Decomposition of Message Sequence Charts

Loïc Hélouët, Pierre Le Maigat
IRISA, Campus de Beaulieu
35042 Rennes Cedex, France
{helouet,plemaiga }@irisa.fr

Abstract: High level Message Sequence charts describe compositions of communication patterns, that can be less intuitive than expected. This points out the need for analysis tools. Due to the complexity of languages generated by HMSCS, analysis such as temporal studies generally take as granularity the basic Message sequence charts. For such cases, a decomposition of a HMSC into smaller patterns can provide a finer analysis. It can also enhance the understanding of a specification.

Keywords: Message Sequence Charts, decomposition.

1 Introduction

Message Sequence Charts is a normalized formalism for the definition of scenarios. It is composed of two levels of definition. At a low level of the description, basic Message sequence Charts (bMSCs for short) describe possible communications between processes. Processes are represented by vertical axes, messages by arrows from the emitting to the receiving process. bMSCs are built form very few graphical elements, which allow for a very fast understanding. In addition to this simplicity, concurrency between events can be easily detected.

BMSCs only allow for the specification of simple and finite communication patterns. In order to allow for the description of more elaborated behaviours, High-level Message Sequence Charts (HMSCs for short) have been proposed ([6]). A HMSC allows for the composition of bMSCs using parallel compositions, sequences, alternatives, and loops. Unfortunately, the initial simplicity of bMSCs disappears when they are composed. This is mainly due to the semantics of the sequential composition. The assumed meaning of bMSC sequence is weak sequential composition, i.e a local concatenation along process axis (similar to the local partial order composition defined in [5]). This composition allows events from basic message sequence charts to be executed in an order that do not respect the declared order in the specification, and may therefore seem counter intuitive. Consider, for example, the High-level Message Sequence Chart of Figure 1. Basic Message Sequence Chart M_1 precedes Basic Message Sequence Chart M_2. However, due to the weak sequential composition, sending of message m_3 can be executed before all events of M_1.

So, a HMSC can describe very complex behaviours. This unexpected complexity underlines the need for tools allowing behavioral and temporal analysis of requirements expressed by means of High-level Message Sequence Charts. Due to the complexity of the language generated by a HMSC, temporal analysis tools often use bMSC as the granularity of study [3], or consider strong sequential composition between charts [7], hence reducing

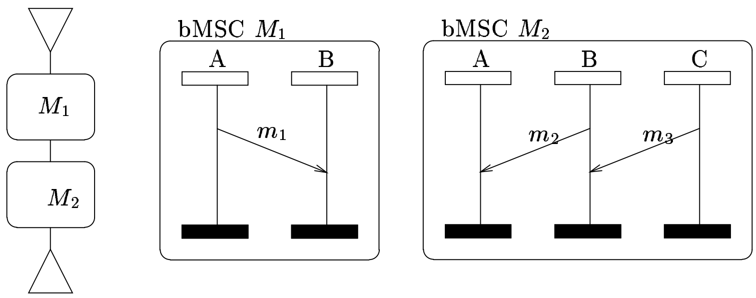

Figure 1: An example of weak sequential composition

the expressivity of the formalism. When weak sequential composition is assumed, using events as the granularity for the analysis is not always possible.

In addition to the granularity problem, all the advantages of a graphical notation such as scenarios can be lost when the specification is too large. When a sequence of communications is too long, detecting concurrency becomes a difficult task. Furthermore, different graphical specification may express the same behaviours. Consider the example in Figure 2: the three specifications express the same behaviours, but are decomposed in a different way. many equivalence problems have been proved undecidable for HMSCs. Yet, one may ask if there is a way of expressing a HMSC as a composition of communication patterns as small as possible, in order to identify obvious equivalences between HMSCs.

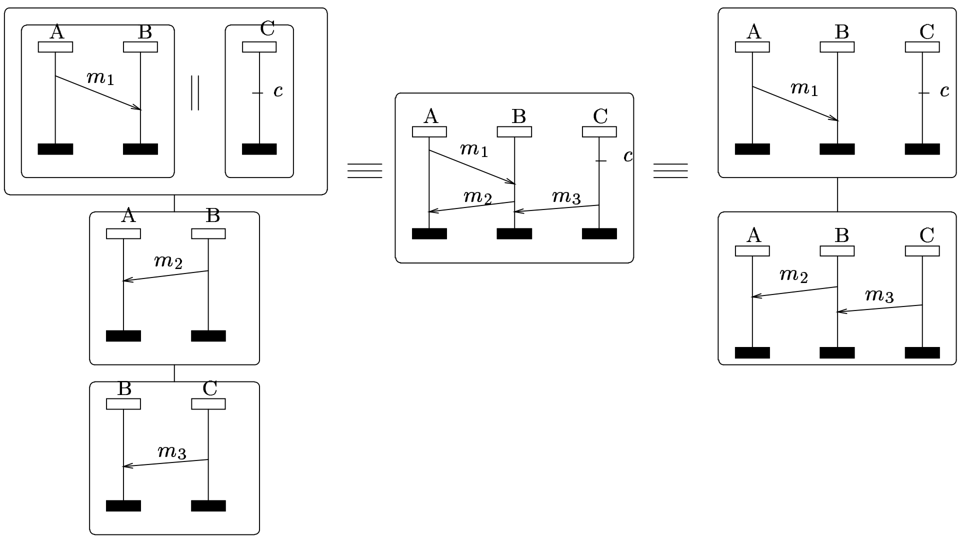

Figure 2: Different decompositions of a MSC

The example of Figure 2 can be treated easily, as messages m_1, m_2, m_3 and action c are disposed in layers. However, MSCs allow for message crossing, and a pattern such as bMSC in Figure 3 can not be decomposed without cutting a message.

We propose an automatic method for decomposing High-level Message Sequence Charts into equivalent HMSCs composing the smallest basic MSCs contained in the specification. This is performed into two steps: first, the bMSCs of the initial specification are decomposed into basic patterns, and then, these patterns are factorized. This article is organized

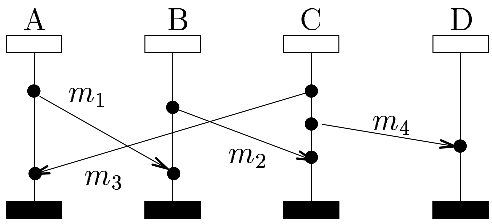

Figure 3: Non decomposable pattern

as follows: section 2 introduces our notation for Message Sequence Charts. Then, section 3 shows how a basic Message Sequence Chart can be decomposed into a set of basic patterns. Section 4 introduces the notion of irreducible message sequence charts, a kind of "normal form" based on composition of basic patterns. Section 5 provides an algorithm for computing an irreducible form for a HMSC, before conclusion.

2 Notations

A Basic Message Sequence Chart define the behaviour of communicating processes. Each process is represented by a vertical axis, and message exchanges are represented by arrows. Clearly, a bMSC defines a precedence relation on a set of events: along the instance axis, events are put in a top-down order (except in specific parts of the axis called coregions), and a message emission must precede the corresponding reception. It seems very natural to provide bMSCs with a non interleaving semantics as in [1, 2]. A bMSC can be defined by a labeled partial order on a set of events $M =< E, \leq, A, I, \alpha >$, where:

- E is a set of events,
- $\leq$ is a partial order relation on events called causal order,
- A is a set of action names,
- I is a set of instance names, denoted by $I(M)$ when the bMSC considered is not obvious,
- $\alpha : E \longrightarrow A \times I$ is a labeling function associating an action name and an instance to each event. We will note $\phi(e) = i$ when $\alpha(e) = (a, i)$.

Definition 1 *Two bMSCs $M =< E, \leq, A, I, \alpha >$ and $M' =< E', \leq', A', I', \alpha' >$ are* independant *if $I(M) \cap I(M') = \varnothing$. The* parallel composition *of two independant bMSCs M and M', is denoted by $M||M'$, and is defined by the bMSC $< E \cup E', \leq \cup \leq', A \cup A', I \cup I', \pi >$, where:*

$$\begin{array}{rccc} \pi : & E \cup E' & \longrightarrow & (A \cup A') \times (I \cup I') \\ & e \in E & \longmapsto & \alpha(e) \\ & e' \in E' & \longmapsto & \alpha'(e') \end{array}$$

When two bMSCs M and M' are not independent, we will write $M \not\parallel M'$. A bMSC N is a parallel component *of M if there exists a bMSC M' such that $M = M'||N$.*

As we are dealing with weak sequential composition, if two bMSCs M_1 and M_2 are independent, then $M_1 || M_2 = M_1 ; M_2$. Parallel composition will be discussed in more detail section 4.

Basic Message Sequence Charts only allow for a finite set of behaviours. In order to compose bMSCs, a higher level formalism called High-level Message sequqnce Charts has been introduced, which allows for parallel composition, alternatives, loops, an hierarchical definitions. Within our approach, we will not consider the parallel composition of highlevel message sequence charts. As a hierarchy can be transformed into a two-level specification (one bMSC level, and one HMSC level), we can give the following definition:

Definition 2 *A HMSC is a graph* $H = < \mathcal{N}, \longrightarrow, q_{init}, Q_{end}, \mathcal{M} >$, *where:*

- $\mathcal{N}$ *is a set of nodes,*
- $\longrightarrow \subseteq \mathcal{N} \times \mathcal{M} \times \mathcal{N}$ *is a transition relation,*
- q_{init} *is the initial node,*
- Q_{end} *is a (possibly empty) set of end nodes,*
- $\mathcal{M}$ *is a set of bMSCs.*

Definition 3 *Let* $M = < E, \leq, A, I, \alpha >$ *be a bMSC. We will say that* $M' = < E', \leq', A', I', \alpha' >$ *is a* sub-bMSC *of* M *if:*

- $E' \subseteq E,\ \leq_{|E'} = \leq',\ A' \subseteq A,\ I' \subseteq I,\ \alpha_{|E'} = \alpha',$
- $!m \in M' \Leftrightarrow ?m \in M'$

The first step for decomposing a bMSC M is to search for the smallest sub-bMSCs of M. These minimal sub-bMSCs are called *basic patterns* of M. The search for such bMSCs is the aim of the next section.

3 Basic Patterns

Definition 4 *A* cutting point *is a subset* $cp \subseteq E$ *such that:*

- $\exists i \in I,\ \forall e \in cp,\ \phi(e) = i$ *(all elements of cp are events of the same instance)*
- $\forall e \in cp,\ \forall e' \leq e,$ *if* $\phi(e') = \phi(e)$ *then* $e' \in cp$, *(all predecessors of an event of cp on the same instance are also in cp)*
- $\forall e \in cp, \{e' \in E, \phi(e') = \phi(e) \wedge e \not\leq e' \wedge e' \not\leq e\} \subseteq cp$

Slightly extending our notations, we will note $\phi(cp) = i$ for a cutting point cp when $\forall e \in cp, \phi(e) = i$. A cutting point defines a "point", on the graphical representation of the instances, situated between two events. Note that our definition do not allow a cutting point to be placed within a coregion.

On the example of Figure 4, cutting points allowed on instance A are: $I_1 = \emptyset$, $I_2 = \{e_1\}$, $I_3 = \{e_1, e_2, e_3\}$, $I_4 = \{e_1, e_2, e_3, e_4\}$, which define points p_1, p_2, p_3, p_4 on instance axis (symbolized by black crosses).

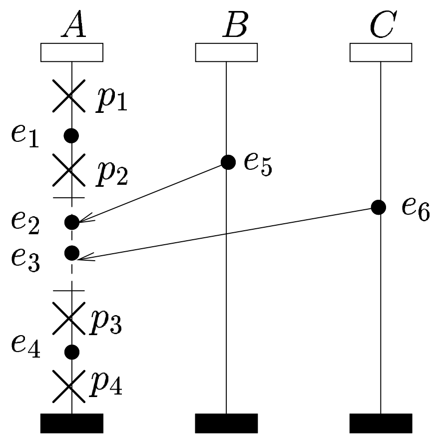

Figure 4: Cutting points

Definition 5 *A* cut *of a bMSC M $=< E, \leq, A, I, \alpha >$ is a subset C of E such that $C = \bigcup_{i \in I} cp_i$ is the union the cutting points associated to each instance.*

A cut can be represented graphically by a line going through the points defined by cutting points. Figure 5 shows three possible cuts C_1, C_2 and C_3. All cuts are not pertinent, ie they do not provide information about how a bMSC can be split.

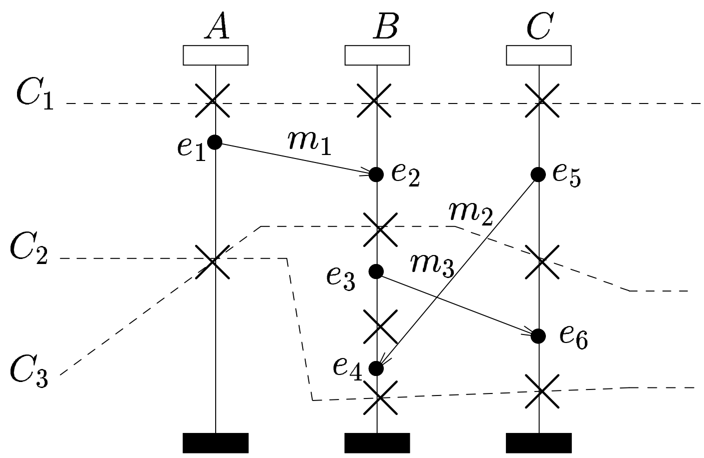

Figure 5: Some cuts

Definition 6 *A cut partitions a bMSC M into two partially ordered sets $M_1 = (C, \leq_{|C})$ and $M_2 = (E \backslash C, \leq_{|E \backslash C})$. A cut C will be called a* valid cut *if M_1 and M_2 are sub-bMSCs of M.*

A valid cut of a bMSC M is a cut that allows to partition M without cutting any message. In example Figure 5, C_3 is not a valid cut, as message m_2 is cut off. Note that our definition of cutting points also prevents coregions from being partitionned. This definition of valid cuts is similar to the definition of distributed control points without Z-Path of [4]. The main difference is that a Z-path free observation point ensures that a reception can appear in an observation if and only if the emission also appears in the same observation point, which allows to save observations in which a message is transitting. In the case of valid cuts, we do not want to separate message emission and reception.

Definition 7 *A* basic pattern *of a bMSC M is a subset $B \subseteq E$ such that any cut C of M with $B \cap C \neq \emptyset$ and $B \cap (E \backslash C) \neq \emptyset$, is not a valid cut.*

A basic patern is a set of events such that any cut separating them is invalid. bMSC in Figure 3 is a basic pattern, as it is impossible to separate its events without cutting a message. A decomposition of a bMSC is a partition into basic patterns, ie into elementary patterns that can not be cut without cutting a message.

Theorem 1 *Let $M =< E, \leq, A, I, \alpha >$ be a bMSC, and $X \subseteq E$ be a subset of events of E, let $\mathcal{G}(M)$ be the covering graph of the quasi-order generated by:*

$$\leq \cup \{(e, e'), e' \xrightarrow{m} e\} \cup \{(e, e') \cup (e', e),\ e \text{ and } e' \text{ in the same coregion}\}$$

then X is a basic pattern of M if and only if X is a strongly connected component of $\mathcal{G}(M)$.

<u>Proof:</u>

- X not basic pattern of $M \Longrightarrow X$ not strongly connected component of $\mathcal{G}(M)$:

  As we know that $\mathcal{G}(M)$ contains an reverse edge between two events connected by a message, then no message is cut in X. Similarly, no coregion is cut. Let us suppose that X is not a basic pattern of M. This would mean that there exists a valid cut C of M such that X can be partitioned into two non empty sets X_1 and X_2 (see illustration in Figure 6). Edges connecting X_1 and X_2 are edges from causality on instances as a valid cut do not split messages. If C partitions X into X_1 and X_2, then $X_1 = X \cap C$ and $X_2 = X \cap (E \backslash C)$. So, any edge between X_1 and X_2 goes from X_1 to X_2. Consequently, X is not a strongly connected component of $\mathcal{G}(M)$.

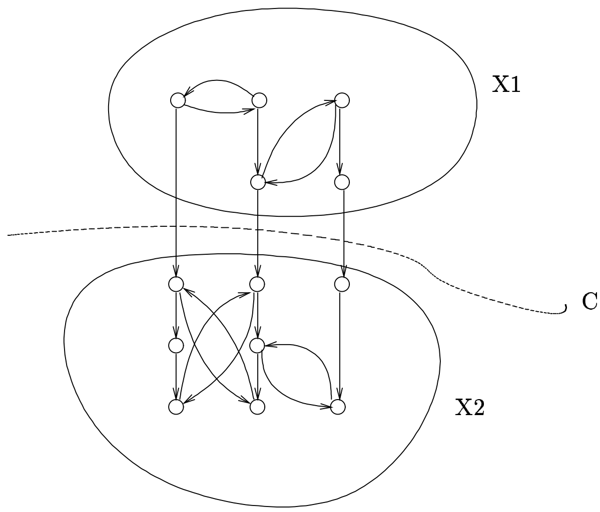

Figure 6: partition of X by the cut C

- X not strongly connected component of $\mathcal{G}(M) \Longrightarrow X$ is not a basic pattern of $\mathcal{G}(M)$:

  Let us suppose that a set $X \subseteq \mathcal{G}(M)$ of events is not strongly connected. Then, there exists a pair of events (e, s) such that no cycle of X goes through e and s. Consequently, the strongly connected component containing e (noted $scc(e)$) do not contain s, therefore $scc(e) \neq X$. Furthermore, as $e \in scc(e)$, $scc(e) \neq \emptyset$. Let $C = \downarrow scc(e)$. Then, X can be partitioned into $\{C, X \backslash C\}$. If X is a bMSC, then C is a valid cut: any edge $x \longrightarrow y$ from C to $X \backslash C$ can not be a message or coregion edge (else y would be contained in C). So, X can not be a basic pattern.□

4 Decomposition of bMSCs into irreducible sub-bMSCs

When considering basic patterns within an bMSC, it seems obvious that independent patterns can be placed in any order. Even more, a bMSC can be considered as layers of parallel basic patterns.

Definition 8 *A bMSC M is said* reducible *if there exist sub-bMSCs $M_1 \neq \emptyset$ and $M_2 \neq \emptyset$ of M with $M_1 \not\parallel M_2$, such that $M = M_1; M_2$. A bMSC that is not reducible will be said to be* irreducible.

Let us note that if there exists a valid cut C partitioning M in M_1 et M_2, then we have $M = M_1; M_2$ and the bMSC M is reducible if $M_1 \not\parallel M_2$. Therefore, a basic pattern is necessarily irreducible. Furthermore, a parallel composition of irreducible bMSCs is irreducible. Conversely, if $M = M_1 || M_2$ is an irreducible bMSCs, then so are M_1 et M_2. This suffices to characterize irreducible bMSCs as follows.

Proposition 1 *A bMSC M is irreducible if and only if M is a parallel composition of basic patterns: $M = b_1 || b_2 || \cdots || b_N$.*

A decomposition of a bMSC into irreducible elements is unique, up to permutations between parallel compositions.

Proposition 2 *The* irreducible form *of a bMSC M is the HMSC $H = N_1; N_2; \ldots; N_n$ equivalent to M (the local concatenation of $N_1, \ldots, N_n$ defines the same order as M), and such that:*

i) $\forall i, 1 \leq i \leq n$, N_i is an irreducible bMSC,

ii) for all basic patterns b of N_j, where $2 \leq j \leq n$, there exists a basic pattern, $\overline{b}$, of N_{j-1} such that $b \not\parallel \overline{b}$.

Proof:
It can be easily shown that the irreducible form of M is unique. To conclude the proof of proposition 2, we need the following lemma, which proof is immediate:

Lemma 1 *Let SC be the set of all strongly connected components of the graph $\mathcal{G}(M)$ and $\preceq$ be the relation defined by:*

$$X \preceq X' \Leftrightarrow \exists e \in X, \exists e' \in X', e \leq e'$$

Then $\preceq$ is a partial order relation on SC.

This lemma implies that if $X \prec X'$ then the basic pattern X' is not parallel to X. Conversely, X is parallel to X' if and only if X et X' are incomparable in $\leq$.

Now, let us consider the following algorithm which constructs a HMSC H from M.

Algorithm:

Build the graph $\mathcal{G}(M)$
Compute the strongly connected components of $\mathcal{G}(M)$: $SC = X_1, .., X_k$
$H := \varnothing$
while $SC \neq \varnothing$ **do**
  $N := \varnothing$
  for all $X \in min(SC)$ **do**
    $N := N || X$
  end for
  $SC := SC \backslash min(SC)$
  $H := H; N$
end while

The computation of the strongly connected components can be performed using the well known algorithm of Tarjan ([8]). As $|SC|$ is finite, this algorithm terminates. One has to show that H verifies the conditions of proposition 2.

According to the construction method, any component N_i in $H = N_1; \ldots ; N_n$ is an irreducible bMSC. Furthermore, as any basic pattern X of N_i, with $2 \leq i \leq n$, is not minimal in SC for the order $\preceq$, there exists $X' \in SC$ such that X' covers X. According to lemma 1, H satisfies the second condition of the proposition. Now, the last thing to prove is that the behaviours of H and M are the same: if $e \prec\!\!\!\cdot_M e'$ (e covers e' for the order of M) then, in the case where e and e' are in the same strongly connected component, there exists i such that $e, e' \in N_i$. As N_i is a sub-bMSC of M, we obtain $e \prec\!\!\!\cdot_H e'$. In the case where $e \in X$ and $e' \in X'$ with $X \neq X'$ and X, X' are two strongly connected components; We necessarily have $\phi(e) = \phi(e')$ and the weak sequential composition implies that $e \prec\!\!\!\cdot_H e'$. The converse is similar.$\square$

Example:
Consider bMSC in Figure 7. The irreducible form of this bMSC is shown in Figure 8.

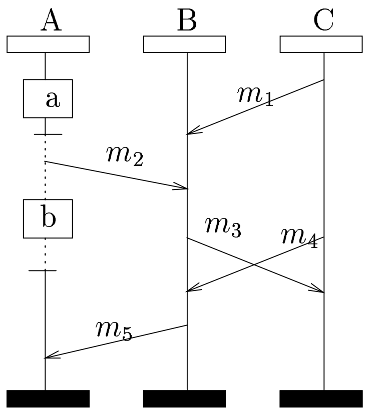

Figure 7: A reducible bMSC

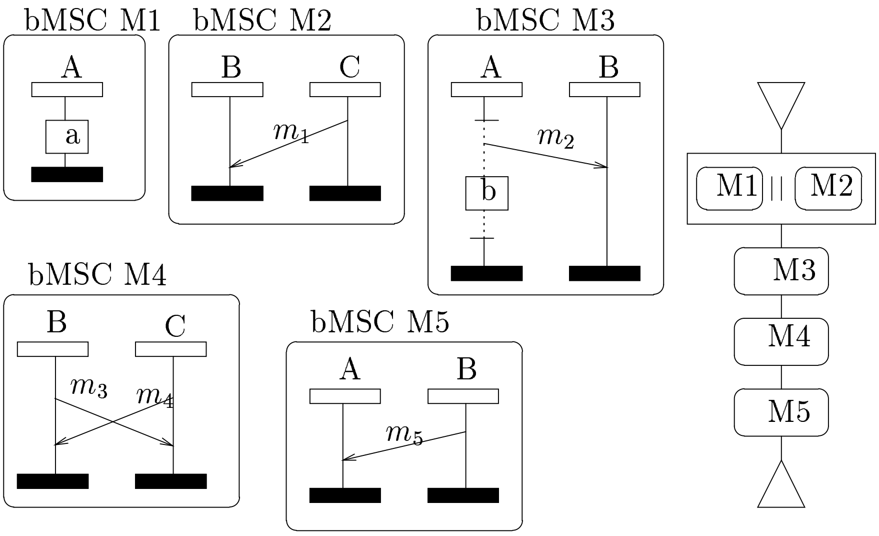

Figure 8: Irreducible form of bMSCs Figure 7

5 Irreducible form for HMSCs

We now want to define an irreducible form for a HMSC. We want a HMSC to define a composition of irreducible bMSCs, and choices to be choices between different basic patterns. Common basic patterns are factorized. Furthermore, in order to detect concurrency between patterns, basic patterns are moved as upward as possible.

Definition 9 *Let $H =< \mathcal{N}, \rightarrow, q_{init}, Q_{end}, \mathcal{M} >$ be a HMSC, let $M_0 = N||N'$ a bMSC of H, where N and N' are bMSCs and $q_0, q_1 \in \mathcal{N}$ such that $q_0 \overset{M_0}{\rightarrow} q_1$; we will say that we can lift up the bMSC N if q_0 is not a choice node, $q_0 \neq q_{init}$, and one of the following conditions holds:*

i) there exists only one predecessor q of q_0 such that $q \xrightarrow{M} q_0$ and $I(N) \cap I(M) = \varnothing$

ii) there exists more than one predecessor of q_0, there is no cycle going through q_0, and for all $q_i \xrightarrow{M_i} q_0$, $i > 1$, $I(N) \cap I(M_i) = \varnothing$

As q_0 is not a choice node, then any cycle going through q_0 goes through at least one of the q_i's. The conditions ii) prevents from copying pattern outside loops, hence doing unfoldings, which may prevent the irreducible form algorithm of section 5 from terminating.

If we can lift up the parallel component N; we define a new HMSC $H' =< \mathcal{N}', \rightarrow, \mathcal{M}' >$ with $\mathcal{N}' = \mathcal{N}$, $M_0 \in \mathcal{M}$ is replaced by N', the bMSCs M are replaced by $M \parallel N$, the relation $\rightarrow$ is unchanged and finally we change $q_0 \xrightarrow{M_0} q_1$ into $q_0 \xrightarrow{N'} q_1$ and $q \xrightarrow{M} q_0$ into $q \xrightarrow{M \| N} q_0$.

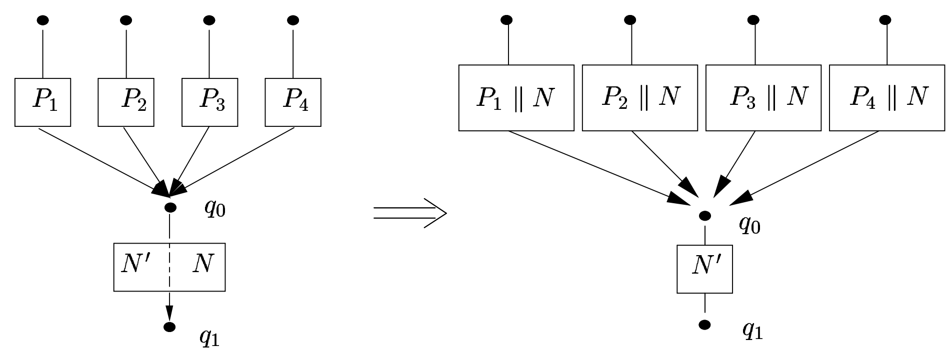

Figure 9: The lift up of a bMSC

Let r be the transformation witch lifts up all the bMSCs of H. This transformation is well defined, as does not depend on the order in which irreducible components are lift up. It is straightforward that the image of H by r has the same behaviour as H. The graph of H has a finite depth, so the iteration, r^n, of the transformation r are constant for n large enough. The limit transformation is noted R, the HMSC $R(H)$ has the property that no bMSC can be lift up, moreover his behaviours are equivalent to those of H.

Definition 10 *Let $H =< \mathcal{N}, \rightarrow, q_{init}, Q_{end}, \mathcal{M} >$ be a HMSC. Let $q \in \mathcal{N}$ be a node such that there exists a set of bMSCs $M_1, \ldots, M_k, k > 1$, defined by $q \xrightarrow{M_i} q_i$. If all M_i's have a common parallel component N, then we will say that N can be* factorized *if q and all q_i's are in the same strongly connected component of H, or if there is no cycle containing q and one of the q_i's.*

If a common bMSC N can be factorized, then all its parallel sub-components can also be factorized. Among all the components that can be factorized, we can always find the maximal component with respect to bMSC inclusion. So, we can define for this component an operation called factorization of N, which constructs a new HMSC H' from H the following way:

From HMSC H, we define a new node q' and a new transition $q \xrightarrow{N} q'$. On each branch of the choice, the bMSC M_i is equivalent to $M'_i \parallel N$ (M'_i can be an empty bMSC). Then, the following cases may occur for each branch of the choice originally starting from q:
<u>Case 1:</u> If $M'_i = \emptyset$, then the transition $q \xrightarrow{M_i} q_i$ is suppressed, and q_i and q' are identified.
<u>Case 2:</u> if $M'_i \neq \emptyset$, then the transition $q \xrightarrow{M_i} q_i$ is removed, and a transition $q' \xrightarrow{M'_i} q_i$ is added.

The factorization procedure is described on Figure 10. In this exemple, nodes q_1 and q_2 are assumed to belong to the same strongly connected component as q.

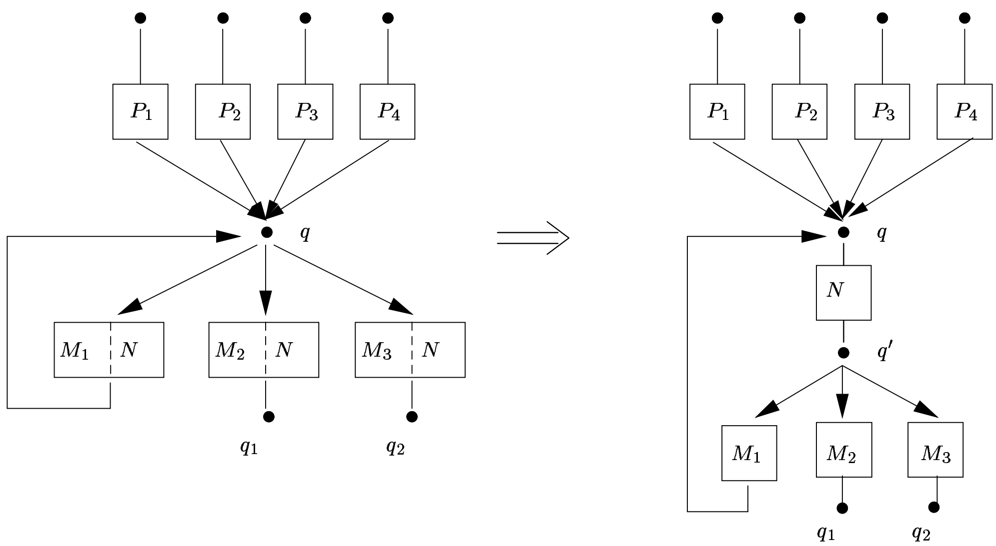

Figure 10: Factorization of bMSC N

We define a transformation F for a HMSC which replaces H by the HMSC H' consisting of the factorization of the maximal common parallel component on each choice of H. The result of this transformation can not be factorized.

Definition 11 *A HMSC* $H = < \mathcal{N}, \rightarrow, q_{init}, Q_{end}, \mathcal{M} >$ *is said to be in* irreducible form *if:*

i) *All the bMSCs of* $\mathcal{M}$ *are irreducible*

ii) *For all choice node* $q \in \mathcal{N}$ *between branches* $q \overset{M_i}{\rightarrow} q_i$ *, the bMSCs* M_i*'s have no common parallel component that can be factorized,*

iii) *For all bMSC* $M = b_1 || \cdots || b_N$ *and for all* $1 \leq i \leq N$*, the bMSC* b_i *can not be lift up.*

We can define an algorithm for computing a HMSC in irreduible form from an initial HMSC H_0. For this, we will need the refinement procedure defined as follows:

Definition 12 *Let* $H = < \mathcal{N}, \longrightarrow, q_{init}, Q_{end} \mathcal{M} >$ *be a HMSC, and* $H_M = < \mathcal{N}_M, \longrightarrow_M, q_{init_M}, Q_{end_M}, \mathcal{M}_M >$ *be the irreducible form of a bMSC* M*. Note that as* H_M *is a sequence,* Q_{end_M} *is reduced to an unique node* q_{end_M}*. The* refinement *of* H *by* H_M *is the HMSC* $H' = < \mathcal{N}', \rightarrow', q_{init}, Q_{end}, \mathcal{M}' >$ *obtained by replacing each transition* $q \xrightarrow{M} q'$ *in* H *by* H_M*:*

- $\mathcal{N}' = \mathcal{N} \backslash \{q, q'\} \cup \mathcal{N}_M$
- $$\longrightarrow' = \begin{array}{l} (\longrightarrow \backslash \{q \xrightarrow{M} q'\}) \bigcup \longrightarrow_M \\ \bigcup \{x \xrightarrow{X} q_{init_M} \textit{ such that } x \xrightarrow{X} q \in \longrightarrow\} \\ \bigcup \{q_{end_M} \xrightarrow{Y} y \textit{ such that } q' \xrightarrow{Y} y \in \longrightarrow\} \end{array}$$
- $\mathcal{M}' = \mathcal{M} \backslash \{M\} \cup \mathcal{M}_M$

Algorithm:

$H_0 = < \mathcal{N}_0, \rightarrow_0, \mathcal{M}_0 >$
$H'' := H_0$
for all bMSC $M \in \mathcal{M}$ **do**
  compute H_M
  $H'' := H''$ refined by H_M
end for
repeat
  $H := H''$
  Compute $H' = R(H)$ `/* parallel components of irreducible bMSCs of` H `are lift up as high as possible */`
  Compute $H'' = F(H')$ `/* for all choices, common parallel components are factorized */`
until $H'' \neq H$ `/* loop if` H `has been modified */`
$H^{irr} := H$

Proposition 3 *This algorithm terminates, and* H^{irr} *is in irreducible form. The HMSC* H^{irr} *will be called the* irreducible form *of* H_0*.*

Proof:

First, let us show that if, for a HMSC H, $F \circ R(H) = H$, then H is in irreducible form. We have seen that for a HMSC in which all bMSCs are irreducible, then the transformation of H by R or F do not contain reducible bMSCs. So, $F(R(H))$ satisfies $i)$ in the definition of an irreducible form. Furthermore, the HMSC $F(R(H))$ can not be factorized, so $ii)$ is also verified. It remains to verify $iii)$.

If there exists an irreducible component N that can be lift up in $F \circ R(H)$, then as $R(H)$ is stable for R, the component N comes from a factorization. Consequently, there exists in $R(H)$ a choice node q such that for all branches $q \xrightarrow{M_i} q'_i$ originating from q, $M_i = N \parallel M'_i$. Now, as $R(F(R(H)) = R(H)$, the transformation $R \circ F$ applied to $R(H)$ will give the same configuration at node q. Let $Conf$ be the set of choice nodes in $R(H)$ where N can be factorized. Let Ω be the set of strongly connected components associated to each node of $Conf$. It is obvious that Ω is ordered. Let $c \in \max \Omega$, we note K the number of choice nodes in c, and n_q the number of branches in a choice $q \in c$. The restriction of $R(H)$ to c contains $\Sigma_{q \in \Omega} n_q$ occurrences of the irreducible bMSCs N, and the restriction of $F(R(H))$ to c contains K occurrences of N. As $|\Omega| > 1$, and $\forall q \in \Omega, n_q > 1$, we have $\Sigma_{q \in \Omega} n_q > K$. As c is maximal in $R(H)$ for the factorization of N, no new occurrence of N can be added from $H \backslash c$ in c by the transformation F followed by R. So, from the equation $R(F(R(H)) = R(H)$, and as $\Sigma_{q \in \Omega} n_q > K$, at least one bMSC N must be duplicated by a lift up operation. So, there exists in $F(R(H))$ restricted to c a node q with at least two predecessors q_1 and q_2 allowing for the lift up of N. As c is strongly connected, there is at least on cycle going through q, which prevents from lifting up N.

Now, let us prove that the algorithm terminates. If the initial HMSC is a tree, then it is obvious that the algorithm terminates after a finite number of steps. Now, let us consider the case when H consist on only one single strongly connected component. We can show by induction on the number of elementary cycles in H that the algorithm terminates. If H is an elementary cycle, then it contains the initial node, and no choice node. Therefore, the factorization F has no effect on both H and $R(H)$. By definition, R is a limit transformation and so, $R \circ R(H) = R(H)$. Consequently, for all $k > 1$, we have $(F \circ R)^k(H) = R^k(H) = R(H)$. Suppose H has $n+1$ elementary cycles, $\sigma_0, \ldots, \sigma_n$, and σ_0 contains the initial node of H. We define a new HMSC H' constituted of the cycles $\sigma_1, \ldots, \sigma_n$. There exist a transition $q \longrightarrow q'$ of σ_0 such that $q' \in H'$ and $q \notin H'$. Let us consider q' as the new initial node for H'. By hypothesis of induction, there is a $k \in \mathbb{N}$ such that $\forall n > 1, (F \circ R)^{k+n}(H') = (F \circ R)^k(H')$. Moreover, $(F \circ R)^n \sigma_0$ is also constant for n large enough. And then, thanks to the decomposition of H into σ_0 and H', $(F \circ R)^n(H)$ also converges.

We can see a HMSC as a tree of strongly connected components. By induction on the number of strongly connected components, and from the previous result on the convergence of the algorithm for a tree, it can be easily shown that the iteration of the transformation converges. $\square$

6 Conclusion

This article has defined an irreducible form for High-level Message Sequence Charts. This transformation can be useful for analysis procedures considering bMSCs as the granularity of the study, such as the temporal study of [3]. It can also help showing structural equivalence of specifications.

References

[1] J.-P. Katoen, L. Lambert. Pomsets for Message Sequence Charts. 1st Workshop of the SDL Forum Society on SDL and MSC, SAM98, Berlin, June/July 1998.

[2] L.Helouet, C.Jard, B.Caillaud An effective equivalence for sets of scenarios represented by Message Sequence Charts. INRIA research report no 3499, ftp://ftp.inria.fr/INRIA/publication/RR/RR-3499.ps.gz

[3] P.Le Maigat, L.Helouet, A (max,+) approach for time in Message Sequence Charts, 5th workshop on Discrete Event Systems (WODES 2000), Ghent, Belgium, August 2000.

[4] R.H.B Netzer, J.Xu, Necessary and sufficient conditions for Consistent Global Snapshots, IEEE Transactions on Parallel and distributed Systems, 6(2):165-169,1995.

[5] V.Pratt Modeling Concurrency with Partial Orders, International Journal of Parallel Programming, Vol 15, No 1, 1986, pp 33-71

[6] E. Rudolph, J. Grabowski, P. Graubmann. Tutorial on Message Sequence Charts (M-SC'96). In: Tutorials of the First joint International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols, and Protocol Specification, Testing, and Verification(FORTE/PSTV'96), Kaiserslautern, Germany, October 1996. FORTE'96.

[7] F. Slomka, J. Zant, L. Lambert. MSC-based Schedulability Analysis. Workshop on Performance and Time in SDL and MSC. Technical Report 1/98, IMMD VII, University of Erlangen-Nuremberg, Erlangen, February 1998

[8] R. Tarjan, Depth-first search and linear graph algorithms, SIAM Journal of Computing, vol 1, no 2, 1992

[9] ITU, Norm Z.120, Message Sequence Charts.