The Incompleteness
of Misra and Chandy's Proof Systems

Van Nguyen

TR 84-634
September 1984

Department of Computer Science
Cornell University
Ithaca, New York 14853

The Incompleteness of Misra and Chandy's Proof Systems

Van Nguyen

Computer Science Department

Cornell University

ABSTRACT

In this paper we show that Misra and Chandy's proof systems for networks of communicating processes ([1, 2]) are incomplete.

1. Introduction

In [1, 2], proof techniques for networks of processes in which component processes communicate exclusively through messages are given. A process is specified solely by its input-output behaviors. This makes proofs and proof rules simple, since internal details of a network are ignored. The proof technique in [2], an extension of that in [1], allows specification of liveness properties.

We give a simple example to show that these proof systems are incomplete. That is, there exists a specification that is true but not provable in the systems. The example is not contrived but very natural.

2. The proof systems

We assume familiarity with the proof systems in [1, 2], but provide a small introduction. In [1], a specification of a process h has the form r | h | s. Formally, this means:

(i) s holds for the empty trace;

(ii) if r holds up to point k in any trace of h, then s holds up to point (k+ 1) in that trace, for all $k \geq 0$.

The proof rules can be conveniently stated in one rule:

If (i) $r_i \mid h_i \mid s_i$, i =1, 2,...

(ii) $\wedge_i s_i \Rightarrow s$

(iii) $\wedge_i s_i \wedge r \Rightarrow \wedge_i r_i$

then r | h | s holds, where h is the network formed from the h_i's.

The proof system in [2] is similar, except that there is one more assertion in the specification and two more conditions in the proof rule. This extension makes it possible to specify and prove (some) liveness properties.

A specification has the form r $\mid \frac{h}{q} \mid$ s. Formally, this means:

(i) s holds for the empty trace;

(ii) if r holds up to point k in any trace of h, then s holds up to point (k+1) in that trace, for all $k \geq 0$;

(iii) if r holds at all points of a trace t of h and q holds for t then there exists a trace t' of h that is an extension of t (i.e. t' is t followed by an event).

The proof rule is:

If (i) $r_i \mid \frac{h_i}{q_i} \mid s_i$, i = 1, 2,...

(ii) $\wedge_i s_i \Rightarrow s$

(iii) $\wedge_i s_i \wedge r \Rightarrow \wedge_i r_i$

(iv) $\wedge_i s_i \wedge q \Rightarrow \vee_i q_i$

(v) $\wedge_i s_i \wedge q \Rightarrow \sum_i(\textit{trace length of } h_i) \leq F(\textit{trace length of } h)$ for some function F

then $r \mid \frac{h}{q} \mid s$.

3. Proof of incompleteness

We now give a simple example to show that the proof systems in [1, 2] are incomplete. We first show that the proof system in [1] is incomplete.

Let process *id* have one input port and one output port. Process *id* iteratively reads values from its input port and writes them on its output port. Let id_1 and id_2 be two copies of *id*. Let H be a network formed by running id_1 and id_2 in parallel, as shown in Figure 1.

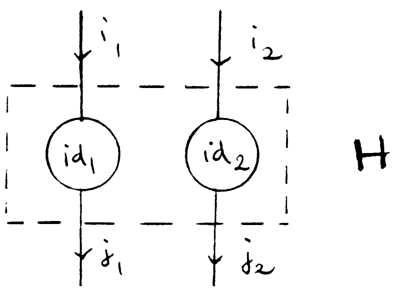

Figure 1

Let r |H |s be a specification of H, where

$r \equiv i_1=i_2=j_1=j_2=[\ ]$	(i.e. the trace is empty)
$s \equiv (i_1=i_2=j_1=j_2=[\ ])$	(i.e. the trace is empty
$\quad \vee\ (\lvert i_1\rvert=1 \wedge (i_2=j_1=j_2=[\ ]))$	or consists of one input value at i_1
$\quad \vee\ (\lvert i_2\rvert=1 \wedge (i_1=j_1=j_2=[\ ]))$	or consists of one input value at i_2)

Here, $[a_1, \ldots, a_m]$ denotes the sequence with elements $a_1, \ldots, a_m$, and | i| denotes the length of the sequence i.

It is clear that r |H |s is a valid specification. However, it is not provable in the proof system.

Suppose, on the contrary that, r |H |s is provable. Then there exist r_1, s_1, r_2, s_2 such that

(i) $r_1 |id_1| s_1$, $r_2 |id_2| s_2$ are true

(ii) $s_1 \wedge s_2 \Rightarrow s$

(iii) $s_1 \wedge s_2 \wedge r \Rightarrow r_1 \wedge r_2$

This implies that either $s_1 \equiv$ *trace is empty* or $s_2 \equiv$ *trace is empty*, since s means that the trace has at most one (input) value. (s_1 cannot refer to i_2, j_2. Similarly, s_2 cannot refer to i_1, j_1. If s_t says (among other things) that under some condition q_t the trace has at least one element then $s_1 \wedge s_2$ says (among other things) that under the condition $q_1 \wedge q_2$ the trace has at least two elements. Clearly, this does not imply s.)

Neither s_1 nor s_2 can be F (false) because they hold for the empty trace.

But if $s_1 \equiv$ *trace is empty*, what can r_1 be? r_1 can only be F, by the interpretation of the specification. So $r_1 \wedge r_2 \equiv F$. From (iii), it follows that $s_1 \wedge s_2 \wedge r \Rightarrow F$. $s_1 \wedge s_2 \wedge r$ cannot be F, since s_1, s_2, r all hold for the empty trace. This leads to a contradiction.

A similar argument holds for the case $s_2 \equiv$ *trace is empty*. So the proof system of [1] is incomplete.

The same example is used to show that the proof system of [2] is incomplete. r $\left| \frac{H}{True} \right|$s is true but not provable in the system (where r, H, s are as above). For, if it is provable then by comparing the two proof rules, it is easy to see that r |H |s is provable in the system of [1], which is a contradiction.

The problem with these proof systems seems to be that an assertion on the whole network cannot be decomposed into a conjunction of assertions on the component processes in general.

One way to make the proof system of [1] complete is to add the rule

If True |h |s then r |h |s $\wedge$ (the trace minus its last element satisfies r)

Clearly, the above rule is valid. Here is a sketch of the proof of its completeness:

Let there be a set of *primitive* processes $\{P_i\}$ together with their *precise* specifications $True \mid P_i \mid s_i$. (A specification r |h |s is precise if s is the strongest postcondition with respect to r, i.e. if r | h | s' then s implies s'.) Let H be a network formed from primitive processes P_i, i = 1,..., n and let r |H |s be a valid specification. Then

True |H $|\wedge_i s_i$ is valid and is precise.

So r |H $|\wedge_i s_i \wedge$ (the trace minus its last element satisfies r) is also valid and precise, as can be verified using the model of traces in [1]. This implies that the postcondition of the last specification implies s. So r |H |s is provable in the system. The system is complete.

Unfortunately, the expression "the trace minus its last element" is complicated and difficult to express formally, since one cannot refer to the whole trace in the assertions, i.e. variables are not allowed to range over traces. The only way to refer to a trace is through port variables and the "precedes" relation on the elements of the trace. This makes the proof rule look artificial.

Acknowledgement

I wish to thank Professor David Gries for a careful reading of the manuscript and Professor Susan Owicki for valuable discussions.

References

[1] Misra, J., and Chandy, K.M. Proofs of networks of processes, IEEE Trans. Soft. Eng. SE-7, July 1981, 417-426.

[2] Misra, J., Chandy, K.M. and Smith, T. Proving safety and liveness of communicating processes with examples, SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Aug 1982, 201-208.