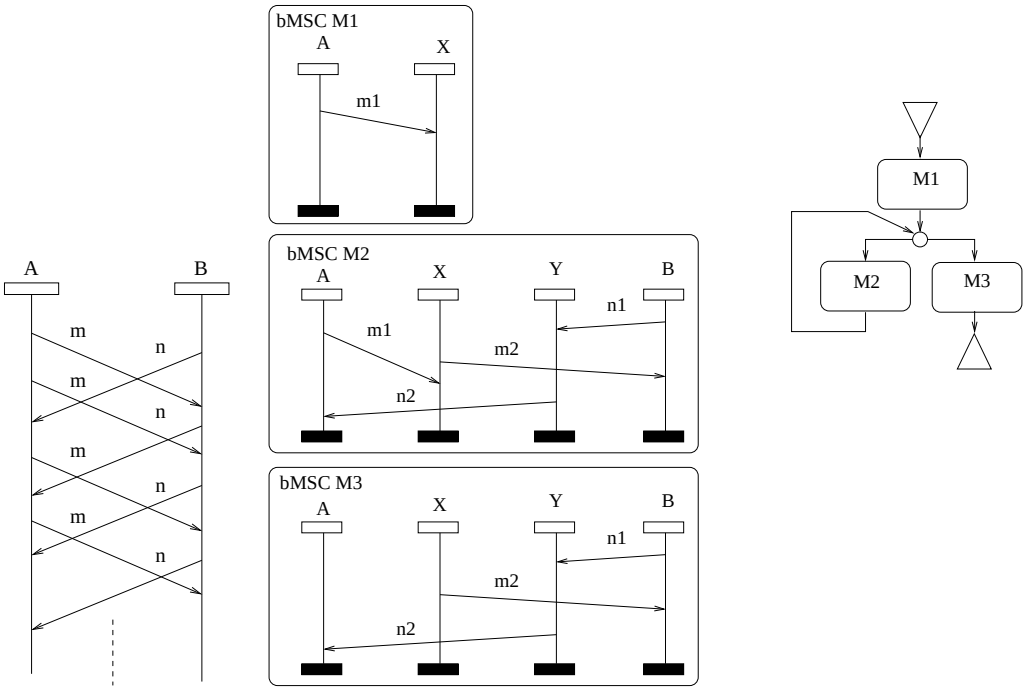

Figure 1: a) A sliding window chronogram b) Design of a sliding window with PHMSCs

notion of projection can be defined similarly for HMSCs, and consists in projecting the orders generated on a subset of events.

Let $H = (N, \longrightarrow, \mathcal{B}, n_0, F)$ be a HMSC. Let us denote by E_H the set of events that are defined by bMSCs of $\mathcal{M}$, and let $E \subseteq E_H$. The projection of H on E' is noted $\Pi_{E'}(H)$ and defines a partial order family $\mathcal{F}_{\Pi_{E'}}$ obtained by restriction of orders in $\mathcal{F}_H$ to all occurrences of events in E'.

A drawback of MSC projections is that after abstraction, the notion of message, or event type may not be as obvious as in bMSCs. Consider, for example the HMSC of Figure 1-b. Projecting this HMSC on instances A and B produces a set of partial orders that is isomorphic to set of orders depicted Figure 1-a. However, to obtain a sliding window where messages of type m and n cross each other, a designer would have to specify in addition to the initial HMSC and to the projection that the image of messages $m1$ and $m2$ via projection should be renamed as m and similarly that $n1 + n2$ should be renamed n. This is not very convenient, even if feasible. Hence, PHMSCs do not preserve event names. Another problem is that pHMSCs do not preserve event types either. In a projection, a causal relation from an event to another located on a different instance is not always a message. If we take as convention that in projections, causalities from events situated on an instance i to other events on another instance $j \neq i$ are messages (assuming, for example that the new message name is the name of the first message leaving i that causes this causal relation), there is still a problem : some events are at the same time emissions and receptions of messages. These events will be called "multitype events". Consider, for example, Figure 2. bMSC M is projected on events e, f and j. In M, event f is a message reception, but in $\Pi_{e,f,j}(M)$, the type of f is not so clear, as it is a causal successor of e and a predecessor of j. Hence, the orders generated by projection cannot be considered as bMSCs anymore.

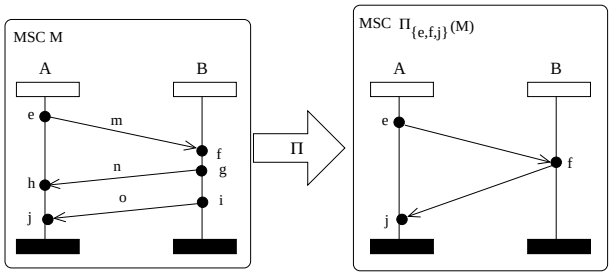

Figure 2: Simple projection of a bMSC

Another drawback of PHMSCs is that the partial order families generated are not necessarily obtained by concatenation of projected orders ($\Pi(M_1 \circ M_2) \neq \Pi(M_1) \circ \Pi(M_2)$), and need to be defined as the abstraction of the whole partial order family $\mathcal{F}_H$. Furthermore, to model communications crossings as in example of Figure 1, a designer needs to introduce hidden instances, i.e., objects that have no meaning in the system under design. Hence, M-

SC projections are more adapted as an heuristic to decide properties concerning the causal orders between events in a system than for modeling.

4 CHMSCS

Communicating Message Sequence charts were first introduced by [6], and extend HMSCs with communications inherited from communicating automata. Another variant of CMSCs with different message matching mechanisms was also proposed in [1].

Definition 5 *A* CMSC *is a bMSC where m is a partial mapping, i.e some sending events are not mapped to a receiving event, receptions are not the image of any message emission. In addition to this, [6] and [2] require m to be FIFO, i.e. for any pair of message emissions e, f located on the same instance such that $e \leq f$, if $\phi(m(e)) = \phi(m(f))$ then $m(e) \leq m(f)$.*

A CMSC is left-closed *if it does not contain unmatched receptions (i.e for all $e \in E_R$, reception, $\exists e'$ such that $(e', e) \in m$). A CMSC that is not left closed cannot be considered as the execution of a communication protocol, as some message are received but never sent. We denote by $UR(M)$ the set of unmatched receptions of M, by $US(M)$ the set of unmatched emissions of M. For a given message $t \in MSG$, we denote by $UR_t(M)$ the set of unmatched receptions of messages of type t, and by $US_t(M)$ the set of unmatched emissions of a message of type t.*

The sequential composition of two CMSCs C_1 and C_2 is slightly different from that of bMSCs: it merges orders along their common instances, but also pairs unmatched emissions of C_1 with unmatched receptions of C_2.

Definition 6 *The* sequential composition *of two CMSCs $C_1 = (E_1, \leq_1, A_1, I_1, \alpha_1, \phi_1, m_1)$ and $C_2 = (E_2, \leq_2, A_2, I_2, \alpha_2, \phi_2, m_2)$ is defined if and only if :*

- *C_1 is left-closed,*
- *$\forall t \in Msg$, $|UR_t(C_2)| \leq |US_t(C_1)|$ (the number of unmatched receptions of messages of type t is not greater than the number of unmatched sends of the same type of message in M_1),*
- *For all $e, e' \in E_2$ such that $(e, e') \in m_2$ and e, e' define a message of type t, $|UR_t(C_1)| = |US_t(C_1)|$. This requirement helps preserving the FIFO property of messages.*

The sequential composition is defined as for bMSCs, with the difference that it pair emissions of C_1 with receptions of C_2. Let $map : E_1 \longrightarrow E_2$ be the (unique) mapping that maps the i^{th} unmatched send event of message m with the i^{th} unmatched reception of the same type. Then, the sequential composition of C_1 and C_2 produces the cMSC $C_1 \circ C_2 = (E_1 \uplus E_2, \leq_{1\circ 2}, A_1 \cup A_2, I_1 \cup I_2, I_2, \alpha_1 \cup \alpha_2, \phi_1 \cup \phi_2, m_1 \uplus m_2 \uplus map)$, where $\leq_{1\circ 2} = (\leq_1 \uplus \leq_2 \uplus \{(e_1, e_2) \in E_1 \times E_2 | \phi(e1) = \phi(e2)\} \uplus m_1 \uplus m_2 \uplus map\})^*$

Definition 7 *A* CHMSC *is a HMSC labeled by an alphabet of CMSCs $\mathcal{C}$. Similarly to HMSCs, a CHMSC $C = (N, \longrightarrow, \mathcal{C}, n_0, F)$ defines a set of path $\mathcal{P}_C$, and to each path $p = n_0 \xrightarrow{C_1} n_1 \ldots \xrightarrow{C_k} n_k$, we can associate the CMSC $C_p = C_1 \circ C_2 \ldots C_k$ obtained by sequential composition. However, sequential composition is not defined for all paths of $\mathcal{P}_C$ (some path do not generate left-closed CMSCs). We say that a CHMSC H is* realizable *iff $\forall p = n_0 \xrightarrow{M_0} n_1 \xrightarrow{M_1} n_2 \ldots n_k \in \mathcal{P}_H$, $\forall i \in 1..k-1$, $(M_1 \circ \cdots \circ M_i) \circ M_{i+1}$ is defined. Realizability is decidable in $O(|I|^2.|N|.| \longrightarrow |)$ (it mainly consists in verifying that cycles in a CHMSC do not produce more receptions than emissions).*

A CHMSC C is sure *iff all orders generated by paths of C contain only matched messages, i.e, $\forall p \in \mathcal{P}_C, \forall t \in MSG, |UR_t(M_p)| = |US_t(M_p)| = 0$. This property is also decidable in polynomial time [2].*

In [3] we show that MSC projections without multitype events have the same expressive power as sure CHMSCs. CHMSCs are very expressive, as with unmatched emissions and receptions, they embed the expressiveness of communicating automata. However, as it often happens, this power has a counterpart, and some simple problems become undecidable for CHMSCs. For example, [6] shows that showing the existence of a path p such that a message m is sent and received in C_p is equivalent to providing a solution for Post's correspondence problem. Note that with HMSCs, answering such question is trivial, as messages are sent and recived within the same bMSC. Figure 3 shows a CHMSC specification for which providing an algorithmic solution to detect if message m is sent and received in a path implies an algorithmic solution to PCP for instance $\{(ab, abb), (bb, bba), (baa, ab)\}$. Clearly, this CHMSC can not be encoded with an equivalent sure CHMSC, as several iterated cMSCs contain more receptions than emissions.. So, embedding the expressive power of communicating automata into a scenario language is not either a good solution, and CMSCs can be considered as too generic. A solution is to restrict the use of CMSCs to sure CMSCs, as proposed by [2]. Another possibility is to find a new class of scenarios based on closed patterns, that increases the expressive power of HMSCs, and for which most of HMSCs decidable properties remain decidable.

5 Sliding Scenarios

This section proposes a new scenario language that contains HMSCs, allows the definition of sliding window-like behaviors while avoiding the expressive power of communicating automata. For this, we allow some overlapping between pomsets in sequential composition. The size of the overlapped zones is defined by a function (called hereafter a *shifter*) attached to each bMSC.

Definition 8 *A* shifter *is a function $f : I \longrightarrow \mathbb{N}$ that maps any instance name $i \in I$ to an integer. The* shifted sequential composition *of two bMSCS $B_1 = (E_1, \leq_1, A_1, I_1, \alpha_1, \phi_1, m_1)$ and $B_2 = (E_2, \leq_2, A_2, I_2, \alpha_2, \phi_2, m_2)$ with a shifter $f : I_2 \longrightarrow \mathbb{N}$ is noted $B_1 \circledcirc_f B_2$, and is defined as $B_1 \circledcirc_f B_2 = (E_1 \uplus E_2, \leq_\circledcirc, A1 \cup A2, I_1 \cup I_2, \alpha_1 \uplus \alpha_2, \phi_1 \uplus \phi_2, m_1 \uplus m_2)$, where :*

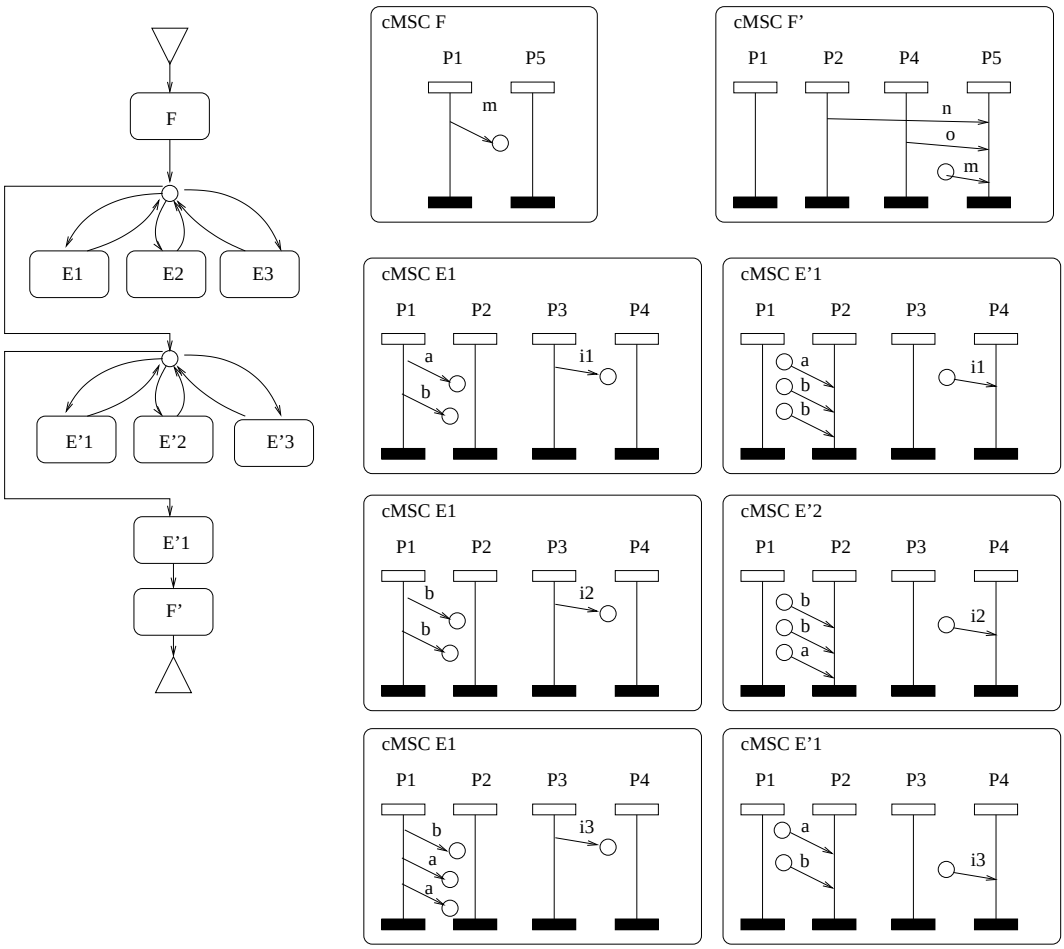

Figure 3: PCP encoded with CHMSCs

$$\leq_{\circledcirc}= \left(\begin{array}{l} \leq 1 \uplus \leq 2 \\ \uplus\{(e_1,e_2) \in E_1 \times E_2 | \phi(e_1) = \phi(e_2) = i \wedge rk(e_1) \leq rk(e_2) + \sharp_i(B_1) - f(i)\} \\ \uplus\{(e_2,e_1) \in E_2 \times E_1 | \phi(e_1) = \phi(e_2) = i \wedge rk(e_1) > rk(e_2) + \sharp_i(B_1) - f(i)\} \end{array} \right)^*$$

Definition 9 *Let $w = w_1.w_2 \dots w_q$ and $v = v_1.v_2 \dots v_p$ be two words. Let $k \in \mathbb{N}$. We will denote by $head_k(v)$ the word $v_1 \dots v_k$, by $tail_k(v)$ the word $v_{p-k+1} \dots v_p$, and by $v||_k w$ the word*
$v_1 \dots v_{p-k}.v_{p-k+1}.w_1 \dots . v_p.w_p.w_{p+1} \dots w_q$. *Less formally, $v||_k w$ interlaces the k last letters of v with the k first letters of w, and leave the rest of v and w unchanged. Note that for two bMSCs B_1 and B_2, and for a shifter f, the projection $\pi_i(B_1 \circledcirc_f B_2)$ is equivalent to $\pi_i(B_1)||_{f(i)}\pi_i(B_2)$.*

Figure 4 illustrates the effects of shifted sequential composition on the respective order of events on a single instance i. If we compose M1 and M2 with a shifter f such that $f(i) = 3$, then we obtain the sequence of events on the rightmost part of the figure. Note that the last 3 events of M1 and the first 3 events of M2 are interlaced, and not concurrent. Figure 5 shows a shifted concatenation of two bMSCs $M1$ and $M2$ with a shifter f such that $f(A) = 2$ and $f(B) = 0$.

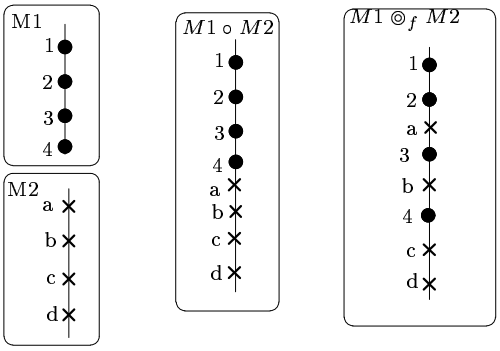

Figure 4: The effects of shifted composition on a single instance

Definition 10 *For two bMSCs B_1, B_2 and a shifter f, the relation $\leq_{\circledcirc}$ obtained by shifted concatenation $M_1 \circledcirc_f M_2$ is not always a partial order. We will say that a bMSC is* well-formed *iff $\leq_{\circledcirc}$ is a partial order relation (reflexive, transitive and antisymmetric).*

Let us illustrate this with two examples. The previous example of Figure 5-a is well-formed. Now, let us Consider bMSCs $M1$ and $M2$ in Figure 5-b. If we compose $M1$ and $M2$ with a shifted composition $\circledcirc_f$ such that $f(A) = 2$ and $f(B) = 0$, we obtain a scenario where message m can only be sent after reception of n, and conversely. Hence, $\leq_{\circledcirc_f}$ is not a partial order. Well formedness of a scenario can be detected in linear time using Tarjan's

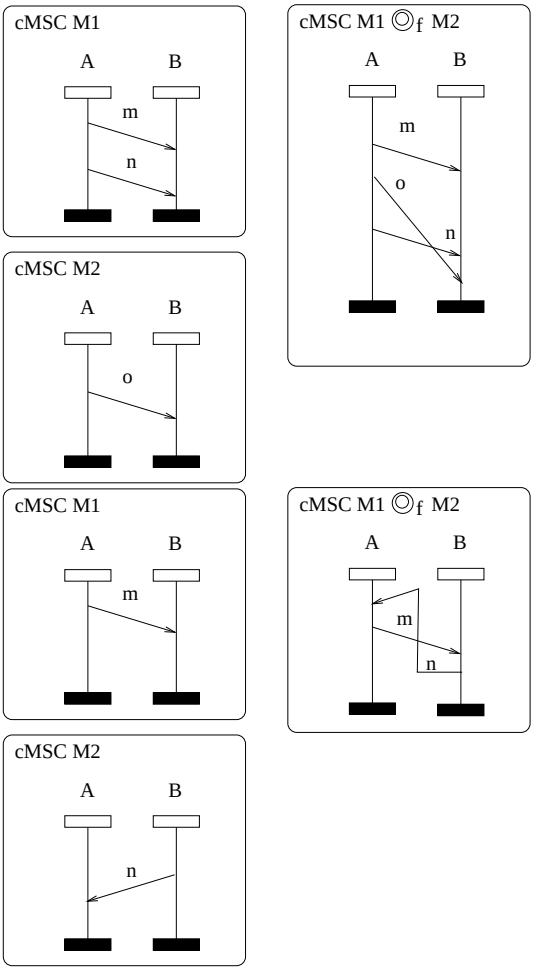

Figure 5: a) well-formed composition b) ill-formed composition

algorithm [15]. It consists in finding strongly connected components of the graph $(E, \leq_{\circledcirc_f})$ that are not singletons. As for HMSCs, shifted concatenation alone is not sufficient to model complex behaviors. The shifted concatenation mechanisms easily extends to HMSCs.

Definition 11 *A* Sliding HMSC *(or SHMSC for short) is a tuple* $S = (N, \longrightarrow, \mathcal{B}, n_0, F, \mathcal{S}, \mu)$, *where* N, $\longrightarrow$, $\mathcal{B}$, n_0, F *have the usual meaning for HMSCS, and :*

- $\mathcal{S}$ *is a set of shifters,*
- $\mu : N \times \mathcal{B} \longrightarrow \mathcal{S}$ *associates a shifter* $\mu_{n,B}$ *to a bMSC* B *in a given node* n. *In addition, we will suppose that for any transition* $n \xrightarrow{B} n'$, $dom(\mu(n, B)) = \phi(B)$.

Though this composition mechanism may seem a bit artificial, it does exactly what we need : it generates families of orders that allow sliding windows-like behaviors. For example, the sliding window of Figure 1 is generated by the SHMSC of Figure 6. To provide a graphical representation of shifters, we define them as a list of integers between two brackets, and suppose that the values of shifters associated to a bMSC at a given node associates the i^{th} value to the i^{th} instance (in lexicographical order). In the example of Figure 6 for instance, $\mu_{n_1,M2}(A) = 0$ and $\mu_{n_2,M1}(A) = 2$.

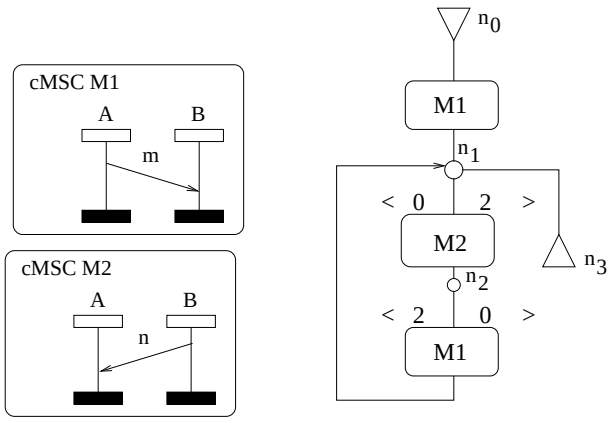

Figure 6: A SHMSC generating the sliding window of Figure 1

A SHMSC defines a set of paths $\mathcal{P}_S$ similar to the set defined by a HMSC. The order associated to a path $p = (n_0, B_0, n_1)(n_1, B_1, n_2)\ldots(n_k, B_k, n_{k+1}) \in \mathcal{P}_S$ is the order $M_p^{\circledcirc} = B_0 \circledcirc_{\mu(n_1,B_1)} B_1 \circledcirc_{\mu(n_2,B_2)} \cdots \circledcirc_{\mu(n_k,B_k)} B_k$. SHMSCs can also be considered as generators for partial order families. Note that similarly to Communicating Message Sequence Charts, the partial order family generated by a SHMSC does not always reflect the structure of the automata, as shifted concatenations that are not well-formed are not part of the generated families. This forces to define properties of SHMSCs.

Definition 12 *A path $p \in \mathcal{P}_S$ is* consistent *iff:*

i) (no self-overtaking) $\forall B \in \mathcal{B}$ that appears at least twice in p, $\forall e \in E_B$ and $\forall k \in \mathbb{N}$, the i^{th} occurrence of e precedes the $i + k^{th}$ occurrence of e in $M_p^{\circledcirc}$,

ii) (well formedness) $M_p^{\circledcirc}$ is well-formed.

A SHMSC S is consistent if and only if all path of $\mathcal{P}_S$ are consistent.

When a SHMSC satisfies i) but not ii), we will say that it is ill-formed. The family of partial orders generated by a SHMSC S is defined as $\mathcal{F}_S = \{M_p^{\circledcirc} | p \in \mathcal{P}(S) \wedge M_p^{\circledcirc}$ consistent $\}$. Let us prove that SHMSC consistency is a decidable property. First, we will study self-overtaking.

Definition 13 *Let $S = (N, \longrightarrow, \mathcal{B}, n_0, F, \mathcal{S}, \mu)$ be a SHMSC and i be an instance. The sequences of events generated by the shifted concatenations on i are the outputs of a transducer T_{S_i} that reads words in $\mathcal{B}^*$ and outputs words in $\bigcup_{B \in \mathcal{B}} E_B$. It is defined as $T_{S_i} = (Q, \Sigma_{in}, \Sigma_{out}, \delta, Q_F, q_0)$, where :*

- $\Sigma_{in} = \mathcal{B} \cup \{\epsilon\}, \Sigma_{out} = \bigcup_{B \in \mathcal{B}} E_B \cup \{\epsilon\}$
- $Q \subseteq N \times \bigcup_{i \in 1..K-1} \Sigma_{out}^i$ *where* $K = max\{s(i) | s \in \mathcal{S}\}$
- $q_0 = (n_0, \epsilon), Q_F = F \times \{\epsilon\}$
- $\delta \subseteq Q \times \Sigma_{in} \times \Sigma_{out} \times Q$ *is a set of transitions built in the following way:*
  - $(q, B, v_{out}, q') \in \delta$ *iff* $q = (n, v)$, $q' = (n', v')$, *and* $\exists n, B, n' \in \longrightarrow$, *and* $v' = tail_k(v||_{\mu_{n,B}(i)}\pi_i(B))$, *and* $v_{out} = head_{k-1}(v||_{\mu_{n,B}(i)}\pi_i(B))$, *or*
  - (q, ϵ, v, q') *iff* $q = (n, v)$, $(n, \epsilon, n') \in \longrightarrow$ *and* $q' = (n', \epsilon)$ *is a final state.*

More intuitively, the transducer associated to each instance memorizes all events that may be overtaken by future shifted concatenations of a bMSC (the number of overtaken events cannot exceed K). Transitions to an accepting state "purge" the memory of the transducer. Figure 7 shows a SHMSC S and the transducer obtained for instance A for the SHMSC S2.

Let us illustrate transducer construction on the examples of Figure 8. Clearly, SHMSC S1 contains self-overtaking, as the second emission of message n overtakes the first emission of n. Note that in a different context a loop on bMSC M2 with the same shifter does not necessarily produce self-overtaking (see for example SHMSC S2 in the same figure). The transducers associated to sliding MSCS S1 and S2 are provided Figure 9. One can note that from state $(n1, !n)$ of T_{S1_A}, firing transition $M2/!n$ produces an overtaking.

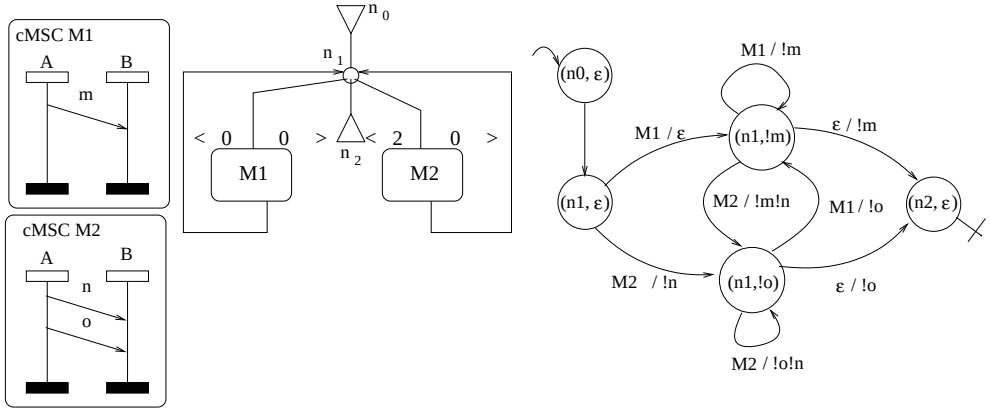

Figure 7: Construction of a transducer

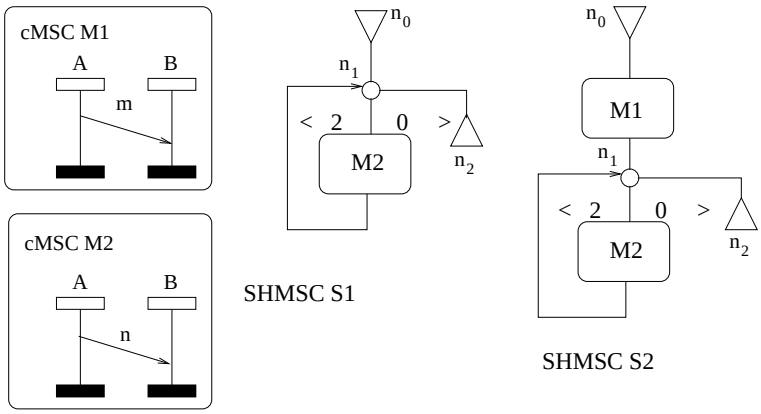

Figure 8: Two sliding MSCs

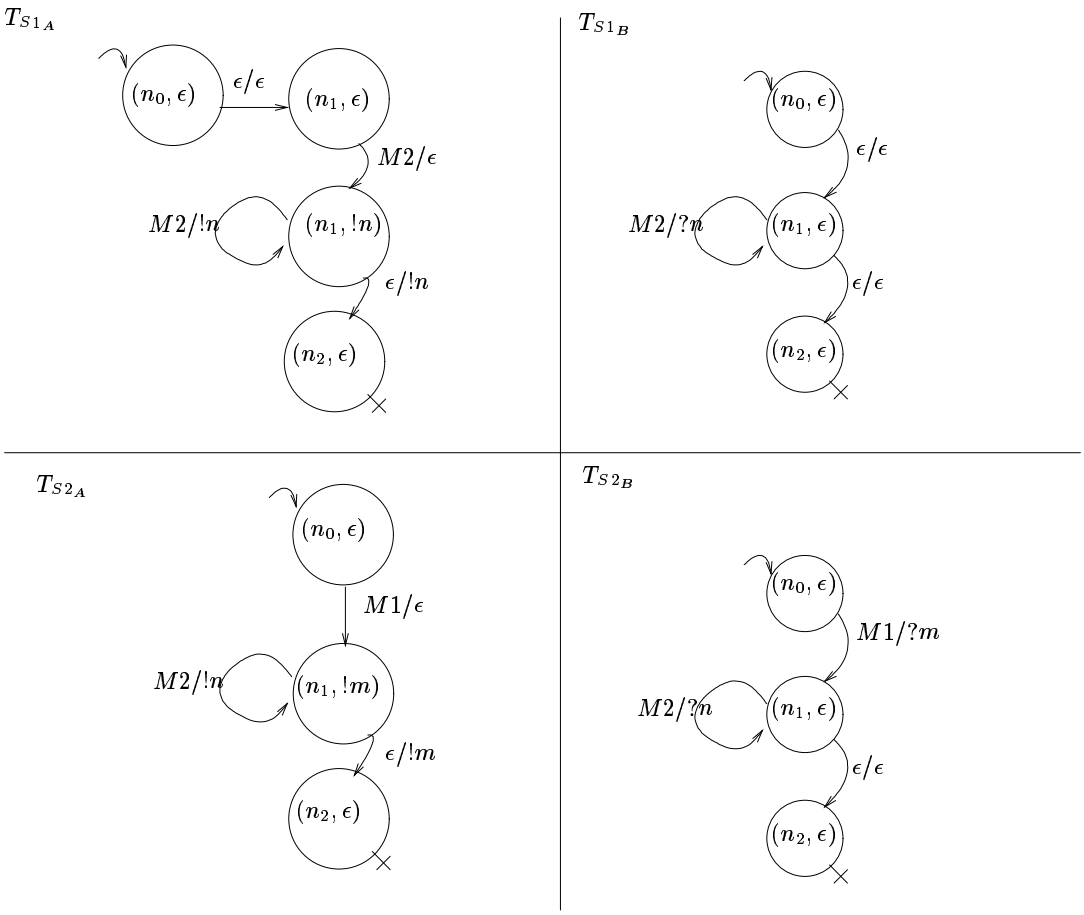

Figure 9: transducers for SHMSCs of Figure 8

Proposition 1 *Let S be a sliding HMSC, and let i be an instance of S. Then, S contains self-overtaking on instance i if and only if $\exists q = (n, w)$ reachable state of the transducer T_{S_i} such that $n \xrightarrow{B} n'$ is a transition of S, $\mu_{n,B}(i) = x$, $w = v.e.v'$ and $\pi_i(B) = u.e.u'$ with $|v'| + |u| + 1 - x < 0$.*

For a SHMSC such that each shifter is bounded by a constant k, and such that the number of events located on each instance of S is bounded by a constant r (i.e. $\forall i \in I, |\bigcup_{B \in \mathcal{B}} E_B \cap \phi^{-1}(i)| \leq r$), self-overtaking can be decided in time at most $O(|I| \times |N| \times |\mathcal{B}| \times r.\frac{1-r^k}{1-r})$. The projection of the shifted concatenation on an instance i can be defined as the output of a transducer that reads as input MSC names, and that outputs shifted sequences of events. The language of MSCs (the input of our transducer) of a SHMSC S is regular. Hence, for any instance i of a SHMSC S, the language $\pi_i(\mathcal{F}_S)$ is *regular*. This property of projections is important: preserving a regular language on all instance means that SHMSCs could be implemented by simple communicating automata (when such implementation is possible). More elaborated compositions with, for example addition of a semi-commutation relation on instance projections would rapidly have lead to non-regular languages on instances. Note also that the projection on an instance is a sequence of events. We could also have allowed some concurrency between events of both operands. However, with this kind of composition, projection on an instance can be a partial order of unbounded width. Regularity can also be useful when trying to detect when a SHMSC is equivalent to a HMSC (a non-regular language could rapidly have meant that this property was undecidable). So far, we have only proved that point i) was decidable for SHMSCs. Let us now show that there is a finite decision procedure for point ii).

Definition 14 *Let E be a set of events, R be a binary relation on these events, and $e \in E$. We will denote by $\downarrow_R e$ the set $\downarrow_R e = \{e' \in E | e' R^* e\}$ and by $\uparrow_R f$ the set $\uparrow_R e = \{e' \in E | eR^* e'\}$. Of course, when R is a partial order relation, then $\uparrow_R e \cap \downarrow_R e = \emptyset$ for any event $e \in E$. A binary relation R on E_M will be called* cyclic *iff there exists two elements $e, f \in E_M$ such that $eR^* f$, $e \neq f$ and $fR^* e$. The* transitive restriction *of R is the relation $R' = \{(e, e') \in E^2 | eRe' \wedge e \neq e' \wedge \nexists x \in E \setminus \{e, e'\}, eRxRe'\}$ (R' removes from R reflexivity and transitive relations). Note that when R is cyclic, its transitive restriction is also cyclic. Detecting cycles in a relation is equivalent to building a partition of E into subset of elements that are connected in a graph (E, R).*

Proposition 2 *Let $S = (N, \longrightarrow, \mathcal{B}, n_0, F, \mathcal{S}, \mu)$ be a SHMSC. If for all $p = n_0 \xrightarrow{B_1} n_1 \ldots n_k$, path of S such that $M_p^{\circledcirc}$ is well-formed, and for all transition $n_k \xrightarrow{B} n_{k+1}$ such that n_{k+1} is coaccessible, $M_p^{\circledcirc} \circledcirc_{\mu(n_k,B)} B$ is well-formed, then all paths of S generate well-formed orders.*

proof: Adding B to a path p such that $M_p^{\circledcirc}$ is well formed builds the well-formed order $M_{p'}^{\circledcirc}$ for path $p' = n_0 \xrightarrow{M_1} n_1 \ldots n_k \xrightarrow{B} n_{k+1}$. Hence, if for all B and all context the concatenation of B creates a well-formed order, then S does not generate ill-formed order. □.

This proposition shows that, in order to study well-formedness of a SHMSC, there is no need to consider all paths of a SHMSC. Showing the absence of cycles after concatenation of a single bMSC at the end of any path is sufficient. Now, let us show that for a given bMSC B, we can also work on a finite number of finite abstractions of the orders generated by a SHMSC before concatenation of B. A first intuition is that after shifted concatenation, the only memory needed is the relative order on events in $\uparrow E_B \cap \downarrow E_B$ (see Figure 10). However, this ordered set of events is not always finite.

Definition 15 *Let $B = (E, \leq, A, I, \alpha, \phi)$ be a bMSC. Let $K = (k_{i_1}, k_{i_2}, \ldots k_{i_{|I|}})$ be a sequence of integers where $(i_1, \ldots i_{|I|})$ is the sequence of instance names in lexicographical order. The* K-reduction *of B is noted $R_K(B)$, and is the projection of B on $E' = \{e \in E_B | \sharp_{\phi(e)}(B) - rk(e) \leq k_{\phi(e)}\}$. More intuitively, $R_K(B)$ is the projections of B on the k_i last events of each instance. From now, for a set of instance I and a function $f : I \longrightarrow \mathbb{N}$, we will denote by K_f the sequence $K_f = \big(f(i_1), \ldots f(i_{|I|})\big)$ (K_f is the image by f of I sorted in lexicographical order).*

Proposition 3 *Let O and B be two well-formed bMSCs. Let μ be a shifter, and let $K = K_\mu$ be the integer sequence associated to μ and I_B. Any cycle in the covering relation of $O \circledcirc_\mu B$ contains at least one event of B and one event of $R_K(O)$.*

proof: The first point is obvious: as O is acyclic, any cycle created in the covering of the order relation is due to the addition of order passing through at least one event of B. Note that all immediate successors of events of B in $O \circledcirc_\mu B$ are either events of E_B, or events situated on the same instance. As the cycles in the covering relation pass through at least one event of E_B, there must be a pair of events $e \in E_B$ and $f \in O$ such that $\phi(e) = \phi(f)$. From the definition of shifted sequential composition, we know that $f \in R_K(O)$. □

This proposition is important, as it shows that to detect that adding a bMSC B to an order O produces a well-formed order, it is sufficient to consider the projection of $O \circledcirc_\mu B$ on events of $R_K(O)$ and B.

Proposition 4 $R_K(O) \circledcirc_\mu B = \Pi_{E_{R_K(O)} \cup E_B}(O \circledcirc_\mu B)$ *(composition of a bMSC B with a K-reduction is equivalent to a projection on events of B and events preserved by reduction)*

proof: To prove this proposition we need the following property of relations and closures. Let $R1$, $R2$ be two binary relations on a set of elements E, and let $X \subseteq E$. Let us denote by $Int_{R1,R2}(x, y)$ the set $\uparrow_{R1} x \cap \downarrow_{R2} y$. If $\forall x, y \in X$ such that $x \neg R1 y$ and $x \neg R2 y$, $Int_{R1,R2}(x, y) \neq \emptyset$ implies $Int_{R1,R2}(x, y) \cap X \neq \emptyset$, then $\big((R1 \cap X^2) \cup (R2 \cap X^2)\big)^* = (R1 \cup R2)^* \cap X^2$ (in general, this equality does not hold, as restriction to elements of X may cut a chain from an element to another that cannot be rebuilt via transitive closure). So, this property only holds when transitivity is not lost during abstraction. We can also note that for any pair of relations R_1,R_2, $\big(R_1 \cup R_2\big)^* = \big(R_1^* \cup R_2^*\big)^*$.

From the definition of projection, we know that $\Pi_{R_K \cup B}(O \circledcirc_\mu B) = (O \circledcirc_\mu B) \cap (R_K \cup B)^2$. Using the definition of shifted composition, this can be rewritten as

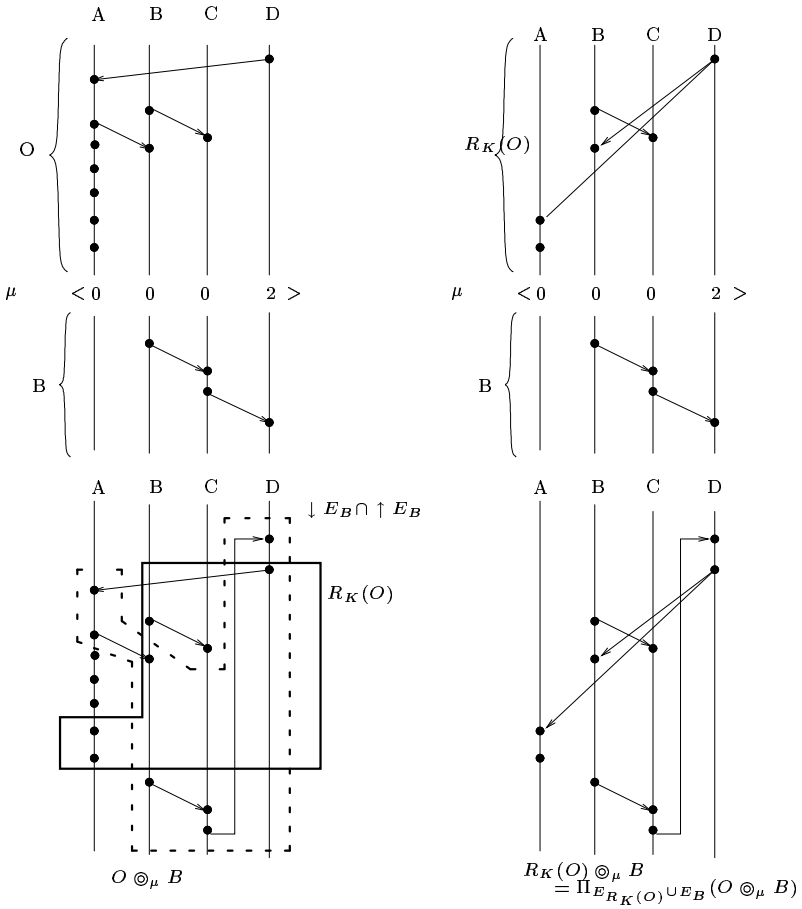

Figure 10: Equivalence between K-restriction composed with a bMSC and projection with $K = (2, 2, 1, 2)$ and $\mu\big((A, B, C, D)\big) = (0, 0, 0, 2)$

.

$$\Pi_{R_K\cup B}(O \circledcirc_\mu B) = \left(\begin{array}{l} \leq_O \cup \leq_B \cup \\ \{(o,b) \in O \times B | \phi(o) = \phi(b) = i \wedge rk(o) \leq rk(b) + \sharp_i(O) - \mu(i)\} \cup \\ \{(b,o) \in B \times O | \phi(o) = \phi(b) = i \wedge rk(o) > rk(b) + \sharp_i(O) - \mu(i)\} \end{array} \right)^* \cap$$
$(R_K(O) \cup B)^2$.

Without loss of causality we can write that

$$\Pi_{R_K\cup B}(O \circledcirc_\mu B) = \left(\begin{array}{l} \leq_O \cup \leq_B \cup \\ \{(r,b) \in R_K(O) \times B | \phi(r) = \phi(b) = i \wedge rk(o) \leq rk(b) + \sharp_i(O) - \mu(i)\} \cup \\ \{(b,r) \in B \times R_K(O) | \phi(r) = \phi(b) = i \wedge rk(o) > rk(b) + \sharp_i(O) - \mu(i)\} \end{array} \right)^* \cap$$
$(R_K(O) \cup B)^2$.

Let us denote by $\leq_2$ the relation

$$\leq_2 = \left(\begin{array}{l} \leq_B \cup \\ \{(o,b) \in O \times B | \phi(o) = \phi(b) = i \wedge rk(o) \leq rk(b) + \sharp_i(O) - \mu(i)\} \cup \\ \{(b,o) \in B \times O | \phi(o) = \phi(b) = i \wedge rk(o) > rk(b) + \sharp_i(O) - \mu(i)\} \end{array} \right)^*$$

Then, we have $\Pi_{R_K(O)\cup B}(O \circledcirc_\mu B) = (\leq_o \cup \leq_2)^* \cap (R_K(O) \cup B)^2$. The next step is to show that $\forall x, y \in R_K(O) \cup B$, such that $x \not\leq_O y$ and $x \not\leq_2 y$, $Int_{\leq_O,\leq_2}(x,y) \neq \emptyset$ implies $Int_{\leq_O,\leq_2}(x,y) \cap (R_K(O) \cup B) \neq \emptyset$. Let us suppose that $Int_{\leq_O,\leq_2}(x,y) \neq \emptyset$, but $Int_{\leq_O,\leq_2}(x,y) \neq \emptyset \cap (R_K(O) \cup B) = \emptyset$. So, in the transitive reduction of $\leq_O \cup \leq_2$, any causal chain from x to y passes through at least one event of $O \setminus (R_K(0) \cup B)$. Furthermore, as $x \not\leq_O y$, we cannot have $x \leq z$, $z \leq x'$ and $x' \leq y$ with $x, x' \in R_K(O)$, $z \in O \setminus (R_K(O) \cup B)$. It also means that a causal chain from x to y passes through an event $b \in B$. Hence, this means that there is a pair (o,b) or a pair (b,o) with $b \in B$ and $o \in O \setminus (R_K(O) \cup B)$ in $\leq_O \cup \leq_2$, which is impossible, and then leads to a contradiction.

So now that we are sure that $\forall x, y \in R_K(O) \cup B$, such that $x \not\leq_O y$ and $x \not\leq_2 y$, $Int_{\leq_O,\leq_2}(x,y) \neq \emptyset$ implies $Int_{\leq_O,\leq_2}(x,y) \cap (R_K(O) \cup B) \neq \emptyset$, we can write :

$$\Pi_{R_K(O)\cup B}(O \circledcirc_\mu B) = \Big(\big(\leq_O \cap (R_K(O) \cup B)^2\big) \cup \big(\leq_2 \cap (R_K(O) \cup B)^2\big) \Big)^* = R_K(O) \circledcirc_\mu B. \ \square$$

Proposition 5 *$O \circledcirc_\mu B$ is well-formed if and only if $\Pi_{E_{R_K(O)}\cup E_B}(O \circledcirc_\mu B)$ is also well-formed. (projection preserves well-formedness)*

proof: This is a consequence of Proposition 3. As the order relation is transitive, whenever two events e, f appear in a cycle, then $e \leq f$ and $f \leq e$, and any projection containing these two events is cyclic. $\square$

Proposition 6 *$O \circledcirc_\mu B$ is well-formed if and only if $R_K(O) \circledcirc_\mu B$ is well-formed. (reduction preserves well-formedness)*

proof: From the preceding propositions, $O \circledcirc_\mu B$ cyclic implies that there is a cycle passing through two events $e \in E_B, f \in R_K(O)$, and hence $\Pi_{R_K\cup B}(O \circledcirc_\mu B)$ contains a cycle, and so does $R_K(O) \circledcirc_\mu B$. Conversely, when $R_K(O) \circledcirc_\mu B$, so does $\Pi_{E_{R_K(O)}\cup E_B}$. As $\Pi_{E_{R_K(O)}\cup E_B}(O \circledcirc_\mu B) \subseteq O \circledcirc_\mu B$, $O \circledcirc_\mu B$ also contains a cycle. $\square$

Proposition 7 *Let O, B be two well-formed bMSCs. Let μ be a shifter, and let K_μ be the sequence associated to μ and I_B. Then $\forall K' > K_\mu$, $R_{K_\mu}(O) \circledcirc_\mu B$ is well-formed if and only if $R_{K'}(O) \circledcirc_\mu B$ is well-formed.*

proof: From proposition 6, we know that $R_{K_\mu}(O) \circledcirc_\mu B$ is well formed iff $O \circledcirc_\mu B$ is well formed. As $K' > K_\mu$, we have $R_{K_\mu}(R'_K(O)) = R_{K_\mu}(O)$, so $R_{K_\mu}(O) \circledcirc_\mu B$ well-formed if and only if $R_{K_\mu}(R'_K(O)) \circledcirc_\mu B$ is also well-formed. Hence, $R_{K_\mu}(O) \circledcirc_\mu B$ iff $R'_K(O) \circledcirc_\mu B$ well-formed. □

From these propositions, we can show that cycle creation can be studied on a finite set of finite approximations of all orders generated by a SHMSC. Note that the detection of cycles in the order relation does not need to take into account the labeling of MSCs, and only concerns the structure of the orders generated by concatenation. For this reason, we define a token representation for bMSCs as follows.

Definition 16 *Let B be a bMSC. The* token representation *of B is a partial order $B^T = (T, \leq_T)$ where $T \subseteq \mathbb{N} \times I$ is a set of tokens, and $\leq_T$ is a partial order relation, defined as:*

- $T = \{(n, i) \in \mathbb{N} \times I | \exists e \in E, \phi(e) = i \wedge n = rk(e)\}$
- $\leq_T = \left\{ \left(\big(rk(e), \phi(e)\big), \big(rk(e'), \phi(e')\big) \right) | e \leq e' \right\}$

The token representation of a bMSC is isomorphic to $(E, \leq)$. Concatenation, projections, and K-reductions are defined similarly for token representations and bMSCs. From now, let us denote by $\mathcal{T}_I$ the set of all token representations of bMSCs defined on a given set of instances I. For a given sequence $K = (n_1, \ldots n_{|I|})$ of integers such that $|I| = |K|$, the set of K-reductions of $\mathcal{T}_I$ is finite, and is denoted by $\mathcal{T}_I^K$. So far, the size of $\mathcal{T}_I^K$ is not known, but [9] gave an upper bound $P_n = O(2^{\frac{n^2}{4}}.2^{n^{\frac{3}{2}}.ln(n)})$ for the number of partial orders on a set of elements of size n. Note however that the number of bMSC reductions must be much lower, as ordering on instances is a total order. For a SHMSC S, as we know that the number of order reductions is finite, we can compute a finite unfolding of S the nodes of which are pairs (n, T), such that n is a node of S, and T represents a finite reduction of an order generated so far by S.

Definition 17 *Let $S = (N, \longrightarrow, \mathcal{B}, n_0, F, \mathcal{S}, \mu)$ be a SHMSC. The* token unfolding *of S is the SHMSC $\mathcal{S}^{\mathcal{T}} = (Q, \longrightarrow', \mathcal{B}, q_0, F', \mathcal{S}, \mu')$, where :*

- *$Q \subseteq \mathcal{N} \times \mathcal{T}_I^{Kmax}$, where $Kmax = \big(max_{\mu \in \mathcal{S}}\{\mu(i_1)\}, \ldots max_{\mu \in \mathcal{S}}\{\mu(i_{|I|})\}\big)$ is the sequence of maximal values associated by the set of shifters to each instance. Note that we require all states in H^T to be composed of a node of N, and of a well-formed reduction.*
- *$q_0 = (n_0, T_{Kmax})$, where T_{Kmax} is a token representation with $max_{\mu \in \mathcal{S}}\{\mu(i)\}$ events on each instance $i \in I$, and such that $\forall (n, i), (m, j) \in T$ with $i \neq j$, $(n, i) \not\leq (m, j)$ and $(m, j) \not\leq (n, i)$*

- $F' = \{(n,T) | n \in F\}$
- $\mu' : Q \times \mathcal{B} \longrightarrow \mathcal{S}$ *associates a shifter* $\mu_{n,B}$ *to a couple* (q,B) *iff* $q = (n,T)$ *fr some token representaion* T.
- $\longrightarrow' = \{ ((n,T),B,(n',T')) | n \xrightarrow{B} n' \wedge T' = R_{Kmax}(T \circledcirc_{\mu_{n,B}} B^T) \wedge T'$ *well-formed* $\}$

Theorem 1 *Let* S *be a SHMSC, and* $S^{\mathcal{T}}$ *be its associated token unfolding.* S *only generates well-formed orders iff for all* (n,T)*, reachable state of* $S^{\mathcal{T}}$*, for all* $B \in \mathcal{B}$ *such that* $n \xrightarrow{B} n'$*,* $T \circledcirc_{\mu_{n,B}} B$ *is well-formed.*

proof: The proof comes easily from the preceding propositions: if a state with token representation T is reachable, then there is a path $p = n_1 \dots n_k$ such that $T = R_K(M_p^{\circledcirc})^T$. Then if for some B such that there existst a transition $n_k \xrightarrow{B} n_{k+1}$, $T \circledcirc B$ is not well-formed, then, for $p' = n_1 \dots n_k \xrightarrow{B} n_{k+1}$, $M_{p'}^{\circledcirc}$ is not well-formed. Conversely, if there is a path p that is such that $M_p^{\circledcirc}$ not well-formed, we can split it into two parts $p1$, $p2$ such that $p = p1.p2$ and $M_{p1}^{\circledcirc}$ is the maximal well formed order for a prefix of p. Hence, p generates a cyclic order relation only after a first transition $n \xrightarrow{B} n'$ of $p2$. $T = R_K(M_{p1}^{\circledcirc}$ is a well-formed token representation, and there is a reachable state (n,T) is $S^{\mathcal{T}}$. According to proposition 6, firing transition $n \xrightarrow{B} n'$ from (n,T) necessarily creates an ill-formed order $T \circledcirc B$. □

Detecting a cycle in a graph can be performed with a complexity that is linear in the size of the graph [15]. The size of each restriction is at most $(|I| * k)^2$, and detecting a cycle in a shifted composition $R_K(O) \circledcirc_\mu B$ has a maximal complexity of $O\big((|I| * k + |E_B|)^2\big)$. Note that we do not have to study all preorders generated by paths of the SHMSC to detect ill-formed SHMSCs, but only the orders. For a SHMSC H which bMSCs define interactions among a set of instances I, and for a given set of shifters S such that $\forall f \in S, i \in I,\ f(i) \leq k$, the number of possible restrictions is lower than $T = O(2^{\frac{(|I|.k)^2}{4}}.2^{(|I|.k)^{\frac{3}{2}}.ln(|I|.k)})$. The number of states in $H^{\mathcal{T}}$ is at most $|N| \times T$ (we do not have to study order reductions with less than k events on each instance, as we can start our search from an order reduction B^T containing k events on each instance and such that $(n,i),(m,j) \in T$ with $i \neq j$,). We have to study the effects of all transitions from each well-formed state. So, if we assume that the size of the biggest bMSC in $\mathcal{B}$ is b, well formedness of H can be decided in time lower than $O\big((|I|.k+b)^2 \times |\mathcal{B}| \times N \times 2^{\frac{(|I|.k)^2}{4}}.2^{(|I|.k)^{\frac{3}{2}}.ln(|I|.k)}\big)$.

Theorem 2 *Let* S *be an ill-formed SHMSC, such that all shifter are bounded by a constant* k*. Then, there exists a well-formed SHMSC* S' *with at most* $O\big(|\mathcal{B}| \times N \times 2^{\frac{(|I|.k)^2}{4}}.2^{(|I|.k)^{\frac{3}{2}}.ln(|I|.k)}\big)$ *transitions such that* $\mathcal{F}_{S'} = \mathcal{F}_S$*. For any inconsistent SHMSC* S*, such that the number of events on each instance of* S *is lower or equal to* r*, there is also a consistent SHMSC* S'' *generating* $\mathcal{F}_S$ *with at most* $O\big(|\mathcal{B}| \times N \times |I| \times r.\frac{1-r^k}{1-r} \times 2^{\frac{(|I|.k)^2}{4}}.2^{(|I|.k)^{\frac{3}{2}}.ln(|I|.k)}\big)$ *transitions.*

proof: When an order O is ill-formed, any concatenation of a bMSC to O remains ill-formed. So, we can take for S' the SHMSC $S^{\mathcal{T}}$. $S^{\mathcal{T}}$ suppresses all transitions creating

an order that is not well-formed, the extensions of which are not contained in the pomset family generated by S. Similarly, we can take for S'' the synchronous product of S' and of all instance transducers from which transitions generating self-overtaking have been discarded. □

6 Comparison with other languages

Note that the undecidable reachability problem of Section 4 becomes decidable for SHMSCs. Hence, SHMSCs do not contain CHMSCS, and the intersection between SHMSCs and CHMSCs is less expressive than CHMSCs. It is also worth comparing SHMSCs with HMSCs and other partial order composition languages such as HMSC projections. HMSCs are trivially embedded in SHMSCs, as a SHMSC equipped with a single shifter $\mu(n, B) : I \longrightarrow 0$ for all $\in N$ and all $B \in \mathcal{B}$ is a HMSC.

The example of Figure 11 shows that not all SHMSCs are CHMSCs. It is possible to define a CHMSC that generates a family of partial orders that contains all orders generated by $S2$, but not a CHMSC that defines the same family. This example cannot either be modeled with a PMSC. Furthermore, we know since [3] that the class of PMSCs without multitype events is contained in the class of CMSCs. Conversely, the example Figure 12 shows that some CMSCS are not SHMSCs.

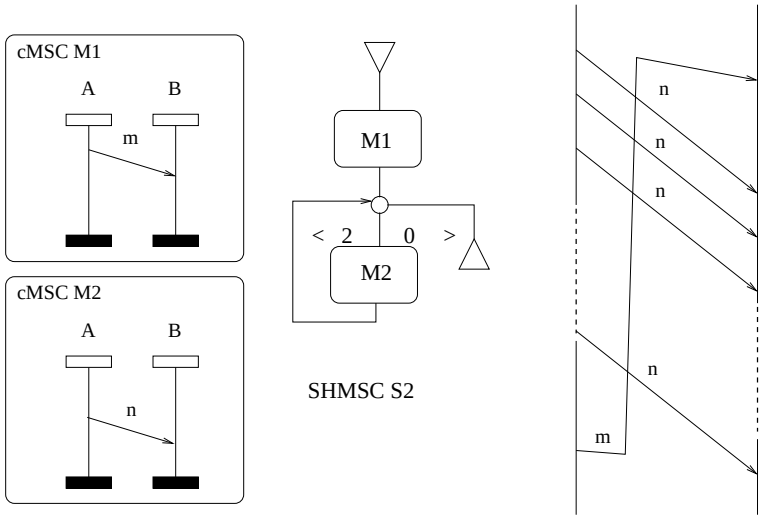

Figure 11: SHMSCS are not embedded in CHMSCs

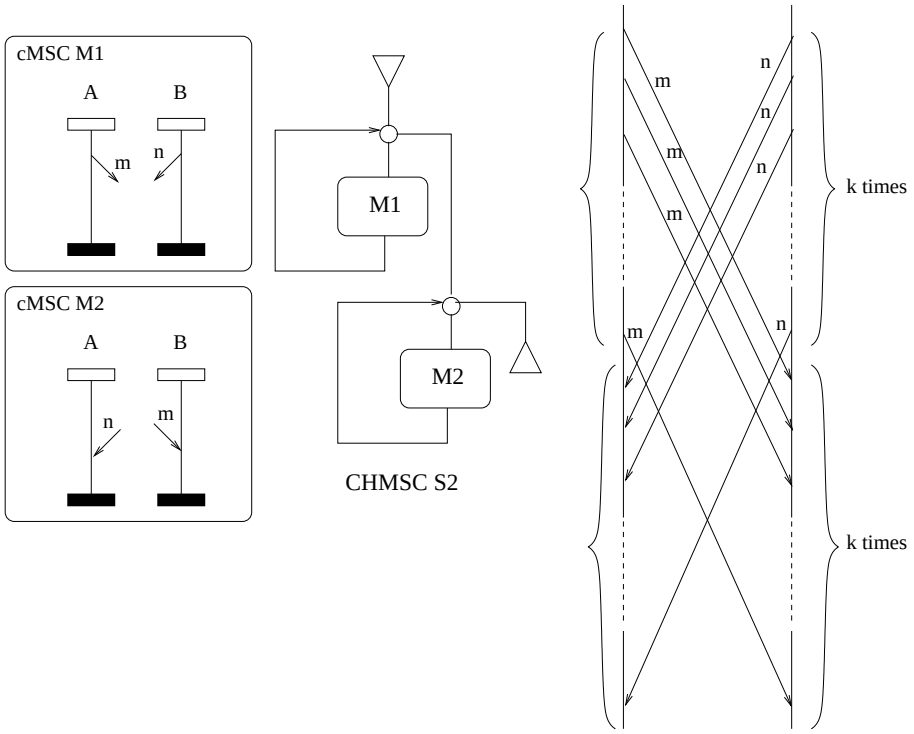

Figure 12: CHMSCS are not embedded is SHMSCS

Proposition 8 *Let us denote by $P, C, S,$ and Sure the sets of partial order families generated by HMSC projections, CHMSCs, SHMSCs and Sure CHMSCs. Using the results of [3], we know that the following properties hold.*

i) $P \cap C = Sure$

ii) $\forall s \in S, \forall M_p^{\circledcirc} \in \mathcal{F}_s, M_p^{\circledcirc}$ *is sure*

iii) $(S \cap C) \subseteq Sure$

iv) $(S \cap C) \subseteq (P \cap C)$

proof of statements i) to iv): Point $i)$ is a result of [3]. Point $ii)$ is true by definition of Sliding MSCs. Let us prove point $iii)$. Let us consider a partial order family $\mathcal{F}$ generated by a sliding MSC S and a CHMSC C. C does not contain loops labeled by a CHMSC containing more receptions than emissions (otherwise the family generated by C and S can not be similar), or such loop are not co-accessible. So, for any path p of C,the number of emissions of each message m that are not yet received in O_p is bounded by a constant B_m. Adding to O_p more than $max\{Bm | m$ not received in $O_p\}$ times a CHMSC containing unmatched receptions of some unmatched messages produces an order with more receptions than emissions, as these emissions will never be matched by further concatenations. This order is not contained in $\mathcal{F}$. Hence, when there is loop of C that contains more emissions than receptions it can be replaced in C by a finite number of iterations of this loop, or it is not coaccessible. Hence C can be rewritten in an equivalent cHMSC C' that is sure. This proves point $iii)$. Point $iv)$ is a consequence of $i)$ and $iii)$.□

From these results, we can depict the relationships between languages in Figure 13.

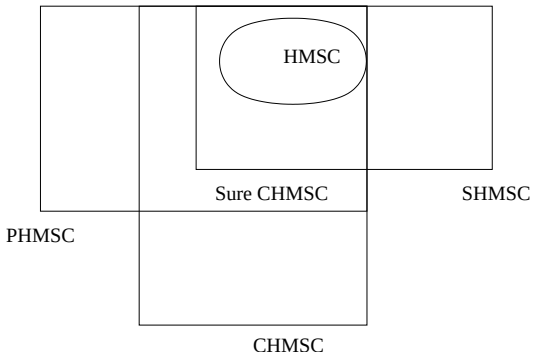

Figure 13: A topology of scenario languages

7 Conclusion

This paper has proposed an extension of weak sequence that allows creation of sliding windows-like behaviors from a set of closed communication patterns. We have shown that well-formed SHMSCs and ill-formed SHMSCs have the same expressive power. The class of partial order families that can be represented by SHMSC is disjoint from the class of families represented by PHMSCs and CHMSCs. An interesting question is whether the intersection of SHMSCs and CMSCs has the same expressive power as MSC projections without multitype events. If the answer is yes, reusing the results of [3] would give immediately an algorithm to decide the existence of a HMSC generating the same partial order family for a given SHMSC in this intersection.

Another open issue is the complexity of the algorithm for well-formedness detection. The upper bound given in this paper relies on the number of partial orders that exist on a set of n elements, which grows rapidly. However, the partial orders generated by shifted composition and restriction are peculiar: they have a bounded width and events are totally ordered on instances. This could help reducing the complexity of the algorithm.

References

[1] B. Bollig, M. Leucker, and P. Lucas. Extending compositional message sequence graphs. In *Proc of LPAR 2002*, 2002.

[2] B. Genest. *L'odyssée des MSC graphes.* PhD thesis, Université Paris 7, nov. 2004.

[3] B. Genest, L. Hélouët, and A. Muscholl. High-level message sequence charts projection. In *Proc. of CONCUR'03*, number 2761 in LNCS, pages 311–326. Springer Verlag, Sep 2003.

[4] B. Genest, D. Kuske, and A. Muscholl. A kleene theorem for a class of communicating automata with effective algorithms. In *Proc of DLT 2004, Eighth International Conference on Developments in Language Theory*, number 3340 in LNCS, pages 30–48, dec. 2004.

[5] Object Management Group. Uinified modeling language specification version 2.0: Superstructure. Technical Report pct/03-08-02, OMG, 2003.

[6] E. Gunter, A. Muscholl, and D. Peled. Compositional message sequence charts. In *Proc. of TACAS'01*, LNCS, 2001.

[7] D. Harel and W. Damm. Lscs: breathing life into message sequence charts. Technical Report CS98-09, Weizmann Institute, Avril 1998.

[8] ITU-TS. *ITU-TS Recommendation Z.120: Message Sequence Chart (MSC).* ITU-TS, Geneva, September 1999.

[9] D.J Kleitman and B.L Rotschild. Asymptotic enumeration of partial orders. *Transactions of the American Mathematical Society*, (205):205–220, 1975.

[10] R. Morin. Recognizable sets of message sequence charts. In *STACS*, pages 523–534, 2002.

[11] A. Muscholl and D. Peled. Message Sequence Graphs and decision problems on Mazurkiewicz traces. In *Proceedings of MFCS'99*, LNCS 1672, 1999.

[12] M. Reniers. *Message Sequence Charts: Syntax and Semantics*. PhD thesis, Eindhoven University of Technology, 1998.

[13] M. Reniers and S. Mauw. High-level message sequence charts. In A. Cavalli and A. Sarma, editors, *SDL97: Time for Testing - SDL, MSC and Trends*, Proc. of the 8th SDL Forum, pages 291–306, Evry, France, September 1997.

[14] E. Rudolph, P. Graubman, and J. Grabowski. Tutorial on message sequence charts. *Computer Networks and ISDN Systems*, 28:1629–1641, 1996.

[15] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal of Computing*, 1(2), 1992.

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