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

Abstract. Message sequence charts (MSCs) and high-level message sequence charts (HMSCs) are popular formalisms for the specification of communication protocols between asynchronous processes. An important concept in this context is the size of the communication buffers used between processes. Since real systems impose limitations on the capacity (or speed) of communication links, we ask whether a given HMSC can be implemented with respect to a given buffer size imposed by the environment. We introduce four different measures for buffer sizes and investigate for each of these measures the complexity of deciding whether a given MSC (or HMSC, or hierarchical MSC) 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 (MSC) and high-level message sequence charts (HMSC) 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, it is natural to ask whether an HMSC can be implemented with respect to a given buffer size imposed by the environment.

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 distinguish whether *all* linearizations of an MSC M satisfy a certain buffer bound (∀-boundedness), respectively whether *at least one* linearization of M respects the bound (∃-boundedness). Universal boundedness is a kind of safety requirement, expressing that any interleaving of the MSC execution is possible within the constraints imposed by the environment. Existential boundedness is important for instance when we want to simulate a given specification, e.g. for determining the reachable nodes of an HMSC. In this case it suffices to consider one interleaving for each MSC execution.

[⋆] Research supported by the INRIA cooperative research action FISC.

[⋆⋆] Current address: IRISA, Campus de Beaulieu, F-35042 Rennes Cedex, France.

In the second dimension we distinguish between measuring the channel 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). The local notion of boundedness is important for implementing an MSC specification in a distributed process environment. Global boundedness arises naturally when one simulates MSC executions on a single-processor environment, for instance for test purposes.

By combining both dimensions we say for instance that an 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 (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 MSCs using references, i.e., already defined MSCs (*nested MSCs*, [1]).

We note that the notion of universal-local-boundedness corresponds to the notion of channel-boundedness used in [2, 3, 5, 6] 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

For complexity results we will use standard classes like non-deterministic logarithmic space (NL), polynomial time (P), non-deterministic polynomial time (NP) and co-NP (complements of NP-problems), see [8] for definitions.

A *linearization* of a partially ordered set $(A, \prec)$ is a total order on A that extends the partial order $\prec$. The transitive closure of a binary relation E is the least transitive relation E^+ containing E. The transitive reduction of E is a minimal relation $F \subseteq E$ such that $E^+ = F^+$. For any alphabets $A \subseteq B$ and any word $w \in B^*$ we define $|w|_A$ as the number of symbols from A in w.

2.1 Message sequence charts

Following the ITU norm Z.120 a *message sequence chart (MSC)* M is a tuple $(\mathcal{E}, P, \lambda, t, m, \prec)$, where:

- $\mathcal{E}$ is a finite 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 is located.
- $t : \mathcal{E} \to \{S, R\}$ 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 $\mathrm{ch}(e)$.
- $\prec$ is a partial order on $\mathcal{E}$, also called the *visual order of* M. We require that the set of events $\lambda^{-1}(p)$ is totally ordered by $\prec$, for every $p \in P$, and that $\prec$ equals the transitive closure of the *acyclic* relation

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

The size of the MSC M is the number of events in M. 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 $\mathrm{ch}(s_1) = \mathrm{ch}(s_2)$ and $s_1 \prec s_2$ then also $m(s_1) \prec m(s_2)$, i.e, message overtaking on any channel is disallowed. A *channel* is a pair (p,q) of distinct processes. The MSC definition may also include message contents or local actions, however this is not important in the present setting. 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 $M = (\mathcal{E}, P, \lambda, t, m, \prec)$ be an MSC. A *linearization* of M is a linearization of the visual order $(\mathcal{E}, \prec)$. 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 $|K|_{\mathrm{ch}^{-1}(S(p,q))} - |K|_{\mathrm{ch}^{-1}(R(p,q))} \leq b$. 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^c_{glob}-bounded, where $c = |P|^2 \cdot b$.

For $Q \in \{\exists, \forall\}$ and $Y \in \{\mathrm{loc}, \mathrm{glob}\}$, we define Q_Y-MSC-BOUNDED as the following decision problem.
INPUT: MSC M and positive integer b.
QUESTION: Is M Q^b_Y–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 $e \prec f \in C$ 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 $\mathrm{gs}(C, M)$ (**g**lobally unmatched **s**ends). The maximum over all channels (p,q) of the number of messages (s,r) in M with $\mathrm{ch}(s) = S(p,q)$, $\mathrm{ch}(r) = R(p,q)$, $s \in C$,

and $r \notin C$ is denote by $\mathrm{ls}(C, M)$ (locally unmatched sends). 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 $\mathrm{gs}(C, M) \leq b$ (resp. $\mathrm{ls}(C, M) \leq b$).

Let $M_i = (\mathcal{E}_i, P, \lambda_i, t_i, m_i, \prec_i)$, $i = 1, 2$, be two MSCs over the same set P of processes, where furthermore $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$. Then the concatenation of M_1 and M_2 is the MSC $M_1 M_2 = (\mathcal{E}_1 \cup \mathcal{E}_2,\ P,\ \lambda_1 \cup \lambda_2,\ t_1 \cup t_2,\ m_1 \cup m_2,\ \prec)$, where

$$\prec\, = \ (\prec_1 \cup \prec_2 \cup \{(e_1, e_2) \mid e_1 \in \mathcal{E}_1, e_2 \in \mathcal{E}_2 \,:\, \lambda_1(e_1) = \lambda_2(e_2)\})^+.$$

The standard ITU definition Z.120 defines also a *high-level message sequence chart (HMSC)* as a finite transition system with nodes labeled by finite MSCs. Formally, let an HMSC H be given as $H = (V, \to, P, \mu, v)$, where $(V, \to)$ is a finite transition system with initial node v, P is the set of processes and μ maps every node $u \in V$ to a finite MSC $\mu(u)$ over the set of processes P. The MSC-language $L(H)$ defined by H is the set of all MSCs $\mu(u_1)\mu(u_2)\cdots$, where $u_1 = v$ and $u_1 \to u_2 \to \cdots$ is a (finite or infinite) maximal path in $(V, \to)$. (Formally, for infinite paths we have to define the limit of $(\mu(u_1 \cdots u_k)_{k \geq 1})$. We impose the restriction that every node u is accessible from the initial node v. Let $Q \in \{\exists, \forall\}$ and $Y \in \{\mathrm{loc}, \mathrm{glob}\}$. We say that an HMSC H is Q^b_Y-bounded, where $b \in \mathbb{N}$, if all $M \in L(H)$ are Q^b_Y-bounded. Finally, we define Q_Y-HMSC-BOUNDED as the following decision problem.
INPUT: HMSC H and positive integer b.
QUESTION: Is H Q^b_Y–bounded?

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 definition of pebble games and related results.

Let $G = (V, E)$ be a finite *directed acyclic graph* (dag) with node set V and edges $E \subseteq V \times V$. A *game-configuration* is a subset of 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:

(a) $C_1 = C_n = \emptyset$, $|C_i| \leq b$ for $1 \leq i \leq n$,

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

If instead of (c) we require that (C_i, C_{i+1}) is a 1-move or a 2-move for $1 \leq i < n$, then we say that the graph G can be *b-pebbled without the move rule.*

Theorem 1 **([9]).** *The following problem is NP-complete:*
INPUT: Finite dag G with only one node of outdegree 0, positive integer b.
QUESTION: Can G be b-pebbled?

It is not hard to see that a dag with exactly one node of outdegree 0 can be b-pebbled if and only if it can be $(b+1)$-pebbled without the move rule. Hence we obtain

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

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 boundedness problems we can argue by considering some additional ordering on events. Let us fix a bound b and an MSC $M = (\mathcal{E}, P, \lambda, t, m, \prec)$. 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 $R(p, q)$. The following lemma is easy to prove.

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

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

Proposition 1. $\exists_{\mathrm{loc}}$-MSC-BOUNDED *can be solved in linear time.*

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. $\exists_{\mathrm{glob}}$-MSC-BOUNDED *is NP-complete.*

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 $M(G) = (\mathcal{E}, P, \lambda, t, m, \prec)$ such that G can be b-pebbled without the move rule if and only if $M(G)$ is $\exists^{(b+1)}_{\text{glob}}$-bounded. With each node $v \in V$ we associate the set of processes $P_v = \{p_{in}(v), p(v), p_{out}(v)\} \cup \{p(v, w) \mid (v, w) \in E\}$, 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 (see Figure 1). We have $\lambda(v) = p_{in}(v)$ and $\lambda(\overline{v}) = p_{out}(v)$. For each node v there is a message $(v, \overline{v})$ from process $p_{in}(v)$ to process $p_{out}(v)$. This messages crosses the chain of the two messages from $p_{in}(v)$ to $p(v)$ and from $p(v)$ to $p_{out}(v)$. Furthermore for each edge $(u, v) \in E$ we have exactly one message from process $p(u, v)$ to $p_{in}(v)$ and back from $p_{in}(v)$ to $p(u, v)$. Finally if $(v, w_1), \ldots, (v, w_n)$ are all outgoing edges of node v (listed in an arbitrary order) then there is exactly one message from process $p_{out}(v)$ to $p(v, w_1)$ and back and exactly one message from process $p(v, w_i)$ to $p(v, w_{i+1})$ and back ($1 \le i \le n - 1$). The order of the events on the processes in P_v is shown in Figure 1, where we show an example where $(u_1, v), (u_2, v), (v, w_1), (v, w_2)$, and (v, w_3) are the adjacent edges of v. The process names labeling message arrows specify the source, resp. target process of the message. Note that $\prec$ is indeed acyclic, for instance the sends of the messages from process $p(u, v)$ to $p_{in}(v)$ must precede the sends of the messages from process $p(v, w)$ to $p_{in}(w)$, with $(u, v), (v, w)$ edges of the dag. Moreover, it is easy to check that $M(G)$ respects the FIFO-restriction, in fact this was the only reason for introducing process $p(v)$. The crucial point of our construction is that the restriction $\prec|_{V \cup \overline{V}}$ of the visual order $\prec$ 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 $p(u_i, v)$ to $p_{in}(v)$ for $1 \le i \le k$, where $(u_1, v), \ldots, (u_k, v)$ are all incoming edges of node v.
(2) Execute send event v on process $p_{in}(v)$.
(3) Send and immediately receive the message from process $p_{in}(v)$ to $p(u_i, v)$ for $1 \le i \le k$.
(4) Send and immediately receive the message from process $p_{in}(v)$ to $p(v)$, followed by the message from $p(v)$ to $p_{out}(v)$.
(5) Send and immediately receive the message from process $p_{out}(v)$ to $p(v, w_1)$ followed by the messages from $p(v, w_i)$ to $p(v, w_{i+1})$ for $1 \le i < n$, where $(v, w_1), \ldots, (v, w_n)$ are all outgoing edges of node v.

Of course if v has indegree 0 (resp. outdegree 0) then (1) and (3) (resp. (5)) disappear. 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:

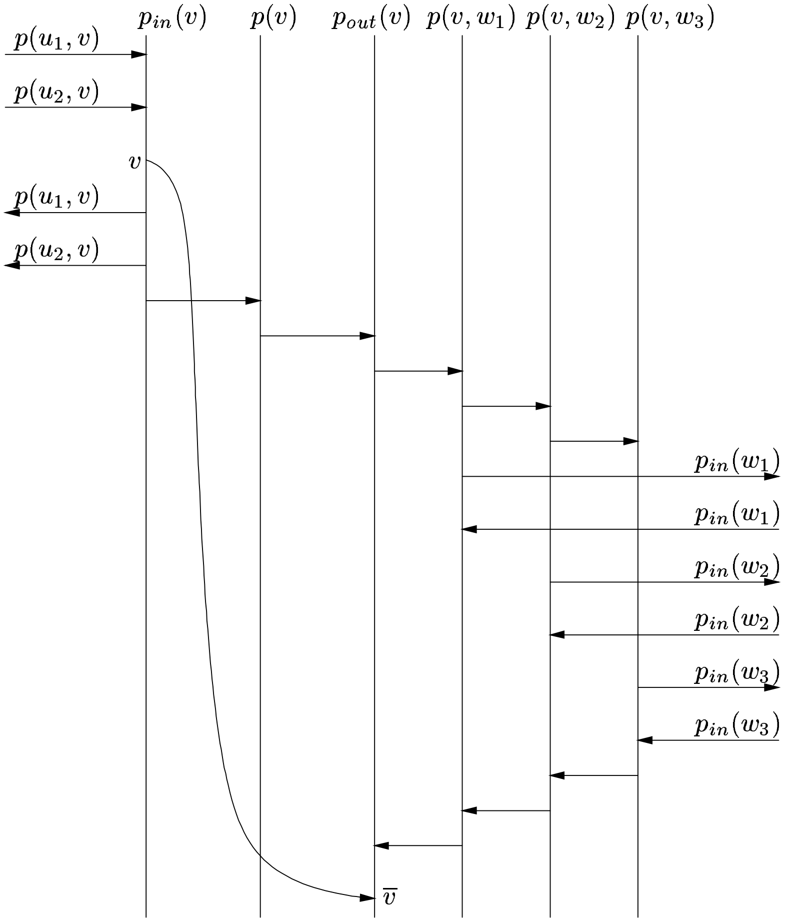


Fig. 1. Communication between the processes in P_v in the MSC $M(G)$

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

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 $V \cup \overline{V} \subseteq \mathcal{E}$. Since of course $\pi(L)$ is a linearization of $\prec|_{V \cup \overline{V}}$, by Lemma 1 it suffices to prove the following claim:

Claim 3: 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 $L_2 = eL_3$, where $e \in \mathcal{E}$ (note that we must have $L_2 \neq \epsilon$). If e would be a send event then $|L_1 ve|_{t^{-1}(S)} - |L_1 ve|_{t^{-1}(R)} \geq b+2$, 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 $v \prec \overline{u}$, since if by the construction of $M(G)$ this implies that several events occur between v and $\overline{u}$. It follows that $L' = L_1 \overline{u} v L_3$ is also a linearization of $M(G)$ 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 3 and the theorem. □

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

Lemma 3. *An MSC is $\forall^b_{\text{loc}}$-bounded if and only if $\rightsquigarrow$ is contained in $\prec$.*

Proposition 2. $\forall_{\text{loc}}$-MSC-BOUNDED *can be solved in time $O(|M|^2)$.*

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

Theorem 3. $\forall_{\text{glob}}$-MSC-BOUNDED *can be solved in time $O(|M|^2 \log(|M|))$.*

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 $\text{gs}(C, M) > b$. This question can be answered in polynomial time using the min-flow max-cut theorem, see e.g. [4]. 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 $\text{gs}(C, M)$. 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. 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 [10] 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 $O(n \log(n) r)$, where n is the number of nodes of $G(M)_{\mathrm{red}}$ and $r \in O(n)$ is the number of edges in a transitive reduction of $G(M)_{\mathrm{red}}$. □

4 Bounded communication in HMSCs

The following result follows easily from Theorem 4.

Theorem 4. $\exists_{\mathrm{glob}}$-HMSC-BOUNDED *is NP-complete.*

Proof. The lower bound follows directly from Theorem 2. For the upper bound note that an HMSC $H = (V, \rightarrow, P, \mu, v)$ 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 4 it follows that $\exists_{\mathrm{loc}}$-HMSC-BOUNDED can be solved in linear time.

For the universal-boundedness question for HMSCs we need the concept of the communication graph $G(M)$ of a finite MSC $M = (\mathcal{E}, P, \lambda, t, m, \prec)$. 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 $H = (V, \rightarrow, P, \mu, v)$ is locally strongly connected if for every cycle $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ of $(V, \rightarrow)$ 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 $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ (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 $v_1 \rightarrow \cdots \rightarrow v_i = w_1 \rightarrow \cdots \rightarrow w_m \rightarrow v_i \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ 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 [5] 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 [5], together with a sharper bound on the buffer size, which we need later.

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

Proof (sketch). The basic idea is the following: Mark on each process line the cut-point, where C cuts this line. Since $m > |P| \cdot |V|$ we can find a loop $u_i \to u_{i+1} \to \cdots \to u_j = u_i$ inside our path $u_1 \to \cdots \to u_m$ such that none of the resulting $|P|$ cut-points is located in the factor $M' = \mu(u_i) \cdots \mu(u_j)$ of M. Since H is locally strongly connected, none of the messages of M' can go from C to its complement, and we may omit the loop $u_i \to \cdots \to u_j$ from the original path. □

For a finite MSC $M = (\mathcal{E}, P, \lambda, t, m, \prec)$ 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 $\text{gs}(C, M) = b$ (resp. $\text{ls}(C, M) = b$).*

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

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 5. $\forall_{\text{glob}}$-HMSC-BOUNDED *is coNP-complete.*

Proof. We show that the complementary problem is NP-complete:
INPUT: HMSC H and positive integer b.
QUESTION: Is there an MSC $M \in L(H)$ which is not $\forall^b_{\text{glob}}$-bounded?
An NP-algorithm that solves this problem proceeds as follows: Fix an HMSC $H = (V, \to, P, \mu, v)$. 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 $\text{gs}(C, M) \geq b + 1$. 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 $\text{gs}(C, M) = b' \geq b + 1$. Again both this path and the configuration can be detected by the algorithm.

In order to prove NP-hardness, we reduce SAT to our problem. A construction similar to the following one was also used in [7]. Let $\{x_1, \ldots, x_n\}$ be a set of propositional variables, and let $C = \{C_1, \ldots, C_m\}$ be a set of clauses, where each clause C_i consists of variables and negated variables. We construct an HMSC $H = (V, \rightarrow, P, \mu, v)$ such that C is satisfiable if and only if there exists $M \in L(H)$ which is not $\forall_{\text{glob}}^{m-1}$-bounded. Let $V = \{v, v_1, \overline{v}_1, \ldots, v_n, \overline{v}_n\}$ and $\rightarrow = \{(v, v_1), (v, \overline{v}_1)\} \cup \{(v_i, v_{i+1}), (v_i, \overline{v}_{i+1}), (\overline{v}_i, v_{i+1}), (\overline{v}_i, \overline{v}_{i+1}) \mid 1 \leq i < n\}$. 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 $\forall_{\text{glob}}^{m-1}$-bounded. □

It should be noted that Theorem 5 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 set of MSCs.

Theorem 6. $\forall_{\text{loc}}$-HMSC-BOUNDED *is coNP-complete. Furthermore this problem is coNP-complete even if the input parameter b is fixed to $b = 2$.*

Proof. Membership in coNP follows in exactly the same way as in Theorem 5. 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 channel (A, B) of H there is an execution with more than 2 sends in the corresponding buffer if and only if there is an assignment as above for C. For every channel different from (A, B) each execution of H will contain at most one message for that buffer, so channels different from (A, B) will be universally bounded by 1.

The graph underlying H is similar to the one from Theorem 5. The node set is $V = \{v, v_1, \overline{v}_1, \ldots, v_n, \overline{v}_n, v'\}$, and $\rightarrow = \{(v, v_1), (v, \overline{v}_1), (v_n, v'), (\overline{v}_n, v')\} \cup \{(v_i, v_{i+1}), (v_i, \overline{v}_{i+1}), (\overline{v}_i, v_{i+1}), (\overline{v}_i, \overline{v}_{i+1}) \mid 1 \leq i < n\}$. 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

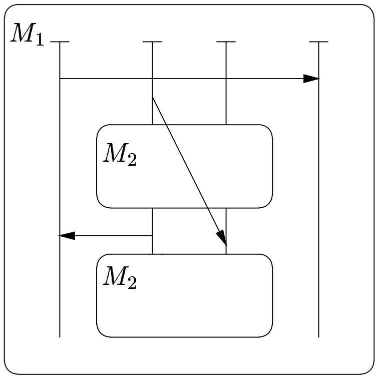
x_i occurs negatively. Precisely, v_i contains a message from $P_{j,k}$ to $P_{j,k+1}$, if x_i is the k-th literal of C_j, $k \in \{1,2,3\}$. Here, the ordering of the literals in each clause has to respect the order $x_1, \dots, x_n$ of the variables. We define analogously the messages from $N_{j,\ell}$ to $N_{j,\ell+1}$ 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 labeled 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 some j and a $\prec$-path either from $P_{j,1}$ to $P_{j,4}$, or from $N_{j,1}$ to $N_{j,4}$. But this holds if and only if there is a clause C_j in which all literals have the same value under the variable assignment corresponding to the chosen path from v to v'. 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 $\forall^b_{\text{loc}}$-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)* is a sequence $M = (M_k)_{1 \le k \le m}$ of modules M_k. Each module M_k is defined as an MSC to which we add references to modules M_i with $k < i \le m$, 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 the following figure.



The definition above is analogous to the notion of straight-line expressions, where any expression may use in its definition sub-expressions that were previously defined. Each module M_k of M can be expanded to a finite MSC $\mathrm{flat}(M_k)$ by replacing inductively in M_k each reference to a module M_j ($j > k$) by the MSC $\mathrm{flat}(M_j)$. Finally we define $\mathrm{flat}(M) = \mathrm{flat}(M_1)$. Let $P(M_k)$ be the set of processes of $\mathrm{flat}(M_k)$. Note that if an occurrence of $\mathrm{flat}(M_i)$ appears within $\mathrm{flat}(M_k)$ 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. 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 nMSC M satisfies the FIFO-restriction, i.e., $\mathrm{flat}(M)$ satisfies the FIFO-restriction. Of course, the algorithms must exploit the hierarchy, since nMSCs can be exponentially more succinct than the MSCs they define (i.e., a module M_k of M may have exponentially many copies in the MSC $\mathrm{flat}(M)$).

For the further consideration let us fix an nMSC $M = (M_k)_{1 \leq k \leq m}$. For $1 \leq k \leq m$ let $\rightsquigarrow_k$ (resp. $\prec_k$) be the $\rightsquigarrow$-relation (resp. visual order) associated with $\mathrm{flat}(M_k)$ (recall the definition of $\rightsquigarrow$ in Section 3). Furthermore let $\prec = \prec_1$, $\rightsquigarrow = \rightsquigarrow_1$, and $P = P(M_1)$.

Lemma 7. *Let M satisfy the FIFO-restriction. Let $\mathcal{E}$ be the set of events of some occurrence of* $\mathrm{flat}(M_k)$ *in* $\mathrm{flat}(M)$, $1 \leq k \leq m$. *Then* $\rightsquigarrow \cap\, (\mathcal{E} \times \mathcal{E}) \;= \rightsquigarrow_k$.

Note that Lemma 7 does not hold for the case where the nMSC violates the FIFO-restriction. In this case different occurrences of the MSC $\mathrm{flat}(M_k)$ in $\mathrm{flat}(M)$ may have different local $\rightsquigarrow$-relations depending on their context.

Lemma 8. *Suppose that the relation $\rightsquigarrow \cup \prec$ contains a cycle. Then for some $k \leq |P|$ there exists a cycle of the form $r_1 \rightsquigarrow s_1 \prec r_2 \rightsquigarrow s_2 \prec \cdots \prec r_k \rightsquigarrow s_k \prec r_1$.*

In the following we call an event e of the MSC $\mathrm{flat}(M)$ a *top level event* if it belongs to the events of the module M_1 (but not to the events of some module M_k with $k > 1$).

Theorem 7. *Let $Q \in \{\forall, \exists\}$. The following problem can be solved in polynomial time:*

INPUT: nMSC M satisfying the FIFO-restriction and positive integer b.
QUESTION: Is $\mathrm{flat}(M)$ *Q^b_{loc}-bounded?*

Proof. We will only consider existential-local boundedness. For universal-local boundedness we can argue similarly. By Lemma 2 it suffices to verify that the transitive closure of the relation $\prec \cup \rightsquigarrow$ associated with the MSC $\mathrm{flat}(M)$ is acyclic. 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 $\mathrm{Succ}_\prec(e) = \{f \mid e \prec f\}$ of $\prec$-successors of e for any given event e of $\mathrm{flat}(M)$, described by the position on its process. 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 $e \prec f$. Note that if $e \prec f$ then there exists a chain $e = e_1 \preceq f_1 \prec e_2 \prec f_2 \prec \cdots \prec f_t \prec e_{t+1} \preceq f_{t+1} = f$ with $\lambda(e_i) = \lambda(f_i)$, $m(f_i) = e_{i+1}$, and $t < |P|$. Here $\preceq$ denotes the reflexive closure of the visual order $\prec$. The computation of $\text{Succ}_{\prec}(e)$ 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 $s \succeq k_p$, and we compute the minimum between its matching receive and k_q. This step can be performed in time $O(|P|^3|M|)$.

A similar argument applies when we want to determine whether there is a cycle in $\rightsquigarrow \cup \prec$. Of course we cannot check for every event e of $\text{flat}(M)$ whether $e \in \text{Succ}_{(\prec \cup \rightsquigarrow)^+}(e)$, since there might be exponentially many such events. But in fact it suffices to show that it can be checked in polynomial time whether $\rightsquigarrow \cup \prec$ contains a cycle that visits some top level event of $\text{flat}(M)$, because then we may apply this polynomial time algorithm to each of the nMSCs $(M_k)_{i \leq k \leq n}$ for $1 \leq i \leq n$. For the correctness of the resulting algorithm we have to use Lemma 7 which ensures that $\prec \cup \rightsquigarrow$ contains a cycle which is completely contained in some occurrence of $\text{flat}(M_k)$ within $\text{flat}(M)$ if and only if $\prec_k \cup \rightsquigarrow_k$ contains a cycle.

Thus, let e be some top level event of $\text{flat}(M)$. We will compute $\text{Succ}_{(\prec \cup \rightsquigarrow)^+}(e)$ and test whether $e \in \text{Succ}_{(\prec \cup \rightsquigarrow)^+}(e)$. We start with $\text{Succ}_{(\prec \cup \rightsquigarrow)^+}(e) = \text{Succ}_{\prec}(e)$, 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 $r \succeq k_q$, and we compute the send s of type $S(p,q)$ with $r \rightsquigarrow s$. Then we compute for each such s the set $\text{Succ}_{\prec}(s)$ and we build the minima with $(k_p)_{p \in P}$ on every process. The overall running time is of order $O(|P|^4|M|^2)$. □

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).

Theorem 8. *Let $Q \in \{\forall, \exists\}$ and $Y \in \{\text{glob}, \text{loc}\}$. Let P be a fixed set of processes. The following problem is solvable in polynomial time (precisely, it is NL-complete):*

INPUT: HMSC M over the set of processes P and positive integer b.
QUESTION: Is M Q_Y^b-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. To appear in *Proc. of LATIN 2002*.
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. E. L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, 1976.
5. 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.
6. 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.
7. A. Muscholl, D. Peled, and Z. Su. Deciding properties for message sequence charts. In *Proc. of FoSSaCS’98*, LNCS 1378, pp. 226–242, 1998.
8. C. H. Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.
9. R. Sethi. Complete register allocation problems. *SIAM Journal on Computing*, 4(3):226–248, 1975.
10. K. Simon. On minimum flow and transitive reduction. In *Proc. of ICALP’88*, LNCS 317, pp. 535–546, 1988.