

## UNBOUNDEDNESS DETECTION FOR A CLASS OF COMMUNICATING FINITE-STATE MACHINES

Yao-Tin YU and Mohamed G. GOUDA

*Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712, U.S.A.*

Communicated by David Gries

Received 8 February 1983

Revised 29 July 1983

Let  $M$  and  $N$  be two communicating finite-state machines that exchange one type of message. We discuss an algorithm to decide whether the communication between  $M$  and  $N$  is bounded. The algorithm is based on constructing a finite representation of the reachability tree of  $M$  and  $N$  assuming that  $M$  and  $N$  progress at equal speeds.

**Keywords:** Bounded communication, communication protocols, communicating machines, deadlock detection, finite-state machines, unboundedness detection

### 1. Introduction

Let  $M$  and  $N$  be two communicating finite-state machines that exchange one type of message over two unbounded, one-directional, FIFO channels. Informally, the communication between  $M$  and  $N$  is bounded iff there is a nonnegative integer  $K$  such that at each 'reachable state' of  $M$  and  $N$ , the number of messages in each channel is less than  $K$ . (Formal definitions are given later.) Cunha and Maibaum [3] have discussed an algorithm to decide whether the communication between  $M$  and  $N$  is bounded. Their algorithm consists of constructing a finite representation  $U$  of the 'reachability tree' of  $M$  and  $N$  and then scanning  $U$  to detect unboundedness, if any. To ensure that all reachable states are represented in  $U$ , the algorithm takes into account all possible relative speeds of  $M$  and  $N$ . Here, we present a more efficient algorithm, which constructs a finite representation  $T$  of the reachability tree assuming that  $M$  and  $N$  progress at *equal* speeds. Thus, the total number of states generated by our algorithm is, in most instances, less than those generated by the Cunha–Maibaum algorithm.

There are two practical reasons for considering this problem:

(i) Many self-timing VLSI arrays can be modeled as arrays of communicating machines in which each two machines exchange, at most, one type of message. As shown in [4], if the communication between each pair is bounded, then the communication within the array is bounded. Thus, our algorithm can be used to prove efficiently that the communication within a VLSI array is bounded.

(ii) Many communication protocols can be modeled as two communicating finite machines that exchange *many types* of messages [1,2,5,6,7]. Let  $\bar{M}$  and  $\bar{N}$  be two such machines. As shown in [9], if  $\bar{M}$  and  $\bar{N}$  are abstracted by two machines  $M$  and  $N$  that exchange one type of message and if the communication between  $M$  and  $N$  is shown to be bounded, then the communication between  $\bar{M}$  and  $\bar{N}$  is also bounded. Thus, our algorithm can be used to prove efficiently that the communication in a protocol is bounded.

Although the algorithm in this paper and the one in [8] are based on similar concepts, they differ in two respects. First, the algorithm in [8] is proposed to detect communication deadlocks

(rather than unboundedness) for a special class of the machines considered in the current paper. Second, the algorithm in [8] uses another finite representation of the reachability tree, that is more 'efficient' than the one in the current paper.

## 2. Communicating machines

A *communicating machine*  $M$  is a directed labeled graph with two types of edges, namely *sending* and *receiving* edges. A sending (receiving) edge is labeled 'send' ('receive'). One of the nodes in  $M$  is identified as its *initial node*; and each node in  $M$  is reachable by a directed path from its initial node. For convenience, we assume that each node has at least one output edge.

Let  $M$  and  $N$  be two communicating finite-state machines. A *state* of  $M$  and  $N$  is a four-tuple  $[v, w, x, y]$  where  $v$  and  $w$  are two nodes in  $M$  and  $N$  respectively and  $x$  and  $y$  are two nonnegative integers. Informally, a state  $[v, w, x, y]$  means that execution of  $M$  has reached node  $v$ , execution of  $N$  has reached node  $w$ , and the input channels of  $M$  and  $N$  store  $x$  and  $y$  messages respectively.

The *initial state* of  $M$  and  $N$  is  $[v_0, w_0, 0, 0]$  where  $v_0$  and  $w_0$  are the initial nodes of  $M$  and  $N$  respectively.

Let  $s = [v, w, x, y]$  be a state of  $M$  and  $N$ , and let  $e$  be an output edge of node  $v$  or  $w$ . A state  $s'$  is said to *follow*  $s$  over  $e$ , denoted by  $s \xrightarrow{e} s'$ , iff the following four conditions are satisfied:

- (i) If  $e$  is a sending edge from  $v$  to  $v'$  in  $M$ , then  $s' = [v', w, x, y + 1]$ .
- (ii) If  $e$  is a sending edge from  $w$  to  $w'$  in  $N$ , then  $s' = [v, w', x + 1, y]$ .
- (iii) If  $e$  is a receiving edge from  $v$  to  $v'$  in  $M$ , then  $x \geq 1$  and  $s' = [v', w, x - 1, y]$ .
- (iv) If  $e$  is a receiving edge from  $w$  to  $w'$  in  $N$ , then  $y \geq 1$  and  $s' = [v, w', x, y - 1]$ .

If  $s \xrightarrow{e} s'$  for some edge in  $M$  or  $N$ , then  $s'$  is said to *follow*  $s$ .

Let  $s$  and  $s'$  be two states of  $M$  and  $N$ , and let  $\langle e_1, \dots, e_r \rangle$  be a sequence of edges in  $M$  or  $N$ .  $s'$  is *reachable from*  $s$  over  $\langle e_1, e_2, \dots, e_r \rangle$  iff there are states  $s_0, s_1, \dots, s_r$  of  $M$  and  $N$  such that  $s_0 = s$ ,  $s_r = s'$  and  $s_i \xrightarrow{e_{i+1}} s_{i+1}$  for  $i = 0, \dots, r - 1$ .

Let  $s$  and  $s'$  be two states of  $M$  and  $N$ .  $s'$  is

*reachable from*  $s$  iff either  $s = s'$  or there is a sequence  $\langle e_1, \dots, e_r \rangle$  of edges in  $M$  or  $N$  such that  $s'$  is reachable from  $s$  over  $\langle e_1, \dots, e_r \rangle$ .

A state  $s$  of  $M$  and  $N$  is *reachable* iff it is reachable from the initial state of  $M$  and  $N$ .

The communication between  $M$  and  $N$  is *bounded* iff there is a positive integer  $K$  such that for each reachable state  $[v, w, x, y]$  of  $M$  and  $N$ ,  $x \leq K$  and  $y \leq K$ . Otherwise, the communication between  $M$  and  $N$  is *unbounded*. A proof of the following lemma is given in Appendix A.

**Lemma 2.1.** *The communication between  $M$  and  $N$  is unbounded iff there are two reachable states  $s_0 = [v, w, x, y]$  and  $s_1 = [v, w, x + i, y + j]$  of  $M$  and  $N$  such that  $s_1$  is reachable from  $s_0$  and either ( $i \geq 0$  and  $j > 0$ ) or ( $i > 0$  and  $j \geq 0$ ).*

Based on this lemma, an efficient algorithm to detect unboundedness for two communicating machines is presented in Section 4. This algorithm is also based on the concept of fair reachability discussed next.

## 3. Fair reachability

A state  $[v, w, x, y]$  of  $M$  and  $N$  is *fair* iff  $x = y$ . The initial global state of  $M$  and  $N$  is fair.

Let  $s$  and  $s'$  be two fair states of  $M$  and  $N$  and let  $e$  and  $f$  be two edges in  $M$  and  $N$  respectively.  $s'$  *fairly follows*  $s$  over  $e$  and  $f$  iff there exists a state  $s''$  such that either ( $s \xrightarrow{e} s''$  and  $s'' \xrightarrow{f} s'$ ) or ( $s \xrightarrow{f} s''$  and  $s'' \xrightarrow{e} s'$ ).  $s'$  *fairly follows*  $s$  iff  $s'$  fairly follows  $s$  over some edges  $e$  and  $f$  in  $M$  and  $N$  respectively.

Let  $s$  and  $s'$  be two fair states of  $M$  and  $N$ ; let  $P$  be a directed path of edges  $e_1, \dots, e_r$  in  $M$ , and  $Q$  be a directed path of edges  $f_1, \dots, f_r$  in  $N$ .  $s'$  is *fairly reachable from*  $s$  over the edges of  $P$  and  $Q$  iff there exist fair states  $s_0, s_1, \dots, s_r$  such that  $s = s_0$ ,  $s' = s_r$  and  $s_{i+1}$  fairly follows  $s_i$  over  $e_{i+1}$  and  $f_{i+1}$ ,  $i = 0, \dots, r - 1$ .  $s'$  is *fairly reachable from*  $s$  iff  $s'$  is fairly reachable from  $s$  over the edges of some two directed paths  $P$  and  $Q$  in  $M$  and  $N$  respectively.

A fair state of  $M$  and  $N$  is *fairly reachable* iff it is fairly reachable from the initial state of  $M$  and  $N$ .

Lemma 2.1 gives a necessary and sufficient condition for unbounded communication between two machines. In the next lemma, whose proof is given in Appendix A, we show that this condition can be equivalently stated in terms of two other conditions (A) and (B).

**Lemma 3.1.** *There are two reachable states  $s_0 = [v, w, x, y]$  and  $s_1 = [v, w, x + i, y + j]$ , of M and N such that  $s_1$  is reachable from  $s_0$  and either ( $i \geq 0$  and  $j > 0$ ) or ( $i > 0$  and  $j \geq 0$ ) iff one of the following two conditions is satisfied:*

(A) *There is a reachable state  $s' = [v', w', x', y']$  of M and N such that either node  $v'$  is in a directed cycle of all sending edges in M or node  $w'$  is in a directed cycle of all sending edges in N.*

(B) *There are two fairly reachable states  $s'_0 = [v', w', x', x']$  and  $s'_1 = [v', w', x' + k, x' + k]$  of M and N such that  $s'_1$  is fairly reachable from  $s'_0$  and  $k > 0$ .*

From Lemmas 2.1 and 3.1, the communication between M and N is unbounded iff either condition (A) or condition (B) is satisfied. Condition (B) is stated in terms of fair states and fair reachability, but condition (A) is not. In the next lemma, whose proof is given Appendix A, we show that condition (A) can also be stated in terms of fair states and fair reachability.

**Lemma 3.2.** *A state  $s = [v, w, x, y]$  of M and N is reachable iff a fair state  $s' = [v, w', x', y']$  of M and N where  $x' = y'$  is fairly reachable. (Similarly, a state  $s = [v, w, x, y]$  of M and N is reachable iff a fair state  $s' = [v', w, x', y']$  of M and N where  $x' = y'$  is fairly reachable.)*

The next theorem immediately follows from Lemmas 2.1, 3.1 and 3.2.

**Theorem 3.3.** *The communication between M and N is unbounded iff one of the following two conditions is satisfied:*

(A) *There is a fairly reachable state  $s = [v, w, x, x]$  of M and N such that either node  $v$  is in a directed cycle of all sending edges in M or node  $w$  is in a directed cycle of all sending edges in N.*

(B) *There are two fairly reachable states  $s_0 =$*

*$[v, w, x, x]$  and  $s_1 = [v, w, x + i, x + i]$  of M and N such that  $s_1$  is fairly reachable from  $s_0$  and  $i > 0$ .*

#### 4. Unboundedness detection algorithm

Based on Theorem 3.3, the following algorithm can be used to decide whether the communication between two machines is bounded.

##### Algorithm 4.1

**Input:** Two communicating machines M and N that exchange one type of message.

**Output:** A decision of whether the communication between M and N is bounded.

**Variable:** A directed rooted tree T whose nodes are labelled with fair states of M and N and whose directed edges correspond to the fairly-follow relation. Initially, T has exactly one node labelled with the initial state of M and N.

**Step 1. while** T has a leaf node n labelled with a state s such that there is at least one state which fairly follows s, and no non-leaf node in T is labelled with the same state s

**do if**  $s = [v, w, x, x]$  is such that one of the following two conditions is satisfied:

- (a) Either node v is in a directed cycle of all sending nodes in M, or node w is in a directed cycle of all sending nodes in N.
- (b) Node n in T has an ancestor node labelled with a state  $[v, w, y, y]$  where  $y < x$

**then stop:** The communication between M and N is unbounded

**else** find all the states  $s_1, \dots, s_r$  that fairly follow s;

add an equal number of nodes  $n_1, \dots, n_r$  to T;

label each new node  $n_i$  with the state  $s_i$ ,  $i = 1, \dots, r$ ;

add a directed edge from node n to each new node  $n_i$ ,  $i = 1, \dots, r$ .

**Step 2. stop:** The communication between M and N is bounded.

**Example 4.2.** Consider the two communicating machines M and N in Fig. 1(a) and (b) respectively. Tree T in Fig. 1(c) is constructed by apply-

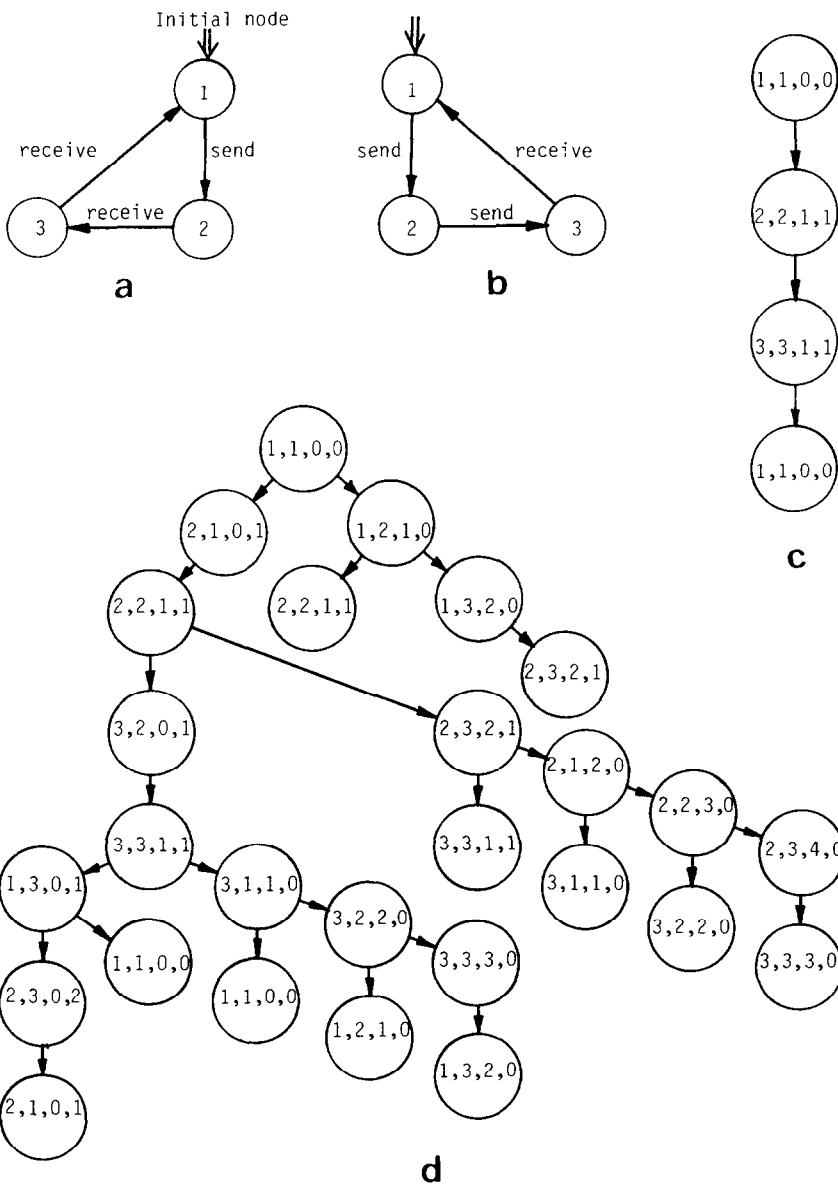


Fig. 1. An example. (a) M. (b) N. (c) T Constructed by Algorithm 4.1. (d) U constructed by the Cunha–Maibaum algorithm.

ing Algorithm 4.1 to M and N. From T, the communication between M and N is bounded. This same result can be obtained from tree U in Fig. 1(d), constructed by the Cunha–Maibaum algorithm [3]. Clearly, T with 4 nodes is better than U with 27 nodes.

### Acknowledgment

We are grateful to Prof. Gries and an anonymous referee for their careful reading of the manuscript and their suggestions for improvement of the paper.

## Appendix A. Proofs of lemmas

**Proof of Lemma 2.1. 'If' part.** Assume that there are two reachable states  $s_0 = [v, w, x, y]$  and  $s_1 = [v, w, x + i, y + j]$  of  $M$  and  $N$  such that  $s_1$  is reachable from  $s_0$  and  $i \geq 0$  and  $j > 0$ . (The proof for the case  $i > 0$  and  $j \geq 0$  is similar.) Assume that  $s_1$  is reachable from  $s_0$  over a sequence of edges  $e_1, e_2, \dots, e_r$ ; these edges form a directed cycle which starts and ends at node  $v$  in  $M$ , and possibly a directed cycle which starts and ends at node  $w$  in  $N$ . Therefore, the state  $s_2 = [v, w, x + 2i, y + 2j]$  of  $M$  and  $N$  is reachable from  $s_1$  over the same sequence of edges  $e_1, e_2, \dots, e_r$ . In general, the state  $s_k = [v, w, x + ki, y + kj]$ ,  $k = 2, 3, 4, \dots$ , is reachable (from  $s_1$ ). Since  $i > 0$ ,  $x + ki$  can be made larger than any given positive integer by selecting  $k$  large enough. Therefore, the communication between  $M$  and  $N$  is unbounded.

**'Only if' part.** Suppose that, for every two reachable states  $s_0 = [v, w, x, y]$  and  $s_1 = [v, w, x + i, y + j]$  of  $M$  and  $N$ , if  $s_1$  is reachable from  $s_0$ , then either  $i < 0$  or  $j < 0$ . Then, the number of reachable states is finite, hence the communication is bounded.  $\square$

**Proof of Lemma 3.1. 'If' part.** There are two cases to consider.

**Case 1. Condition (A) is satisfied.** Assume that there is a reachable state  $s' = [v', w', x', y']$  of  $M$  and  $N$  where node  $v'$  is in some directed cycle of all sending edges in  $M$ . Assume that this cycle has  $j$  sending edges; then the state  $s'_1 = [v', w', x' + 0, y' + j]$  is reachable from  $s'_0$ . The two reachable states  $s'$  and  $s'_1$  satisfy the required condition. (A similar argument can prove the case where node  $w'$  is in some directed cycle of all sending edges in  $N$ .)

**Case 2. Condition (B) is satisfied.** The two (fairly) reachable states  $s'_0$  and  $s'_1$  satisfy the required condition.

**'Only if' part.** Assume that there are two reachable states  $s_0 = [v, w, x, y]$  and  $s_1 = [v, w, x + i, y + j]$  of  $M$  and  $N$  such that  $s_1$  is reachable from  $s_0$  and  $i \geq 0$  and  $j > 0$ . (The proof for the case  $i > 0$  and  $j \geq 0$  is similar.) Assume also that  $s_1$  is reachable from  $s_0$  over a sequence of edges  $e_1, e_2, \dots, e_r$ . These edges form a directed cycle  $C_M$  which starts and ends with node  $v$  in  $M$  and possibly a directed

cycle  $C_N$  which starts and ends with node  $w$  in  $N$ . There are two cases to consider.

**Case 1.** If  $C_M$  is a cycle of all sending edges in  $M$  or  $C_N$  is a cycle of all sending edges in  $N$ , then condition (A) is satisfied.

**Case 2.** Otherwise, each of  $C_M$  and  $C_N$  is a directed cycle that contains at least one receiving edge. Let  $s_M$  and  $r_M$  be the numbers of sending and receiving edges (respectively) in  $C_M$ . Similarly, let  $s_N$  and  $r_N$  be the numbers of sending and receiving edges (respectively) in  $C_N$ . Since  $i \geq 0$  and  $j > 0$ , we have  $s_N \geq r_M$  and  $s_M > r_N$ . Also, let  $s'_0 = [v', w', z, z]$  be a fairly reachable state of  $M$  and  $N$  where nodes  $v'$  and  $w'$  are in cycles  $C_M$  and  $C_N$  respectively. Define the directed cycle  $D_M$  which starts and ends with node  $v'$  in  $M$  and which consists of cycle  $C_M$  repeated  $(s_N + r_N)$  times; similarly define the directed cycle  $D_N$  which starts and ends with node  $w'$  in  $N$  and which consists of cycle  $C_N$  repeated  $(s_M + r_M)$  times. Each of  $D_M$  and  $D_N$  has  $(s_M + r_M)(s_N + r_N)$  directed edges;  $D_M$  has  $s_M(s_N + r_N)$  sending edges and  $D_N$  has  $r_N(s_M + r_M)$  receiving edges. Therefore, there is a fairly reachable state  $s'_1 = [v', w', z + k, z + k]$  of  $M$  and  $N$  which is fairly reachable from  $s'_0$  over the edges in  $D_M$  and  $D_N$  where

$$\begin{aligned} k &= s_M(s_N + r_N) - r_N(s_M + r_M) \\ &= s_M s_N - r_M r_N > 0. \end{aligned}$$

Thus, the two states  $s'_0$  and  $s'_1$  satisfy condition (B).  $\square$

**Proof of Lemma 3.2. 'If' part.** Since  $s'$  is fairly reachable, it is reachable.

**'Only if' part.** Assume that  $s = [v, w, x, y]$  is reachable; i.e.,  $s$  is reachable from the initial state  $s_0 = [v_0, w_0, E, E]$  of  $M$  and  $N$  over a sequence of edges  $e_1, e_2, \dots, e_r$ . These edges form a directed path  $P$  from  $v_0$  to  $v$  in  $M$  and a directed path  $Q$  from  $w_0$  to  $w$  in  $N$ . Let  $|P|$  and  $|Q|$  denote the number of edges in paths  $P$  and  $Q$  respectively. There are three cases to consider:

**Case 1.  $|P| = |Q|$ :** In this case,

$$\begin{aligned} x &= \text{number of sending edges in } Q \\ &\quad - \text{number of receiving edges in } P \\ &= |Q| - \text{number of receiving edges in } Q \\ &\quad - |P| + \text{number of sending edges in } P \end{aligned}$$

= number of sending edges in P  
 – number of receiving edges in Q  
 = y.

*Case 2.*  $|P| < |Q|$ : Consider the proper prefix  $Q'$  of  $Q$  such that  $|P| = |Q'|$ . The state  $s' = [v, w', x', y']$ , reachable from the initial state of M and N over the edges of P and  $Q'$ , is fair (i.e.,  $x' = y'$ ), and the lemma is true.

*Case 3.*  $|P| > |Q|$ : Extend the directed path  $Q$  in any possible way in N until the extended path  $Q'$  is such that  $|P| = |Q'|$ . The state  $s' = [v, w', x', y']$ , reachable from the initial state of M and N over the edges of P and  $Q'$ , is fair (i.e.,  $x' = y'$ ), and the lemma is true.  $\square$

## References

- [1] G.V. Bochmann, Finite state description of communication protocols, *Computer Networks* 2 (1978) 361–372.
- [2] D. Brand and P. Zafiropulo, On communicating finite-state machines, IBM Res. Rept. RZ 1053 (#37725) 1981; J. ACM, to appear.
- [3] P. Cunha and T. Maibaum, A synchronization calculus for message-oriented programming, *Proc. 2nd Internat. Conf. on Distributed Computing Systems* (1981) pp. 433–445.
- [4] M.G. Gouda, Systems of communicating machines that are bounded and deadlock-free, Tech. Rept. 199, Dept. of Computer Sciences, Univ. of Texas at Austin, 1982.
- [5] M.G. Gouda and Y.T. Yu, Protocol validation by maximal progress state exploration, *IEEE Trans. Comm.* (1983) to appear.
- [6] J. Rubin and C.H. West, An improved protocol validation technique, *Computer Networks* 6 (1982) 65–73.
- [7] C.A. Sunshine, Formal modeling of communication protocols, Res. Rept. 81–89, USC/Inform. Sci. Institute, 1981.
- [8] Y.T. Yu and M.G. Gouda, Deadlock detection for a class of communicating finite state machines, *IEEE Trans. Comm.* COM-30 (12) (1982) 2514–2518.
- [9] Y.T. Yu, Communicating finite state machines: Analysis and synthesis of communication protocols, Ph.D. Thesis, Dept. of Computer Sciences, Univ. of Texas at Austin, 1983.