First publ. in: Computer Journal 37 (1994), 9, pp. 814-817

Comments on a Proposed Semantics for Basic Message Sequence Charts

Peter B. Ladkin[1] and Stefan Leue[2]

[1]INRIA Lorraine, Nancy, France. Peter.Ladkin@loria.fr

[2]Institut für Informatik, Universität Bern, Switzerland. leue@iam.unibe.ch

In [MR94], Mauw and Reniers propose an interpretation for a basic form of Message Sequence Charts (MSCs) [CCI92], based on process algebra and Plotkin semantics. They "*use [the textual notation for Basic MSCs] for the definition of the semantics*" [MR94, Section 2.3]. They define "*the semantics of a Basic Message Sequence Chart [as] the free merge of its constituent instances. By this construction we enable all interleavings of the message outputs and message inputs*" [MR94, Section 5.2]. They also claim to be able to give (in other work) a "*complete algebraic semantics for Message Sequence Charts*" [MR94, Section 1].

We have done more extensive work than reported in [MR94], summarised in [LL94] which they reference, based on calculating the global states described by an MSC [LL92][LLar]. (For Basic MSCs, this calculation is very easy.) However, our attempts to compare their work with ours met with some difficulties, which we explain here.

- Firstly, the textual definition which Mauw and Reniers use does not suffice to describe Basic Message Sequence Charts. In [LL95] (also included in [LL92]), we give an example of two different graphical Basic MSCs which have different sets of allowed executions, but which nevertheless have the *same* representation in 'textual notation'. This shows that the textual representation does not suffice to describe Basic MSCs; *prima facie*, Mauw and Reniers are giving a semantics to a trivial programming language and not to the graphical Basic MCSs, as they claim.
- Secondly, we are uncertain as to what Mauw and Reniers consider to be the meaning of Basic MSCs. Their paper says the following. The semantics is given by taking the "*free merge of [..] constituent instances, [enabling] all interleavings of the message inputs and outputs*" [MR94, Section 5.2]. Translating into English, and conforming with their example in Section 5.3, they describe "*the interpretation of this [Basic MSC]*" in terms of the allowable temporal orderings of message `sends` and `receives`. Hence, they are considering a straightforward ordering semantics. They also give a deductive calculation of such an ordering for their example. But calculating the interleavings of a Basic Message Sequence Chart is easy, and doesn't need such apparatus. Just define the global state as the collection of positions of the program counters of the processes, and follow through the Basic MSC from the start state to see how the processes may progress (an easy example is given in [LL94]). This process is trivial for the example they give in Figure 3. Mauw and Reniers prefer to define process algebra and state-operator axioms, and use these axioms to calculate a term which

denotes essentially the same thing. The question is: why? The size of the resulting term appears to be of similar complexity as the entire state-transition calculation itself. But the state-transition calculation doesn't require the use of 24 axioms and a Plotkin semantics.

So we queried Mauw on his approach. He said that his semantics "*does not give the set of allowable traces. It gives a process algebra term which can be interpreted in any model of the theory, including bisimulation semantics and trace semantics*" [Mau94]. (Which theory he is referring to is unclear.) This puzzled us, since it suggests that Mauw believes, contrary to what he suggests and illustrates in the paper, that the process algebra term is something other than a simple description of the interleavings of atomic `send` and `receive` actions. By way of clarification, he added "*there are finer semantics too, which for example allow you to test if the moment of choice is respected in your implementation, or if the deadlock behavior is respected [..].*" [Mau94]. However, there is no such object as a '*moment of choice*' in an MSC, and deadlock behavior is represented in the usual way in the collection of global states and transitions.

We believe the main purpose of a semantics is to define for each MSC the set of possible allowed orderings of atomic actions, which for MSCs are the `send` and `receive` actions of individual processes. Mauw and Reniers have not explained what their idea of a "*finer semantics*" amounts to. A reasonable comparison hinges on Mauw and Renier clarifying the apparent confusion between what they say and show in the paper, and what they say in correspondence.

(If representing traces by a process algebra term has an advantage over compact representations of the global state space, Mauw has given no indication of what that might be. There are many compactification schemes available for global state space representations, see for example [CK91, LS92, BP93]. Most of the existing practical analysis tools for telecommunications systems, e.g., [Hol91], use state space interpretations, as do commercial MSC tools such as those provided by Verilog.)

- Thirdly, Mauw and Reniers claim to be able to give a "*complete algebraic semantics for Message Sequence Charts*" [MR94, Section 1]. We don't see how this can be so. The MSC standard includes *conditions*, which are to be used to 'join' MSCs, in such a way that different MSCs may describe consecutive parts of the same system execution. (Basic MSCs do not include conditions, which simplifies things considerably.) However, Mauw notes that he "*[does] not consider the composition of MSCs with conditions. It is recognised by the ITU study group that one needs explicit composition operators for combining MSCs.*" [Mau94]. We do agree with Mauw that unrestricted use of conditions in MSCs are problematic. For example, they seem to imply the existence of unbounded global shared variables containing the complete control history of each process [LL95]. However, the current MSC standard does not restrict the use of conditions to unproblematic cases. Until it does, it is hard to see how Mauw and Reniers, or anybody else, can claim to be able to give a "*complete algebraic semantics for Message Sequence Charts*". [LL92] defines a composition of MSCs by means of conditions, insofar as we believe it can be done.

In summary, Mauw and Reniers have presented a semantics in terms of process algebra and Plotkin rules for Basic Message Sequence Charts, relying on a textual representation which does not correspond with the Charts. They have shown how to use this algebraic apparatus to calculate a term representing the set of allowed temporal interleavings of `send`s and `receive`s, but have not demonstrated the advantage of their method over a simple calculation of global states. Mauw and Reniers have not provided any justification for considering a 'finer semantics' than the interleavings of atomic `send` and `receive`s, and their paper belies that any such finer semantics underlies their effort. They also claim to be giving a 'complete algebraic semantics' for MSCs, but until the problems with the use of conditions as defined by the standard are resolved, we don't believe this is technically feasible.

We are also concerned by Mauw and Reniers's characterisation of our work, which suggests they may not have understood what we have done. Mauw and Reniers describe our semantics as one of some "*attempts towards [..] a formal semantics*". They also note that "*None of these papers contain a formal semantics of the complete language*". Mauw notes that we do not give an explicit semantics for "*local actions, timers, environments, coregions and refinement*" [Mau94]. Neither do Mauw and Reniers in [MR94]. In fact, in [LL92] [LL94] we gave meaning to all of the constructs that Mauw and Reniers consider, and more: namely, defining the *composition* of MSCs by means of *conditions*, and showing how to specify by means of Manna-Pnueli temporal logic the liveness properties which thereby need to be defined to determine a unique set of allowable executions.

We note that Baeten and Mauw consider a composition operator for MSCs in [BM95]. We have not yet had the opportunity to compare it with our composition in [LL92], but presume that they have done so in their paper.

We became aware of the work of Mauw and Reniers through their announcement to the ITU-TS Message Sequence Chart mailing list of [MR94] in early 1994. They became aware of our work during its presentation to the CCITT (now ITU-TS) Q.MSC standards committee meeting in Geneva in November 1992, which Mauw attended, when [LL92] was distributed and accepted as input document WD-3-13. Mauw and Reniers have not yet addressed the work in [LL92] [LL94] [LLar], and have not offered their own comparison. We hope, however, that they will clarify the technical problems we have raised regarding their approach, to enable us or others to make the necessary comparison.

References

[BM95] J.C.M. Baeten and S. Mauw. Delayed choice: an operator for joining message sequence charts. In *[HL94]*. 1995. To appear.

[BP93] G. v. Bochmann and D. K. Probst, editors. *CAV'92: Computer Aided Verification*, volume 663 of *Lecture Notes in Computer Science*. Springer Verlag, 1993.

[CCI92] CCITT. Recommendation Z.120: Message Sequence Chart (MSC). CCITT, Geneva, 1992.

[CK91] E. M. Clarke and R. P. Kurshan, editors. *Computer Aided Verification: Proceedings of CAV'90*, volume 531 of *Lecture Notes in Computer Science*. Springer Verlag, 1991.

[HL94] D. Hogrefe and S. Leue, editors. *Formal Description Techniques, VII.* Proceedings of the Seventh International Conference on Formal Description Techniques. Chapman & Hall, 1994. To appear.

[Hol91] G. J. Holzmann. *Design and Validation of Computer Protocols.* Prentice-Hall International, 1991.

[LL92] P. B. Ladkin and S. Leue. Interpreting message sequence charts. Technical Report IBM RJ 8965, IBM Almaden Research Center, San Jose, CA, 1992.

[LL94] P. B. Ladkin and S. Leue. What do Message Sequence Charts mean? In *[TAU94]*. 1994.

[LL95] P. B. Ladkin and S. Leue. Four issues concerning the semantics of Message Flow Graphs. In *[HL94]*. 1995. To appear.

[LLar] P. B. Ladkin and S. Leue. Interpreting Message Flow Graphs. *Formal Aspects of Computing*, To appear.

[LS92] K. G. Larsen and A. Skou, editors. *Computer Aided Verification: Proceedings of CAV'91*, volume 575 of *Lecture Notes in Computer Science*. Springer Verlag, 1992.

[Mau94] S. Mauw. Personal communication, 1994.

[MR94] S. Mauw and M.A. Reniers. An algebraic semantics of basic message sequence charts. *The Computer Journal*, 37(4), 1994.

[TAU94] R. L. Tenney, P. D. Amer, and M. Ü. Uyar, editors. *Formal Description Techniques, VI.* IFIP Transactions C, Proceedings of the Sixth International Conference on Formal Description Techniques. North-Holland, 1994.