Divergence and channel-bound of high-level message sequence charts revisited

Florent Avellaneda & Rémi Morin

Laboratoire d'Informatique Fondamentale de Marseille
Université de la Méditerranée — UMR 6166 — CNRS

Message Sequence Charts (MSCs) are a popular model often used for the documentation of telecommunication protocols. They profit by a standartized visual and textual presentation (ITU-T recommendation Z.120) and are closed to other formalisms such as sequence diagramms of UML. An MSC gives a graphical description of the intended communications between processes. Usually it abstracts away from the values of variables and the actual contents of messages. However, this formalism can be used at some early stage of design to detect errors in the specification [4]. In this work we focus on the problems of checking process divergence, as introduced in [2], and how to compute an appropriate buffer size for channels, as already investigated in [5].

High-level MSCs (HMSCs), or equivalently MSC graphs, are a usual formalism to describe possibly infinite sets of scenarios in some algebraic way. Because asynchronous distributed systems provide no information about the relative speed of processes or the delay for a message to be delivered, *divergence* can appear in specifications: This means that there is no bound on the number of pending messages along an execution of specified scenarios. However a simple criterion allows us to decide whether a given HMSC is not divergent [2, Th. 5]. We have to check that all connected components of the communication graph of any (simple) loop are strongly connected.

In [2], Ben-Abdallah and Leue derived from this property an algorithm to check divergence, which is exponential in the number of states in the HMSC and linear in the number of channels. One has simply to search for all simple loops and to compute the strongly connected components of the resulting communicating graph [7]. It follows that checking Divergence is in NP. Now an alternative algorithm was suggested in [1]: It consists in first fixing a diverging channel and some associated partition of processes and next searching for a simple loop matching these message exchanges, by computing the strongly connected components of the HMSC reduced to corresponding transitions [7]. This second approach is exponential in the number of processes, but only linear in the number of transitions in the HMSC. As already stated in [1, Th. 7] checking divergence of HMSCs in NP-complete. We shall present here a simple and linear reduction from SAT to Divergence. More interesting we present a simple

reduction from Divergence to SAT which allows for using SAT-solvers to check Divergence in MSC specifications. In the same way slightly more involved techniques can be developped to check *local-synchronization* [1, 3, 6] similarly.

Given a non-divergent HMSC, a natural issue is to compute a buffer size for channels so that any pending message can be stored within the system before it gets delivered. As established by Lohrey and Muscholl, checking whether a buffer size is appropriate for all scenarios of a (possibly divergent) HMSC is co-NP-complete [5, Th. 4.6]. Since the HMSC used in the proof of this theorem shows no loop, this result extends immediately to non-divergent HMSCs. Thus, computing an optimal appropriate buffer size for a non-divergent HMSC is hard. In order to cope with this difficulty, we show that $p \times n$ is an appropriate buffer size for any non-divergent HMSC with p processes and at most n identical message exchanges in all transitions.

References

1. Rajeev Alur and Mihalis Yannakakis. Model checking of message sequence charts. In Jos C. M. Baeten and Sjouke Mauw, editors, *CONCUR*, volume 1664 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 1999.
2. Hanêne Ben-Abdallah and Stefan Leue. Syntactic detection of process divergence and non-local choice inmessage sequence charts. In Ed Brinksma, editor, *TACAS*, volume 1217 of *Lecture Notes in Computer Science*, pages 259–274. Springer, 1997.
3. Jesper G. Henriksen, Madhavan Mukund, K. Narayan Kumar, Milind A. Sohoni, and P. S. Thiagarajan. A theory of regular MSC languages. *Inf. Comput.*, 202(1):1–38, 2005.
4. Gerard J. Holzmann. Early fault detection tools. In Tiziana Margaria and Bernhard Steffen, editors, *TACAS*, volume 1055 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1996.
5. Markus Lohrey and Anca Muscholl. Bounded MSC communication. *Inf. Comput.*, 189(2):160–181, 2004.
6. Anca Muscholl and Doron Peled. Message sequence graphs and decision problems on mazurkiewicz traces. In Miroslaw Kutylowski, Leszek Pacholski, and Tomasz Wierzbicki, editors, *MFCS*, volume 1672 of *Lecture Notes in Computer Science*, pages 81–91. Springer, 1999.
7. Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.