Bounded MSC Communication*

Markus Lohrey and Anca Muscholl

LIAFA, Université Paris VII
2, place Jussieu, case 7014
75251 Paris cedex 05, France
{lohrey,muscholl}@liafa.jussieu.fr

Keywords: message sequence charts, verification, computational complexity

Abstract. Message sequence charts (MSCs) and different kinds of extensions like high-level message sequence charts (HMSCs) and nested message sequence charts (nMSCs) are popular formalisms for the specification of asynchronously communicating processes. An important concept in this context are the channel-buffers between communicating processes. Since real systems impose limitations on the capacity (or speed) of communication links, we consider the size of these buffers as a computational resource, similar to memory in classical computing devices. We introduce four different measures for buffer sizes and investigate for each of these measures the complexity of deciding whether a given MSC (HMSC, nMSC) satisfies a given bound on the buffer size. The complexity of these problems varies between the classes P, NP, and coNP.

1 Introduction

Message sequence charts (MSCs) and high-level message sequence charts (HMSCs) are popular visual formalisms for the specification of communication of asynchronous processes, where most of the details (variables, timing constraints, etc) are abstracted away. An important aspect for implementing such specifications is the size of the channel buffers used by the communicating processes. Since real systems impose limitations on the capacity (or speed) of communication links, we view the size of these buffers as a computational resource similar to memory in classical computing devices. In this paper we introduce four different measures for buffer sizes of (H)MSCs. These measures result from two orthogonal dimensions: In the first dimension we require that *all* linearizations of an MSC M have to satisfy a certain buffer bound ($\forall$-boundedness), respectively *at least one* linearization of M has to respect the bound ($\exists$-boundedness). In the second dimension we distinguish between measuring the buffer size as the number of undelivered messages over all channels (*global* boundedness), respectively as the maximum of the number of undelivered messages in a given communication channel, where the maximum is taken over all channels (*local* boundedness). Hence we say for instance that a MSC M is $\exists^b_{\mathrm{glob}}$-bounded (existentially, globally) for some $b \in \mathbb{N}$, if there exists a linearization of M such that for each prefix of this linearization the total number of undelivered messages is at most b. The notions of $\forall^b_{\mathrm{glob}}$-boundedness (universally, globally), $\exists^b_{\mathrm{loc}}$-boundedness (existentially, locally), and $\forall^b_{\mathrm{loc}}$-boundedness

* Research supported by the INRIA cooperative research action FISC.

(universally, locally) for MSCs are defined similarly. All these notions can be extended to HMSCs, by referring to all executions of an HMSC. We also consider the channel boundedness problem for nested MSCs (nMSCs). Nested MSCs are MSCs defined hierarchically, similar to straight-line expressions. We note that the notion of universal-local-boundedness corresponds to the notion of channel-boundedness used in [2, 3, 6, 7] in the context of regular MSC-languages. For each of our four measures we investigate the complexity of deciding whether a given (H)MSC or nMSC satisfies a given bound on the buffer size. The complexity of these problems varies between the classes P, NP, and coNP, see Table 1 in Section 7 for a summary of our results.

2 Preliminaries

We assume that the reader has some basic knowledge of computational complexity, in particular regarding the classes NL (nondeterministic logarithmic space), P (deterministic polynomial time), NP (nondeterministic polynomial time) and co-NP (complements of of NP-problems), see [9] for more details.

2.1 Message sequence charts

A *linearization* of a partial order $(A, <)$ is a linear order on A that extends the partial order $<$. It is written as a word over A, where every $a \in A$ occurs exactly once. A message sequence chart (MSC) M is a tuple $(\mathcal{E}, P, \lambda, t, m, <)$, where

- $\mathcal{E}$ is a (countable) set of *events*.
- P is a finite set of *processes*.
- $\lambda : \mathcal{E} \to P$ associates with each event e a process $\lambda(e)$ on which e occurs.
- $t : \mathcal{E} \to \{S, R\}$ associates with each event a type. Events in $t^{-1}(S)$ (resp. $t^{-1}(R)$) are called *send* (resp. *receive*) *events*.
- $m : t^{-1}(S) \to t^{-1}(R)$ is a bijection. A pair $(s, m(s))$ with s a send event is also called a *message* from process $p = \lambda(s)$ to process $q = \lambda(m(s))$. The *channel type* of s (resp. $m(s)$) is defined as $S(p, q)$ (resp. $R(p, q)$). The channel type of $e \in \mathcal{E}$ is denoted $\mathrm{ct}(e)$.
- $<$ is a partial order on $\mathcal{E}$, also called the *visual order of* M, such that for each process $p \in P$ the restriction $<|_{\lambda^{-1}(p)}$ of $<$ to the events on p is a linear order (either finite or of type ω) and furthermore $<$ is equal to the transitive closure of

$$\bigcup_{p \in P} <|_{\lambda^{-1}(p)} \ \cup \ \{(s, m(s)) \mid s \in t^{-1}(S)\}.$$

Often MSCs are further restricted to satisfy the FIFO-condition, which means that whenever there are two send events s_1 and s_2 with $\mathrm{ct}(s_1) = \mathrm{ct}(s_2)$ and $s_1 < s_2$ then also $m(s_1) < m(s_2)$, i.e, message overtaking on any channel is disallowed. A channel is a pair (p, q) of distinct processes. The MSC definition may also take message names into account. The complexity results in this paper mostly hold independently of the FIFO-restriction (respectively whether message names are allowed or not). This is due to the fact that all lower bound proofs in this paper hold under the

FIFO-restriction, whereas all upper bound proofs (excepting those for nMSCs) hold without the FIFO-restriction.

Let $M = (\mathcal{E}, P, \lambda, t, m, <)$ be an MSC. A *linearization* of M is a linearization of the visual order $(\mathcal{E}, <)$. Let L be a linearization of M and let $b \in \mathbb{N}$. We say that L is *globally-bounded* by b if $|K|_{t^{-1}(S)} - |K|_{t^{-1}(R)} \leq b$ for every prefix K of L. We say that L is *locally-bounded* by b if for all channels (p,q) and every prefix K of L it holds $|K|_{\mathrm{ct}^{-1}(S(p,q))} - |K|_{\mathrm{ct}^{-1}(R(p,q))} \leq b$. We say that M is $\exists^b_{\mathrm{glob}}$-bounded (resp. $\exists^b_{\mathrm{loc}}$-bounded) if there exists a linearization of M which is globally-bounded (resp. locally-bounded) by b. We say that M is $\forall^b_{\mathrm{glob}}$-bounded (resp. $\forall^b_{\mathrm{loc}}$-bounded) if every linearization of M is globally-bounded (resp. locally-bounded) by b. Of course for $Q \in \{\exists, \forall\}$, if M is Q^b_{glob}-bounded, then M is also Q^b_{loc}-bounded. Vice versa, if M is Q^b_{loc}-bounded, then M is $Q^{|P|^2 \cdot b}_{\mathrm{glob}}$-bounded.

Instead of speaking about prefixes of linearizations of MSCs, it is sometimes more convenient to consider configurations of MSCs. A *configuration* C of M is a downward-closed subset $C \subseteq \mathcal{E}$, i.e., if $e < f \in C$ then also $e \in C$. A prefix of a linearization of M defines in the obvious way a unique configuration of M. Vice versa, for every configuration C of M there exists at least one prefix K of a linearization of M such that K defines C. Let C be a configuration of M. The number of messages (s,r) in M with $s \in C$ and $r \notin C$ is denote by $\mathrm{gus}(C,M)$ (**g**lobally **u**nmatched **s**ends). The maximum over all channels (p,q) of the number of messages (s,r) in M with $\mathrm{ct}(s) = S(p,q)$, $\mathrm{ct}(r) = R(p,q)$, $s \in C$, and $r \notin C$ is denote by $\mathrm{lus}(C,M)$ (**l**ocally **u**nmatched **s**ends). Note that M is $\forall^b_{\mathrm{glob}}$-bounded (resp. $\forall^b_{\mathrm{loc}}$-bounded) if and only if for every configuration C of M it holds $\mathrm{gus}(C,M) \leq b$ (resp. $\mathrm{lus}(C,M) \leq b$).

Let $M_1 = (\mathcal{E}_1, P, \lambda_1, t_1, m_1, <_1)$ and $M_2 = (\mathcal{E}_2, P, \lambda_2, t_2, m_2, <_2)$ be two MSCs over the same set P of processes, where furthermore $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$ and M_1 is finite. Then the concatenation of M_1 and M_2 is the MSC $M_1 M_2 = (\mathcal{E}_1 \cup \mathcal{E}_2,\ P,\ \lambda_1 \cup \lambda_2,\ t_1 \cup t_2,\ m_1 \cup m_2,\ <)$, where $<$ is the transitive closure of

$$<_1 \cup <_2 \cup \{(e_1, e_2) \mid e_1 \in \mathcal{E}_1, e_2 \in \mathcal{E}_2, \exists p \in P : \lambda_1(e_1) = \lambda_2(e_2)\}.$$

An HMSC is a tuple $H = (V, \rightarrow, P, \mu, v)$, where V and P are finite sets, $\rightarrow \subseteq V \times V$, μ maps every node $u \in V$ to a finite MSC $\mu(u)$ over the set of processes P, and $v \in V$ is the initial node. The MSC-language $L(H)$ defined by H is the set of all MSCs $\mu(u_1)\mu(u_2)\mu(u_3)\cdots$, where $u_1 = v$ and $u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow \cdots$ is a (finite or infinite) maximal path in $(V, \rightarrow)$. We impose the restriction that for every node $u \in V$ there exists a path from v to u in $(V, \rightarrow)$. Of course this restriction can always be ensured and it does not change the defined MSC-language. We say that an HMSC H is $\exists^b_{\mathrm{glob}}$-bounded if all $M \in L(H)$ are $\exists^b_{\mathrm{glob}}$-bounded. The notions of $\{\exists^b_{\mathrm{loc}}, \forall^b_{\mathrm{glob}}, \forall^b_{\mathrm{loc}}\}$-boundedness for HMSCs are defined analogously.

2.2 Pebble games

As we will see in Section 3 there is a tight connection between the existential-global boundedness problem and pebble games on directed graphs. In this section we recall the notion of pebble game and state the results used for the boundedness problem.

Let $G = (V, E)$ be a finite *directed acyclic graph* (dag) with node set V and edges $E \subseteq V \times V$. For two game-configurations $C_1, C_2 \subseteq V$ and a node $v \in V$ we

write $C_2 = C_1 \dot{\cup} \{v\}$ whenever $C_2 = C_1 \cup \{v\}$ and $v \notin C_1$, i.e., C_2 is the disjoint union of C_1 and v. A *move* in G is a pair (C_1, C_2) of game-configurations such that one of the following three cases holds:

(1) There exists a node $w \in C_1$ with $C_2 = C_1 \backslash \{w\}$.
(2) There exists a node $v \in C_2$ such that $C_2 = C_1 \dot{\cup} \{v\}$ and for all $u \in V$ with $(u, v) \in E$ it holds $u \in C_1$.
(3) There exist nodes $w \in C_1$, $v \in C_2$ such that $(w, v) \in E$, $C_2 = (C_1 \backslash \{w\}) \dot{\cup} \{v\}$, and for all $u \in V$ with $(u, v) \in E$ it holds $u \in C_1$.

More precisely we say that (C_1, C_2) is an i-move, $i \in \{1, 2, 3\}$, if case (i) in the enumeration above holds. Let $b \in \mathbb{N}$. We say that the graph G can be *b-pebbled* if there exists a sequence $C_1, C_2 \ldots, C_n \subseteq V$ of game-configurations such that the following holds:

- $C_1 = C_n = \emptyset$
- $|C_i| \leq b$ for $1 \leq i \leq n$
- (C_i, C_{i+1}) is a move for $1 \leq i < n$.
- For every node $v \in V$ there exists exactly one $i \in \{1, \ldots, n-1\}$ such that $v \notin C_i$ and $v \in C_{i+1}$.

We say that the graph G can be *b-pebbled without the move rule* if there exists a sequence $C_1, C_2 \ldots, C_n \subseteq V$ such that the following holds

- $C_1 = C_n = \emptyset$
- $|C_i| \leq b$ for $1 \leq i \leq n$
- (C_i, C_{i+1}) is a 1-move or a 2-move for $1 \leq i < n$.
- For every node $v \in V$ there exists exactly one $i \in \{1, \ldots, n-1\}$ such that $v \notin C_i$ and $v \in C_{i+1}$.

In [10] the following theorem was shown.

Theorem 1. *The following problem is NP-complete:*
INPUT: Finite dag G with exactly one node of outdegree 0, positive integer b.
QUESTION: Can G be b-pebbled?

From this theorem we can easily deduce the following corollary. For the proof we refer to the appendix.

Corollary 1. *The following problem is NP-complete:*
INPUT: Finite dag G with exactly one node of outdegree 0, positive integer b.
QUESTION: Can G be b-pebbled without the move rule?

Note that if a dag G contains more than one node of outdegree 0, then it is no longer the case that G can be b-pebbled if and only if G can be $(b+1)$-pebbled without the move rule. A counterexample for this would be the dag with veretx set $(\{a, b, c, d\}$ and edges (a, c), (a, d), (b, c), (b, d).

For a finite dag (V, E) let $\overline{V} = \{\overline{v} \mid v \in V\}$ be a disjoint copy of the node set V and let $\tilde{E} = \{(u, v), (v, \overline{u}) \mid (u, v) \in E\} \cup \{(u, \overline{u}) \mid u \in V\}$. Note that the graph $(V \cup \overline{V}, \tilde{E})$ is again a finite dag. The following lemma is easy to prove.

Lemma 1. *A finite dag (V, E) can be b-pebbled without the move rule if and only if there exists a linearization ℓ of the partial order $(V \cup \overline{V}, \tilde{E}^+)$ such that for every prefix k of ℓ we have $|k|_V - |k|_{\overline{V}} \leq b$.*

3 Bounded communication in finite MSCs

For the local channel boundedness question we can argue by considering some additional ordering on events. Let us fix a bound b and an MSC $M = (\mathcal{E}, P, \lambda, t, m, <)$. We define a binary relation $\rightsquigarrow$ on $\mathcal{E}$ as follows. We let $r \rightsquigarrow s$ whenever for some channel (p, q) and some $i \geq 1$ we have that s is the $(i+b)$-th send of channel type $S(p, q)$, whereas r is the i-th receive of channel type $R(p, q)$.

Lemma 2. *An MSC is $\exists^b_{\mathrm{loc}}$-bounded if and only if the relation $< \cup \rightsquigarrow$ is acyclic, where $<$ is the visual order of M and $\rightsquigarrow$ is the relation associated with M, b.*

Proof. Assume that the transitive closure of $< \cup \rightsquigarrow$ is acyclic. Then clearly any linearization of $< \cup \rightsquigarrow$ is a linearization of M that is locally-bounded by b. Conversely, if L is a linearization of M that is locally-bounded by b, then every prefix K of L is such that $|K|_{\mathrm{ct}^{-1}(S(p,q))} - |K|_{\mathrm{ct}^{-1}(R(p,q))} \leq b$. Hence, the i-th receive of channel type $R(p, q)$ precedes the $(i+b)$-th send of channel type $S(p, q)$, for each i. Thus, L is a linearization of $< \cup \rightsquigarrow$. □

Since it can be checked in linear time, whether a directed graph is acyclic, the previous lemma immediately yields the following result:

Proposition 1. *The following problem can be solved in linear time:*
INPUT: Finite MSC M and positive integer b.
QUESTION: Is M $\exists^b_{\mathrm{loc}}$-bounded?

Proof. Note that the size of the relation $\rightsquigarrow$ is linear in the size of M. □

Surprisingly, if we consider the *global* boundedness instead of the local one, the existential variant of the problem becomes more difficult, more exactly it is NP-complete:

Theorem 2. *The following problem is NP-complete:*
INPUT: Finite MSC M and positive integer b.
QUESTION: Is M $\exists^b_{\mathrm{glob}}$-bounded?

Proof. Membership in NP is obvious. In order to prove NP-hardness, we will construct for a finite dag $G = (V, E)$ a finite MSC $M(G) = (\mathcal{E}, P, \lambda, t, m, <)$ such that G can be b-pebbled without the move rule if and only if $M(G)$ is $\exists^{(b+1)}_{\mathrm{glob}}$-bounded. With each node $v \in V$ we associate the set of processes $P_v = \{v_{\mathrm{in}}, v_{\mathrm{out}}\} \cup \{(v, w) \mid (v, w) \in E\}$, the set of all processes is then $P = \bigcup_{v \in V} P_v$. The set $\mathcal{E}$ of events consists of $V \cup \overline{V}$ plus some additional events, which need not be specified for the further consideration. We have $\lambda(v) = v_{\mathrm{in}}$ and $\lambda(\overline{v}) = v_{\mathrm{out}}$. For each node v there is a message $(v, \overline{v})$ from process v_{in} to process v_{out}. Furthermore for each edge $(u, v) \in E$ we have exactly one message from process (u, v) to v_{in} and back from v_{in} to (u, v). Finally if $(v, w_1), \ldots, (v, w_n)$ are all outgoing edges of node v (listed in an arbitrary order) then there exists exactly one message from process (v, w_i) to (v, w_{i+1}) and back $(1 \leq i \leq n-1)$ and exactly one message from process v_{in} to (v, w_1) and from process (v, w_1) to v_{out}. The order of the events on the processes in P_v is shown in Figure 1, where $(u_1, v), (u_2, v), (v, w_1), (v, w_2)$, and (v, w_3) are all

edges that are adjacent to the node v. The process names labeling the message arrows specify the source and target processes of the corresponding messages. Note that $<$ is indeed acyclic, for instance the sends of the messages from (u,v) to v_{in} must precede the sends of the messages from (v,w) to w_{in}, with $(u,v),(v,w)$ edges of the dag. Moreover, it is easy to check that $M(G)$ respects the FIFO-restriction. The crucial point of our construction is that the restriction $<|_{V\cup\overline{V}}$ of the visual order $<$ of $M(G)$ to the set $V \cup \overline{V} \subseteq \mathcal{E}$ is precisely the transitive closure of the relation $\tilde{E}$ from Lemma 1.

Claim 1: If G can be b-pebbled without the move rule then $M(G)$ is $\exists^{(b+1)}_{\text{glob}}$-bounded. Assume that G can be b-pebbled without the move rule by a sequence of moves. We translate each move into a sequence of events, such that the resulting sequence of events is a linearization of M which is globally-bounded by $b+1$. Consider a move $(\mathcal{C}_1, \mathcal{C}_2)$. If $\mathcal{C}_2 = \mathcal{C}_1 \dot{\cup} \{v\}$, i.e., node v is pebbled in the move, then we execute the following sequence of events:

(1) Send and immediately receive the message from process (u_i, v) to v_{in} for $1 \leq i \leq k$, where $(u_1, v), \ldots, (u_k, v)$ are all incoming edges of node v.
(2) Execute send event v on process v_{in}.
(3) Send and immediately receive the message from process v_{in} to (u_k, v) for $1 \leq i \leq k$.
(4) Send and immediately receive the message from process v_{in} to (v, w_1).
(5) Send and immediately receive the message from process (v, w_i) to (v, w_{i+1}) for $1 \leq i < n$, where $(v, w_1), \ldots, (v, w_n)$ are all outgoing edges of node v.

On the other hand if $\mathcal{C}_2 = \mathcal{C}_1 \backslash \{v\}$, i.e., a pebble is taken from node v in the move, then we execute the following sequence of events:

(1) Send and immediately receive the message from process (v, w_{i+1}) to (v, w_i) for $1 \leq i < n$, where $(v, w_1), \ldots, (v, w_n)$ are all outgoing edges of node v.
(2) Send and immediately receive the message from process (v, w_1) to v_{out}.
(3) Execute the receive event $\overline{v}$ on process v_{out}.

Claim 2: If $M(G)$ is $\exists^{(b+1)}_{\text{glob}}$-bounded then G can be b-pebbled without the move rule.

Let L be a linearization of $M(G)$, which is globally-bounded by $b+1$, such that furthermore the number of prefixes K of L that satisfy $|K|_{t^{-1}(S)} - |K|_{t^{-1}(R)} = b+1$ is minimal among all linearizations of $M(G)$ that are globally-bounded by $b+1$. Clearly such an L exists. Let $\pi(L)$ be the projection of the word L onto the subalphabet $V \cup \overline{V} \subseteq \mathcal{E}$. Since of course $\pi(L)$ is a linearization of $<|_{V\cup\overline{V}}$, by Lemma 1 it suffices to prove the following claim:

Claim 2.1: For every prefix k of $\pi(L)$ it holds $|k|_V - |k|_{\overline{V}} \leq b$.

Clearly we have $|k|_V - |k|_{\overline{V}} \leq b+1$ for every prefix k of $\pi(L)$. In order to prove the claim let us assume that $|\pi(L_1 v)|_V - |\pi(L_1 v)|_{\overline{V}} = b+1$, where $v \in V$ and $L = L_1 v L_2$. Let $L_2 = e L_3$, where $e \in \mathcal{E}$ (note that we must have $L_2 \neq \epsilon$). If e would be a send event then $|L_1 v e|_{t^{-1}(S)} - |L_1 v e|_{t^{-1}(R)} \geq b+2$, a contradiction. Thus e must be a receive event. If $e \notin \overline{V}$ then we would have $|L_1 v|_{t^{-1}(S)} - |L_1 v|_{t^{-1}(R)} \geq b+2$ (note that already $|\pi(L_1 v)|_{t^{-1}(S)} - |\pi(L_1 v)|_{t^{-1}(R)} = b+1$). Thus $e = \overline{u}$ for some $u \in V$. We cannot have $v < \overline{u}$, since if $v < \overline{u}$ then by the construction of $M(G)$ there have to occur several events between v and $\overline{u}$. It follows that $L' = L_1 \overline{u} v L_3$ is

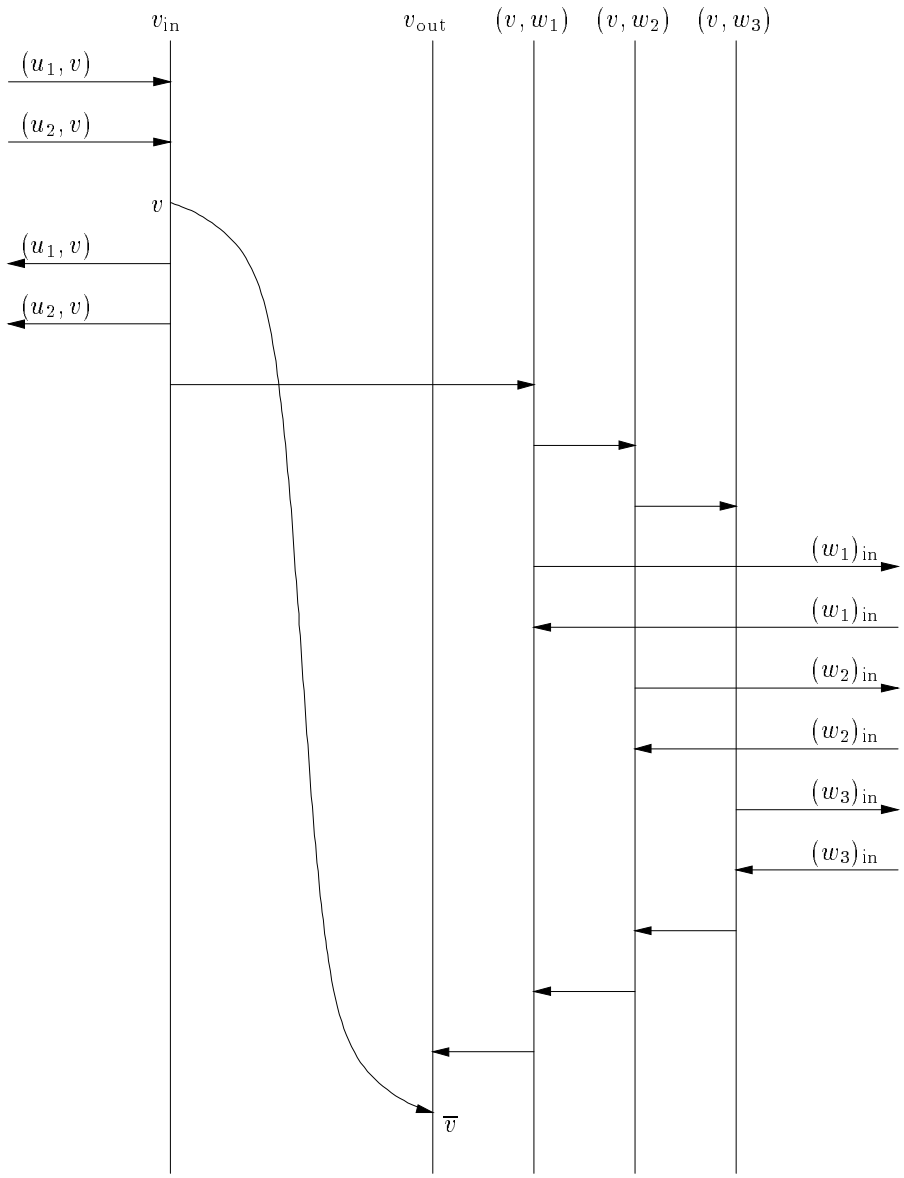


Fig. 1. Communication between the processes in P_v in the MSC $M(G)$

also a linearization of L that is globally-bounded by $b+1$. Since furthermore the number of prefixes K of L' such that $|K|_{t^{-1}(S)} - |K|_{t^{-1}(R)} = b+1$ is smaller then for L, we have a contradiction. This proves claim 2.1 and the theorem. □

If we consider universal-boundedness instead of the existential one, then by the following lemma we will obtain again a polynomial algorithm in the local setting of the problem.

Lemma 3. *An MSC is $\forall^b_{\mathrm{loc}}$-bounded if and only if the transitive closure of the relation $< \cup \rightsquigarrow$ is equal to the visual order $<$.*

Proof. Consider an MSC M and let s be the i-th send of channel type $\mathrm{ct}(s) = S(p,q)$, whereas r is the $(i+b)$-th receive of channel type $\mathrm{ct}(r) = R(p,q)$. Then M is $\forall^b_{\mathrm{loc}}$-bounded if and only if for all p, q, i, the receive r precedes the send s in every linearization of M, hence $r < s$ in the visual order. This shows the claim. □

Theorem 3. *The following problem can be solved in time $O(|M|^2)$:*
INPUT: Finite MSC M and positive integer b.
QUESTION: Is M $\forall^b_{\mathrm{loc}}$-bounded?

For the universal-global-boundedness we can also obtain a polynomial solution using flow theory:

Theorem 4. *The following problem can be solved in time $O(|M|^2 \log(|M|))$:*
INPUT: Finite MSC M and positive integer b.
QUESTION: Is M $\forall^b_{\mathrm{glob}}$-bounded?

Proof. In order to check universal-global-boundedness we consider the complementary problem, namely whether given a finite MSC M and $b \in \mathbb{N}$ there exists a configuration C of M such that $\mathrm{gus}(C, M) > b$. This question can be answered in polynomial time using the min-flow max-cut theorem, see e.g. [5]. More precisely we construct from M a dag as follows: View M as a dag, where the nodes are the events of M, and the edges are the messages of M plus pairs of events (e, f) such that e immediately precedes f on some process. To this dag we add two nodes σ and τ. We add an edge from σ to each minimal event of M, and similarly we add an edge from each maximal event of M to τ. Let us call the resulting dag $G(M)$. To each edge (v, w) of $G(M)$ we assign an *upper capacity* $c_{v,w}$ and a *lower capacity* $\ell_{v,w}$ as follows: All edges receive the upper capacity ∞. For all messages $(s, m(s))$ of M we let $\ell_{s,m(s)} = 1$, whereas for all other edges (v, w) of $G(M)$ we let $\ell_{v,w} = 0$. By the min-flow max-cut theorem the minimum value of a (σ, τ)-flow of $G(M)$ is equal to the maximum of $\sum_{v \in S, w \in T} \ell_{v,w} - \sum_{v \in S, w \in T} c_{w,v}$, where the maximum is taken over all partitions $\{S, T\}$ of the nodes of $G(M)$ with $\sigma \in S$ and $\tau \in T$. By the choice of the capacities this is precisely the maximum over all configurations C of M of $\mathrm{gus}(C, M)$. We note also that before computing the minimal flow we may reduce the graph $G(M)$ as follows: If two nodes v and w are such that v immediately precedes w on some process and either v has outdegree one, or w has indegree one (initially this holds for all immediate successors on some process), then the edge (v, w) can be contracted to a single node. Furthermore this reduction step can be iterated as long as possible. Call the resulting graph $G(M)_{\mathrm{red}}$. Note that $G(M)_{\mathrm{red}}$

may be considerably smaller than $G(M)$. Finally, since all upper capacities are ∞, we can use [11] in order to compute $\max(\sum_{v\in S, w\in T} \ell_{v,w} - \sum_{v\in S, w\in T} c_{w,v})$ in time $O(n\log(n)e)$, where n is the number of nodes of $G(M)_{\mathrm{red}}$ and e is the number of edges in a transitive reduction of $G(M)_{\mathrm{red}}$. Note that e is in $O(n)$. □

4 Bounded communication in HMSCs

The following result follows easily from Theorem 5.

Theorem 5. *The following problem is NP-complete:*
INPUT: HMSC H and positive integer b.
QUESTION: Is H $\exists^b_{\mathrm{glob}}$-bounded?

Proof. The lower bound follows directly from Theorem 5. For the upper bound note that an HMSC $H = (V, \to, P, \mu, v)$ is $\exists^b_{\mathrm{glob}}$-bounded if for every node $u \in V$ the MSC $\mu(u)$ is $\exists^b_{\mathrm{glob}}$-bounded. □

Analogously to Theorem 5 it follows that it can be decided in polynomial time whether a given HMSC is $\exists^b_{\mathrm{loc}}$-bounded for some given b.

For the universal-boundedness question for HMSCs we need the concept of the communication graph $G(M)$ of a finite MSC $M = (\mathcal{E}, P, \lambda, t, m, <)$. It is defined as $G(M) = (P, \mapsto)$, where $p_1 \mapsto p_2$ if and only if there exists a message $(s, m(s))$ in M with $\lambda(s) = p_1$ and $\lambda(m(s)) = p_2$. We say that $G(M)$ is *locally strongly connected* if every connected component of $G(M)$ is strongly connected. We say that M is *locally strongly connected* if $G(M)$ is locally strongly connected. Finally an HMSC $H = (V, \to, P, \mu, v)$ is locally strongly connected if for every cycle $v_1 \to v_2 \to \cdots \to v_n \to v_1$ of $(V, \to)$ the MSC $\mu(v_1)\mu(v_2)\cdots\mu(v_n)$ is locally strongly connected. It is easy to see that H is locally strongly connected if and only if for all simple cycles $v_1 \to v_2 \to \cdots \to v_n \to v_1$ (i.e, $v_i \neq v_j$ for $i \neq j$) the MSC $\mu(v_1)\mu(v_2)\cdots\mu(v_n)$ is locally strongly connected. For this just note that if we have some cycle $v_1 \to \cdots \to v_i = w_1 \to \cdots \to w_m \to v_i \to \cdots \to v_n \to v_1$ then $G = G(\mu(v_1)\cdots\mu(v_i)\mu(w_2)\cdots\mu(w_m)\mu(v_i)\cdots\mu(v_n))$ is the union of the two communication graphs $G(\mu(v_1)\cdots\mu(v_n))$ and $G(\mu(w_1)\cdots\mu(w_m))$. Thus if both of them are locally strongly connected then the same holds for G. Recently it was shown in [6] that an HMSC H is $\forall^b_{\mathrm{loc}}$-bounded for some b if and only if H is locally strongly connected. The fact that H is locally strongly connected if H is $\forall^b_{\mathrm{loc}}$-bounded by some b is quite easy to see. Lemma 4 below will allow us to present a simpler proof of the other direction of the result of [6], together with a sharper bound on the buffer size, which we need later. The proofs for the two following lemmas can be found in the appendix.

Lemma 4. *Let the HMSC $H = (V, \to, P, \mu, v)$ be locally strongly connected. Let $u_1 \to u_2 \to \cdots \to u_m$ be a path in $(V, \to)$ with $m > |P| \cdot |V|$ and let C be a configuration of the MSC $M = \mu(u_1)\cdots\mu(u_n)$. Then there exists a path $v_1 \to v_2 \to \cdots \to v_n$ in $(V, \to)$ and a configuration D of the MSC $N = \mu(v_1)\cdots\mu(v_n)$ such that $n < m$, $\mathrm{gus}(C, M) = \mathrm{gus}(D, N)$, and $\mathrm{lus}(C, M) = \mathrm{lus}(D, N)$.*

For a finite MSC $M = (\mathcal{E}, P, \lambda, t, m, <)$ let $S(M) = |t^{-1}(S)|$. For an HMSC $H = (V, \to, P, \mu, v)$ let $S(H) = \max\{S(\mu(u)) \mid u \in V\}$.

Lemma 5. *Let the HMSC $H = (V, \to, P, \mu, v)$ be locally strongly connected. Then H is $\forall^b_{\text{glob}}$-bounded (and hence also $\forall^b_{\text{loc}}$-bounded) for some $b \leq |P| \cdot |V| \cdot S(H)$. Furthermore if $b \in \mathbb{N}$ is minimal such that H is $\forall^b_{\text{glob}}$-bounded (resp. $\forall^b_{\text{loc}}$-bounded) then there exists a path $v_1 \to v_2 \to \cdots \to v_n$ in $(V, \to)$ and a configuration C of the MSC $M = \mu(v_1) \cdots \mu(v_n)$ such that $n \leq |P| \cdot |V|$, and $\text{gus}(C, M) = b$ (resp. $\text{lus}(C, M) = b$).*

We should remark that Theorem 2.8 of [6], which corresponds to the first statement of Lemma 5, is formulated in terms of regular MSC-expressions instead of HMSCs. Using the notation of [6], we can prove the following statement in completely the same fashion as Lemma 5, the proof is even simpler.

Lemma 6. *Let L be a set of finite MSCs over some fixed set of processes P. Assume that each $M \in L$ is locally strongly connected and $\forall^b_{\text{glob}}$-bounded (resp. $\forall^b_{\text{loc}}$-bounded). Then every MSC in L^* is $\forall^{|P| \cdot b}_{\text{glob}}$-bounded (resp. $\forall^{|P| \cdot b}_{\text{loc}}$-bounded).*

Theorem 6. *The following problem is coNP-complete:*
INPUT: HMSC H and positive integer b.
QUESTION: Is H $\forall^b_{\text{glob}}$-bounded?

Proof. We show that the complementary problem is NP-complete:
INPUT: HMSC H and positive integer b.
QUESTION: Does there exist an MSC $M \in L(H)$ which is not $\forall^b_{\text{glob}}$-bounded?
An NP-algorithm, which solves this problem proceeds, as follows: Let $H = (V, \to, P, \mu, v)$. First we guess in $(V, \to)$ a simple cycle $u_1 \to u_2 \to \cdots \to u_m \to u_1$ and a path $v_1 \to v_2 \to \cdots \to v_n$ with $n \leq |P| \cdot |V|$ together with a configuration C in the MSC $M = \mu(v_1) \cdots \mu(v_n)$. Then the algorithm outputs "yes" if and only if either the communication graph of the MSC $\mu(u_1)\mu(u_2) \cdots \mu(u_m)$ is not locally strongly connected or $\text{gus}(C, M) \geq b+1$. We claim that this NP-algorithm is correct. Clearly if the algorithm outputs "yes" then there exists an MSC in $L(H)$ which is not $\forall^b_{\text{glob}}$-bounded. On the other hand assume that there exists an MSC in $L(H)$ which is not $\forall^b_{\text{glob}}$-bounded. Either H is not locally strongly connected, which can be detected by the algorithm, or there exists some b' such that H is $\forall^{b'}_{\text{glob}}$-bounded. Let b' be minimal with this property. Then $b < b'$ and by Lemma 5 there exists a path $v_1 \to v_2 \to \cdots \to v_n$ with $n \leq |P| \cdot |V|$ together with a configuration C in the MSC $M = \mu(v_1) \cdots \mu(v_n)$ such that $\text{gus}(C, M) = b' \geq b + 1$. Again both this path and the configuration can be detected by the algorithm.

In order to prove NP-hardness we will reduce SAT to our problem. The following construction was also used in [8]. Let $\{x_1, \ldots, x_n\}$ be a set of propositional variables, and let $C = \{C_1, \ldots, C_m\}$ be a set of m clauses, where each clause C_i consists of variables and negated variables. We will define an HMSC $H = (V, \to, P, \mu, v)$ such C is satisfiable if and only if there exists an $M \in L(H)$ which is not $\forall^{m-1}_{\text{glob}}$-bounded. We define $V = \{v, v_1, \overline{v}_1, \ldots, v_n, \overline{v}_n\}$, and $\to = \{(v, v_1), (v, \overline{v}_1)\} \cup \{(v_i, v_{i+1}), (v_i, \overline{v}_{i+1}), (\overline{v}_i, v_{i+1}), (\overline{v}_i, \overline{v}_{i+1}) \mid 1 \leq i < n\}$. The set of processes is $P = \{c_i, c'_i \mid 1 \leq i \leq m\}$. It remains to define the MSCs $\mu(u)$ for $u \in V$. The MSC $\mu(v)$ is empty. The MSC $\mu(v_i)$ contains a message from process c_j to process c'_j and back from c'_j to c_j if $x_i \in C_j$. Similarly the MSC $\mu(\overline{v}_i)$ contains a message

from process c_j to process c'_j and back from c'_j to c_j if $\overline{x}_i \in C_j$. No other messages are present. It follows that C is satisfiable if and only if there exists an MSC $M \in L(H)$ such that for every $1 \leq j \leq m$ the projection of M onto the processes c_j and c'_j is a non-empty iteration of the MSC that sends a message from c_j to c'_j and back. This holds if and only if there exists an MSC $M \in L(H)$ that is not $\forall^{m-1}_{\text{glob}}$-bounded. □

It should be noted that Theorem 6 holds no matter whether the buffer bound $b \in \mathbb{N}$ is represented unary or binary. Our lower bound proof holds also for the unary representation, whereas the upper bound proof holds for the binary representation. Furthermore note that the HMSC H used for the lower-bound proof is based on an acyclic graph $(V, \rightarrow)$, thus H defines a finite MSC-language.

Theorem 7. *The following problem is coNP-complete:*
INPUT: HMSC H and positive integer b.
QUESTION: Is H $\forall^b_{\text{loc}}$-bounded?
Furthermore this problem is coNP-complete even for $b = 2$.

Proof. Membership in coNP follows in exactly the same way as in Theorem 6. In order to show coNP-hardness we will reduce NAE-SAT (not-all-equal-SAT) to the complement of our problem. We consider a collection of m clauses $C = \{C_1, \ldots, C_m\}$ each of length three, over variables $\{x_1, \ldots, x_n\}$ and we want to find out whether there is a variable assignment such that for each clause C_i, the literals of C_i do not have the same value. We will construct an HMSC H such that for a given buffer (A, B) of H there is an execution with more than 2 sends in this buffer if and only if there is an assignment as above for C. For every buffer different from (A, B) each execution of H will contain at most one message for that buffer, so buffers different from (A, B) will be universally bounded by 1.

The graph underlying H has almost the same shape as in Theorem 6. We have the vertex set $V = \{v, v_1, \overline{v}_1, \ldots, v_n, \overline{v}_n, v'\}$, and $\rightarrow = \{(v, v_1), (v, \overline{v}_1)\} \cup \{(v_i, v_{i+1}), (v_i, \overline{v}_{i+1}), (\overline{v}_i, v_{i+1}), (\overline{v}_i, \overline{v}_{i+1}) \mid 1 \leq i < n\} \cup \{(v_n, v'), (\overline{v}_n, v')\}$. Again, vertex v_i stands for x_i true, whereas $\overline{v}_i$ stands for x_i false. The HMSC H uses processes A, B and processes $P_{j,1}, P_{j,2}, P_{j,3}, P_{j,4}, N_{j,1}, N_{j,2}, N_{j,3}, N_{j,4}$ for $j \in \{1, \ldots, m\}$ ranging over the clauses. We denote as P_j-group the processes in $P_{j,1}, P_{j,2}, P_{j,3}, P_{j,4}$, and as N_j-group the processes in $N_{j,1}, N_{j,2}, N_{j,3}, N_{j,4}$. The initial node v contains two messages from A to B, followed by one message from B to each of $P_{j,1}$ and $N_{j,1}$. Each node v_i contains a message in the P_j-group for every clause C_j where x_i occurs positively, and a message in the N_j-group for every clause C_j where x_i occurs negatively. Precisely, assume that x_i is the first literal in C_j. Then v_i contains a message from $P_{j,1}$ to $P_{j,2}$. If x_i is the second literal of C_j then v_i contains a message from $P_{j,2}$ to $P_{j,3}$. Finally, if x_i is the third literal of C_j then v_i contains a message from $P_{j,3}$ to $P_{j,4}$. Here, the ordering of the literals in each clause has to respect the order $x_1, \ldots, x_n$ of the variables. We define analogously the messages from $N_{j,l}$ to $N_{j,l+1}$ in v_i, for each C_j containing $\overline{x}_i$. Finally for the messages in $\overline{v}_i$ we switch the roles of P_j and N_j. The final node v' is labelled by messages from each of $P_{j,4}$, $N_{j,4}$ to A, followed by a message from A to B.

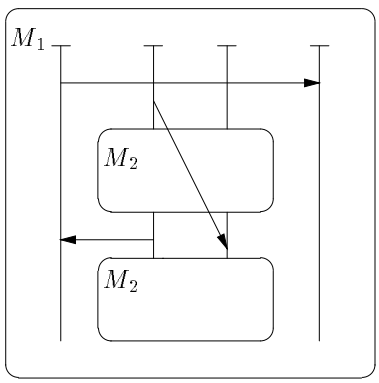
Note that paths from v to v' correspond precisely to variable assignments. Moreover, for a given path, the second receive of type $R(A, B)$ precedes in the visual order

the third send of type $S(A,B)$ if and only if there is a clause C_j in which all literals have the same value, if and only if there is some j and a $<$-path either from $P_{j,1}$ to $P_{j,4}$, or from $N_{j,1}$ to $N_{j,4}$. But this is exactly the case where C is not satisfied as an NAE-SAT instance. □

Let us remark that a simple extension of the construction from the previous proof also shows that it is coNP-complete, whether a given HMSC is $\forall^b_{\mathrm{loc}}$-bounded for some b, i.e., whether it is locally strongly connected. For this we have to add an edge from the final node v' back to the initial node v. Furthermore we have to add confirm messages that ensure that only the buffer (A,B) may contain an arbitrary number of undelivered messages. For this we simply confirm each message from a process p to q where $(p,q) \neq (A,B)$ directly by a message from q back to p.

5 Local Boundedness and Nested MSCs

A nested MSC (nMSC) M_1 is defined by a sequence of modules $(M_k)_{1 \leq k \leq m}$. Each module M_k is defined as an MSC to which we add references to modules M_i with $k < i \leq m$, by specifying the start and end of each reference to M_i on the process lines belonging to M_i. We use the definition of [1], where messages are restricted to be matched on the same hierarchy level (in particular, we don't consider ports), but they can cross submodules (see figure below). This definition is analogous to the notion of straight-line expressions, where any expression may use in its definition sub-expressions that were previously defined. The derivation tree of an nMSC is defined in the obvious way. An nMSC defines an MSC obtained by replacing inductively each reference by the MSC it defines. We denote by $\subset$ the (transitive) inclusion relation between modules, i.e., the transitive closure of the relation $M_i \subset M_k$, where M_k contains a reference to M_i. Let $P(M_k)$ be the set of processes of the MSC defined by M_k. Note that if $M_i \subset M_k$ then $P(M_i) \subseteq P(M_k)$. Thus, if M_k contains a message (s,r) from p to q that crosses a reference M_i, $i > k$, (i.e., p,q are processes of M_i and s precedes the beginning of M_i on p, whereas r succeeds the end of M_i on q), then M_i cannot contain any message from p to q, unless the FIFO-restriction is violated. For $1 \leq k \leq m$ let $\rightsquigarrow_k$ (resp. $<_k$) be the $\rightsquigarrow$-relation (resp. visual order) associated with the MSC defined by M_k (recall the definition of $\rightsquigarrow$ in Section 3).



We show in this section that both versions (existential and universal) of the local-boundedness problem for nMSCs can be solved in polynomial time, provided that the nMSCs satisfy the FIFO-restriction. Of course, the algorithms must exploit the hierarchy, since nMSCs can be exponentially more succint than the MSCs they define (i.e., a module M_k may have exponentially many copies in the MSC defined by M_1).

Lemma 7. *Let $(M_k)_{1\leq k\leq m}$ be an nMSC satisfying the FIFO-restriction. Let $1 \leq i \leq m$ and let $\mathcal{E}$ be the set of events of some copy of the MSC defined by module M_k within the MSC defined by M_1. Then $\rightsquigarrow_1 \cap\, \mathcal{E} = \rightsquigarrow_k$.*

Note that Lemma 7 does not generalize to the case where the nMSC violates the FIFO-restriction. In this case different copys of the same module may have different local $\rightsquigarrow$-relations depending on the context in which they appear in the derivation tree.

Lemma 8. *Let $(M_k)_{1\leq k\leq m}$ be an nMSC satisfying the FIFO-restriction. Suppose that the relation $\rightsquigarrow \cup <$ contains a cycle, where $\rightsquigarrow = \rightsquigarrow_1$, $< = <_1$. Then for some $k \leq |P|$ there exists a cycle of the form*

$$r_1 \rightsquigarrow s_1 < r_2 \rightsquigarrow s_2 < \cdots < r_k \rightsquigarrow s_k < r_1\,.$$

In the following we say that an event is *defined* in a module M_k if it belongs to the events of M_k, but not a module referenced in M_k.

Theorem 8. *The following problem can be solved in polynomial time:*

INPUT: nMSC $M = (M_k)_{1\leq k\leq m}$ satisfying the FIFO-restriction and positive integer b.

QUESTION: Is M $\exists^b_{\mathrm{loc}}$-bounded?

Proof. By Lemma 2 it suffices to verify that the transitive closure of the relation $< \cup \rightsquigarrow$ associated with the MSC defined by M_1 is acyclic, where $\rightsquigarrow = \rightsquigarrow_1, < = <_1$. Of course, we cannot explicitly generate the $\rightsquigarrow$-edges, since there can be exponentially many $\rightsquigarrow$-edges leading out of a copy of M_i within M_k, or vice versa. By Lemma 8 it suffices to look for a cycle containing at most $|P|$ new $\rightsquigarrow$-edges.

Let us first describe how we can compute the set $\mathrm{Succ}_<(e) = \{f \mid e < f\}$ of $<$-successors of e for any given event e of M_1. Since this set may be of exponential size, we describe it by a tuple $(k_p)_{p\in P}$ of positions, one for each process p. The position k_p corresponds to the first event $f \in \lambda^{-1}(p)$ with $e < f$. Note that if $e < f$ then there exists a chain $e = e_1 \leq f_1 < e_2 < f_2 < \cdots < f_t < e_{t+1} \leq f_{t+1} = f$ with $\lambda(e_i) = \lambda(f_i)$, $m(f_i) = e_{i+1}$, and $t < |P|$. Here $\leq$ denoted the reflexive closure of the visual order $<$. The computation of $\mathrm{Succ}_<(e)$ can be performed by induction on t. We start by setting $k_{\lambda(e)}$ to the direct successor of e on process $\lambda(e)$ and $k_{\lambda(m(e))} = m(e)$ if e is a send, all k_p that are not defined in this way are set to ∞. For the inductive step we determine for all $k_p < \infty$ and all processes $q \neq p$ the first send s of type $S(p,q)$ with $s \geq k_p$, and we compute the minimum between its matching receive and k_q. This step can be performed in time $O(|P|^3|M|)$, where $P = P(M_1)$.

A similar argument applies when we want to determine whether there is a cycle in $\rightsquigarrow \cup <$. Suppose that e is an event in M_1 (not necessarily at the highest level),

described by the position on its process. We want to compute $\mathrm{Succ}_{(<\cup\rightsquigarrow)^+}(e)$ and test whether $e \in \mathrm{Succ}_{(<\cup\rightsquigarrow)^+}(e)$. We start with $\mathrm{Succ}_{(<\cup\rightsquigarrow)^+}(e) = \mathrm{Succ}_<(e)$, represented by the tuple of positions $(k_p)_{p\in P}$. For the inductive step we determine for all $k_q < \infty$ and all processes $p \neq q$ the first receive r of type $R(p,q)$ with $r \geq k_q$, and we compute the send s of type $S(p,q)$ with $r \rightsquigarrow s$. By Lemma 7 the existence of the edge $r \rightsquigarrow s$ is determined only by M_1. Then we compute for each such s the set $\mathrm{Succ}_<(s)$ and we build the minima with $(k_p)_{p\in P}$ on every process. The overall running time is of order $O(|P|^4|M|^2)$. □

The proof for the polynomial-time algorithm for universal-local-boundedness can be found in the appendix.

Theorem 9. *The following problem can be solved in polynomial time:*
INPUT: nMSC $M = (M_k)_{1\leq k\leq m}$ satisfying the FIFO-restriction and positive integer b.
QUESTION: Is M $\forall^b_{\mathrm{loc}}$-bounded?

6 Fixed number of processes

In practice, the set of processes of an MSC can be much smaller than the number of messages. Hence we are interested in the complexity of our problem when the number of processes is fixed. The main result of this section states that for a fixed number of processes all the variants of the channel boundedness problem can be solved in polynomial time (more precisely in nondeterministic logspace). The proofs can be found in the appendix.

Theorem 10. *Let P be a fixed set of processes. The following problem is in NL:*
INPUT: Finite MSC M over the set of processes P and positive integer b.
QUESTION: Is M $\exists^b_{\mathrm{glob}}$-bounded?

Theorem 11. *Let P be a fixed set of processes. The following problem is in NL:*
INPUT: HMSC H and positive integer b.
QUESTION: Is H $\forall^b_{\mathrm{glob}}$-bounded (resp. $\forall^b_{\mathrm{loc}}$-bounded)?

7 Summary and open problems

Table 1 summarizes our results for boundedness problems for finite MSCs and HMSCs, for which we precisely determined the tractable boundedness problems. Concerning nMSCs we have shown that the two local-boundedness problems can be decided in polynomial time. The precise complexity of the two global-boundedness problems remains open for nMSCs. An NP-lower bound for existential-global-boundedness follows trivially from the NP-lower bound for finite MSCs. Concerning the upper bound we can only prove membership in PSPACE. For universal-global-boundedness we can prove membership in coNP for nMSCs, but the existence of a polynomial time algorithm remains open. Another interesting problem might be to investigate the complexity of boundedness problems for a fixed buffer-bound b, which means that b does not contribute to the input size. One might expect that the complexity of boundedness problems decreases under this restriction.

Table 1. Complexity results for finite MSCs and HMSCs

finite MSCs	∃	∀
global	NP-complete	P
local	P	P
local (nMSC)	P	P

HMSCs	∃	∀
global	NP-complete	coNP-complete
local	P	coNP-complete

References

1. B. Genest, and A. Muscholl. Pattern Matching and Membership for Hierarchical Message Sequence Charts. Submitted, 2001.
2. J. G. Henriksen, M. Mukund, K. N. Kumar, and P. Thiagarajan. On message sequence graphs and finitely generated regular MSC languages. In *Proc. of ICALP'00*, LNCS 1853, pp. 675–686, 2000.
3. J. G. Henriksen, M. Mukund, K. N. Kumar, and P. Thiagarajan. Regular collections of message sequence charts. In *Proc. of MFCS'00*, LNCS 1893, pp. 675–686, 2000.
4. N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17(5):935–938, 1988.
5. E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.
6. R. Morin. On regular message sequence chart languages and relationships to Mazurkiewicz trace theory. In *Proc. of FoSSaCS'01*, LNCS 2030, pp. 332–346, 2001.
7. M. Mukund, K. N. Kumar, and M. A. Sohoni. Synthesizing distributed finite-state systems from MSCs. In *Proc. of CONCUR'00*, LNCS 1877, pp. 521–535, 2000.
8. A. Muscholl, D. Peled, and Z. Su. Deciding properties for message sequence charts. In *Proc. of FoSSaCS'98*, LNCS 1378, pp. 226–242, 1998.
9. C. H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.
10. R. Sethi. Complete register allocation problems. *SIAM Journal on Computing*, 4(3):226–248, 1975.
11. K. Simon. On minimum flow and transitive reduction. In *Proc. of ICALP'8*, LNCS 317, pp. 535–546, 1988.
12. R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26(3):279–284, 1988.
13. P. van Emde Boas and J. van Leeuwen. Move rules and trade-offs in the pebble game. In *Proc. of the 4th GI Conference*, LNCS 67, pp. 101–112, 1979.

Appendix

Corollary 1
The following problem is NP-complete:

> INPUT: Finite dag G with exactly one node of outdegree 0, positive integer b.
> QUESTION: Can G be b-pebbled without the move rule?

Proof. Membership in NP is obvious. We prove NP-hardness by showing that a finite dag G with exactly one node of outdegree 0 can be b-pebbled if and only if G can be $(b+1)$-pebbled without the move rule. If G can be b-pebbled then obviously G can be $(b+1)$-pebbled without the move rule. Now assume that G can be $(b+1)$-pebbled without the move rule and let $\emptyset = C_1, C_2 \ldots, C_n = \emptyset$ be a sequence of subsets of V such that $|C_i| \leq b+1$ for $1 \leq i \leq n$, every C_i results from C_{i-1} by a 1-move or a 2-move, and finally every node is pebbled exactly once. We show that G can be b-pebbled by successively eliminating all C_i with $|C_i| = b+1$. Let $|C_i| = b+1$. Then we must have $2 \leq i \leq n-1$, $C_i = C_{i-1} \dot\cup \{v\}$, and $C_{i+1} = C_i \backslash \{w\}$ for some $v, w \in C_i$ such that $u \in C_{i-1}$ for all $u \in V$ with $(u,v) \in E$. We can distinguish the following three cases, where excepting the third case, we can proceed analogously to the proof of Theorem A in [13].

Case 1. $(w,v) \in E$. We proceed with the sequence $C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_j$. Note that (C_{i-1}, C_{i+1}) is a 3-move.

Case 2. $(w,v) \notin E$ and $w \neq v$. Then we can proceed with the sequence $C_1, \ldots, C_{i-1}, C_{i-1} \backslash \{w\}, C_{i+1}, \ldots, C_j$.

Case 3. $w = v$. This implies that v must be the unique node with outdegree 0. Thus v is the last node that is pebbled. So we can choose any $u \in V$ with $(u,v) \in E$ and replace the moves (C_{i-1}, C_i), (C_i, C_{i+1}) by the 3-move $(C_{i-1}, C_i \backslash \{u\})$. □

Lemma 4
Let the HMSC $H = (V, \to, P, \mu, v)$ be locally strongly connected. Let $u_1 \to u_2 \to \cdots \to u_m$ be a path in $(V, \to)$ with $m > |P| \cdot |V|$ and let C be a configuration of the MSC $M = \mu(u_1) \cdots \mu(u_n)$. Then there exists a path $v_1 \to v_2 \to \cdots \to v_n$ in $(V, \to)$ and a configuration D of the MSC $N = \mu(v_1) \cdots \mu(v_n)$ such that $n < m$, $\mathrm{gus}(C, M) = \mathrm{gus}(D, N)$, and $\mathrm{lus}(C, M) = \mathrm{lus}(D, N)$.

Proof. Let $u_1 \to u_2 \to \cdots \to u_m$ be a path in $(V, \to)$ with $m > |P| \cdot |V|$ and let C be a configuration of $M = \mu(u_1) \cdots \mu(u_m)$. Assume that $P = \{1, \ldots, |P|\}$. For each $i \in P$ let e_i be the maximal event of M which is located on process i and which belongs to C, if this event exists. Otherwise e_i is undefined. By reordering the processes suitably, we may assume that $e_1, \ldots, e_{k-1}$ are undefined and event e_i belongs to the MSC $\mu(u_{m_i})$ for $k \leq i \leq |P|$, where $1 \leq m_k \leq m_{k+1} \leq \cdots \leq m_{|P|} \leq m$. Let $m_0 = m_1 = \cdots = m_{k-1} = 0$. Since $m > |P| \cdot |V|$ either $m > m_{|P|}$, or $m_{l+1} - m_l > |V|$ for some $0 \leq l < |P|$. In the first case we may take the path $u_1 \to u_2 \to \cdots \to u_{m-1}$ for $v_1 \to \cdots \to v_n$. Now assume that $m_{l+1} - m_l > |V|$ for some $0 \leq l < |P|$. Then there exist $m_l < i < j \leq m_{l+1}$ such that $u_i = u_j$. Note that $\mu(u_i)\mu(u_{i+1}) \cdots \mu(u_{j-1}) \cap C$ contains precisely the events of $\mu(u_i)\mu(u_{i+1}) \cdots \mu(u_{j-1})$ that are located on some process $p \in \{l+1, \ldots, |P|\}$. Let G be the communication

graph of the factor $\mu(u_i)\mu(u_{i+1})\cdots\mu(u_{j-1})$ of M. Then G must be locally strongly connected. Assume that there exists a message (s,r) in $\mu(u_i)\mu(u_{i+1})\cdots\mu(u_{j-1})$ such that $s \in C$ and $r \notin C$. Since $i > m_l$ and $j-1 < m_{l+1}$, the send s is located on some process $p \geq l+1$ and the receive r is located on some process $q \leq l$. Since G is locally strongly connected, there exists a directed path from process $q \leq l$ to process $p \geq l+1$ in G. Thus there exists a message (s',r') in $\mu(u_i)\mu(u_{i+1})\cdots\mu(u_{j-1})$ such that s' is located on some process $q' \leq l$ and r' is located on some process $p' \geq l+1$. It follows that $r' \in C$ and $s' \notin C$. But this is a contradiction to the fact that C is a configuration of M. Thus the message (s,r) cannot exist. It follows that in the MSC $N = \mu(u_1)\cdots\mu(u_{i-1})\mu(u_j)\cdots\mu(u_m)$ there exists a configuration D such that $\mathrm{gus}(C,M) = \mathrm{gus}(D,N)$ and $\mathrm{lus}(C,M) = \mathrm{lus}(D,N)$. □

Lemma 5
Let the HMSC $H = (V, \to, P, \mu, v)$ be locally strongly connected. Then H is $\forall^b_{\mathrm{glob}}$-bounded (and hence also $\forall^b_{\mathrm{loc}}$-bounded) for some $b \leq |P| \cdot |V| \cdot S(H)$. Furthermore if $b \in \mathbb{N}$ is minimal such that H is $\forall^b_{\mathrm{glob}}$-bounded (resp. $\forall^b_{\mathrm{loc}}$-bounded) then there exists a path $v_1 \to v_2 \to \cdots \to v_n$ in $(V, \to)$ and a configuration C of the MSC $M = \mu(v_1)\cdots\mu(v_n)$ such that $n \leq |P|\cdot|V|$, and $\mathrm{gus}(C,M) = b$ (resp. $\mathrm{lus}(C,M) = b$).

Proof. For the first statement assume that there exists a path $v = u_1 \to u_2 \to \cdots \to u_n$ in $(V, \to)$ and a configuration C in the MSC $M = \mu(u_1)\cdots\mu(u_n)$ such that $\mathrm{gus}(C,M) > |P| \cdot |V| \cdot S(H)$. Furthermore let n be minimal with the properties above. Since each of the MSCs $\mu(u_i)$ can only contribute $S(H)$ messages (s,r) with $s \in C$ and $r \notin C$, it follows that $n > |P| \cdot |V|$, which by Lemma 4 gives us a contradiction to the minimality of n. The second statement follows immediately from Lemma 4.

Theorem 9
The following problem can be solved in polynomial time:

INPUT: nMSC $M = (M_k)_{1 \leq k \leq m}$ satisfying the FIFO-restriction, positive integer b.
QUESTION: Is M $\forall^b_{\mathrm{loc}}$-bounded?

Proof. We apply Lemma 3, that is, we check whether there exists some r of channel type $R(p,q)$ such that the send s with $r \rightsquigarrow s$ does not satisfy $r < s$, where $\rightsquigarrow = \rightsquigarrow_1$, $<= <_1$. First we compute for each M_i and each event e (or start/end of a reference) *defined in* M_i the future and the past of e *within* M_i. The future $\mathrm{Succ}_i(e)$ (past $\mathrm{Pred}_i(e)$, respectively) of e are described as $|P|$ intervals of events, one for each process.

There are two main cases to consider. Suppose that the events r (resp. s) are contained in some copy of the module M_i (resp. M_j). The two copies containing r, s are either ordered within the derivation tree of M_1, or they are incomparable w.r.t. the tree ancestor relation:

(1) Suppose first that the copy of M_i is above the copy of M_j in the derivation tree. In particular, $M_j \subset M_i$. Then we have to check for every m-th event of type

$R(p,q)$ *defined in* M_i that the $(m+b)$-th event s of type $S(p,q)$ in M_i is such that $s \in \mathrm{Succ}_i(r)$. Note that the test is applied to a linear number of events.
If the copy of M_i is below the copy of M_j in the derivation tree, in particular $M_i \subset M_j$, then we use the past of the $(m+b)$-th event of type $S(p,q)$ and check that it contains the m-th event of type $R(p,q)$.

(2) Suppose now that the copies of M_i, M_j are incomparable in the derivation tree, and let M_k be the lowest common ancestor of these copies. Since M_i, M_j both contain events on processes p, q, their copies must be ordered within M_k. Furthermore, the copy of M_i has to appear before the copy of M_j in M_k. For $r < s$ to hold in the visual order, there must be some process t in M_k such that there is a $<$-chain from r to s leading through t in M_k. More precisely there is either an event defined in M_k and located on t (or the start/end of a reference of M_k) e_t such that the m-th event r of type $R(p,q)$ in M_k belongs to $\mathrm{Pred}_k(e_t)$, whereas the $(m+b)$-th event s of type $S(p,q)$ in M_k belongs to $\mathrm{Succ}_k(e_t)$. All we have to do is to check that for each possible value m in M_k there is a suitable process t. That is, we examine the pairs of intervals $\mathrm{Pred}_k(e_t), \mathrm{Succ}_k(e_t)$ of each e_t and verify that each suitable m is such that there is some e_t with $r \in \mathrm{Pred}_k(e_t)$ and $s \in \mathrm{Succ}_k(e_t)$. This is again a polynomial test that is done for a linear number of events. □

Theorem 10
Let P be a fixed set of processes. The following problem is in NL:

INPUT: Finite MSC M over the set of processes P, positive integer b.
QUESTION: Is M $\exists^b_{\mathrm{glob}}$-bounded?

Proof. Let $M = (\mathcal{E}, P, \lambda, t, m, <)$. Our NL-algorithm guesses a sequence $\emptyset = C_1, C_2, \ldots, C_n = \mathcal{E}$ of configurations of M which forms an execution of M. Note that each configuration C_i can be stored using $|P|$ many pointers, one for each process in P. Since P is fixed we need only logarithmic space for this. Each time, a new configuration C_i is computed non-deterministicely from C_{i-1}, we calculate $\mathrm{gus}(C_i, M)$ from $\mathrm{gus}(C_{i-1}, M)$, check whether $\mathrm{gus}(C_i, M) \leq b$, and forget the old configuration C_{i-1}. Since we may assume that $b \leq |\mathcal{E}|$ (which we can check at the beginning) we can write down $\mathrm{gus}(C_i, M)$ in logarithmic space. □

Theorem 11
Let P be a fixed set of processes. The following problem is in NL:

INPUT: HMSC H and positive integer b.
QUESTION: Is H $\forall^b_{\mathrm{glob}}$-bounded (resp. $\forall^b_{\mathrm{loc}}$-bounded)?

Proof. Since NL is closed under complement [4, 12], it suffices to check in NL whether an HMSC $H = (V, \to, P, \mu, v)$ is not $\forall^b_{\mathrm{glob}}$-bounded (resp. $\forall^b_{\mathrm{loc}}$-bounded). First we show that it can be verified in NL, whether H is not locally strongly connected. For this we guess a node $u \in V$ and a cycle $u_1 \to u_2 \to \cdots \to u_m \to u_1$ in $(V, \to)$ with $u_1 = u$. Along the guessing of this cycle, only the current node u_i and the communication graph of the MSC $\mu(u_1)\mu(u_2)\cdots\mu(u_i)$ are stored. Since

P is fixed, we need only constant space in order to store this graph. The communication graph of the MSC $\mu(u_1)\mu(u_2)\cdots\mu(u_{i+1})$ can be easily constructed in logspace from the graph for the MSC $\mu(u_1)\mu(u_2)\cdots\mu(u_i)$. At the end we just have to check whether the communication graph of the whole cycle is not locally strongly connected.

Now in order to check whether H is not $\forall^b_{\text{glob}}$-bounded, we first nondeterministically branch into two cases. In the first case we check in NL whether H is not locally strongly connected, which is possible by the preceding paragraph. In the second case we first test whether $b \leq |P| \cdot |V| \cdot S(H)$. If not we reject, otherwise we have to check whether there exists a path $v_1 \to v_2 \to \cdots \to v_n$ in $(V, \to)$ together with a configuration C in the MSC $M = \mu(v_1)\cdots\mu(v_n)$ such that $\text{gus}(C, M) > b$. It remains to prove that this can be done in NL, the correctness of this procedure follows from Lemma 5. Assume that $P = \{1, \ldots, |P|\}$. We will guess the path $v_1 \to v_2 \to \cdots \to v_n$, where only the current node v_i will be stored. Furthermore we have two variables Q and g, where initially $Q = \emptyset$ and $g = 0$. The variable Q will store a subset of processes, the variable g will store an integer not larger then b. Since we already know that $b \leq |P| \cdot |V| \cdot S(H)$, also the variable g can be stored in logspace. Assume that during the execution of our NL-algorithm we go from node v_{i-1} to node v_i. We update the variables Q and g as follows: First we guess an arbitrary set $P' \subseteq P \backslash Q$ and set $Q := Q \cup P'$. Then we guess for each process $p \in P'$ an event e_p of $\mu(v_i)$ which is located on process p. Let C be the set of all events e of $\mu(v_i)$ that are either located on some process in $P \backslash Q$ (here Q refers already to the updated value), or such that e is located on some process $p \in P'$ and lies in the past of e_p. We can easily check in deterministic logspace whether C is a configuration of $\mu(v_i)$, for this it is not necessary to construct C explicitly (which would not be possible in logspace). If C is not a configuration of $\mu(v_i)$ then we reject. Otherwise we can easily compute in logspace the value $\text{gus}(C, \mu(v_i))$. If $g + \text{gus}(C, \mu(v_i)) > b$ we immediately accept, otherwise we update g by $g := g + \text{gus}(C, \mu(v_i))$ and proceed to the next node v_{i+1}.

In order to check whether an HMSC H is not $\forall^b_{\text{loc}}$-bounded we can proceed similarly. □