Protocol Validation by Fair Progress State Exploration

M.G. Gouda and Ji-Yun Han
Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712, USA

Consider a network of two communicating finite state machines that exchange messages over two one-directional, unbounded, FIFO channels. The fair reachability graph of such a network is a directed graph whose vertices correspond to global states (of the network) that are reachable by forcing the two machines in the network to progress in equal speeds. It is shown earlier that if the fair reachability graph of a given network is finite, then it can be used to decide whether the communication of this network is free from deadlocks and unspecified receptions. In this paper, we complement this result by showing that if the fair reachability graph of a given network is finite, then it can be used to decide whether the communication of this network is bounded. Moreover, if the communication is found to be bounded, then the finite fair reachability graph can be also used to compute the smallest possible capacities for the two channels in the network.

Computer network protocols, communicating machines, finite state machines, fair reachability graphs, bounded communication, computing channel capacities, decision of node reachability.

1. Introduction

Many communication protocols can be modeled as a network of two communicating finite state machines that exchange messages over two one-directional, unbounded FIFO channels [1,6,9]. It is useful to validate these protocol models by showing that their communications satisfy certain desirable properties such as boundedness, and freedom from deadlocks and unspecified receptions [2]. (Definitions of these properties are given in Section 2.)

The most straightforward technique to validate a given network of two communicating finite state machines is called *state exploration* [9]. In this technique, the *reachability graph* of the given network is constructed; it is a directed graph whose vertices correspond to the reachable states of the network, and whose arcs correspond to its state transitions. If the reachability graph is found to be finite, then the communication of the network is determined to be bounded, and its freedom from deadlocks and unspecified receptions can be decided. Moreover, the smallest possible capacities for the two channels in the network can be computed. Unfortunately, this technique suffers from a major drawback: Since most reachability graphs are "huge," constructing them requires "large" execution time and memory. The problem arises, because a reachability graph exhibits the progress of its two machines by allowing *all* their relative progress speeds.

This last observation has led Rubin and West

Mohamed G. Gouda received the B.Sc. degrees in engineering and mathematics in 1968 and 1971, respectively, from Cairo University, Cairo, Egypt, the M.A. degree in mathematics in 1972 from York University, Toronto, Ont., Canada, and the M. Math and Ph.D. degrees in computer science in 1973 and 1977, respectively, from the University of Waterloo, Waterloo, Ont. From 1977 to 1980 he worked for the Honeywell Systems and Research Center and the Honeywell Corporate Technology Center, Minneapolis, MN. Since 1980 he has been an Assistant Professor in the Department of Computer Sciences at the University of Texas at Austin, His research interests include formal verification and synthesis of distributed systems and communication protocol.

Ji-Yun Han graduated from the University of Science and Technology of China, and is currently working toward his Ph.D. in the Department of Computer Sciences at the University of Texas at Austin. He was an engineer with the Karst Research Institute of China, and is currently working for the Burroughs Corp., Austin Research Center. His research interests include computer networks, communication protocols and distributed systems.

North-Holland
Computer Networks and ISDN Systems 9 (1985) 353–361

[5] to suggest an improved protocol validation technique that uses a special type of reachability graph, called *fair reachability graphs* in the current paper. In this technique, the fair reachability graph of the given network is constructed by forcing the two machines to progress in equal speeds. (Indeed, the resulting fair reachability graph is usually much smaller than its corresponding reachability graph.) Rubin and West argued [5] that if the constructed fair reachability graph is found to be finite, then it can be used to decide whether the communication is free from deadlocks and unspecified receptions. However, they left open the problem of deciding boundedness.

In this paper, we close this open problem. In particular, we present an algorithm that uses the finite fair reachability graph of a given network to decide whether its communication is bounded. We also present another algorithm (based also on finite fair reachability graphs) to compute the smallest possible capacities for the two channels of a network whose communication is bounded.

2. Networks of Communicating Finite State Machines

A *communicating finite state machine* M is a labeled directed graph with two types of edges, namely *sending* and *receiving edges.* A sending (or receiving) edge is labeled $-g$ (or $+g$, respectively) for some *message* g in a finite set G of messages. For convenience, we assume that each node in M has at least one outgoing edge. A node in M whose outgoing edges are all sending (or all receiving) edges is called a *sending* (or *receiving*) *node.* A node in M whose outgoing edges include both sending and receiving edges is called a *mixed node.* One of the nodes in M is identified as its *initial node,* and each node in M is reachable by a directed path from the initial node.

Let M and N be two communicating finite state machines with the same set G of messages. $[M, N]$ denotes the network consisting of machines M and N connected by two FIFO channels in opposite directions.

A *state* of a network $[M, N]$ 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 strings over the messages in G. Informally, a state $[v, w, x, y]$ means that the executions of M and N have reached nodes v and w respectively, while the input channels of M and N store the strings x and y respectively.

The *initial state* of a network $[M, N]$ is $[v_0, w_0, E, E]$ where v_0 and w_0 are the initial nodes in M and N respectively, and E is the empty string.

Let $s = [v, w, x, y]$ be a state of a network $[M, N]$; and let e be an outgoing edge of node v or w. A state s' is said to *follow s over e* iff one of the following four conditions is satisfied:

i. e is a sending edge, labeled $-g$, from v to v' in M, and $s' = [v', w, x, y.g]$, where "." is the concatenation operator.
ii. e is a sending edge, labeled $-g$, from w to w' in N, and $s' = [v, w', x.g, y]$.
iii. e is a receiving edge, labeled $+g$, from v to v' in M, and $s' = [v', w, x', y]$, where $x = g.x'$.
iv. e is a receiving edge, labeled $+g$, from w to w' in N, and $s' = [v, w', x, y']$, where $y = g.y'$.

Let s and s' be two states of a network $[M, N]$, s' *follows* s iff there is a directed edge e in M or N such that s' follows s over e.

Let s and s' be two states of a network $[M, N]$, s' is *reachable from* s iff $s = s'$ or there exist states $s_1, \ldots, s_r$ such that $s = s_1$, $s' = s_r$ and s_{i+1} follows s_i for $i = 1, \ldots, r-1$.

A state of a network $[M, N]$ is said to be *reachable* iff it is reachable from the initial state of $[M, N]$.

A state $[v, w, x, y]$ of a network $[M, N]$ is a *deadlock state* iff (i) both v and w are receiving nodes, and (ii) $x = y = E$ (the empty string). If no reachable state of network $[M, N]$ is a deadlock state, then the communication of $[M, N]$ is said to be *deadlock-free.*

A state $[v, w, x, y]$ of a network $[M, N]$ is an *unspecified reception state for M* iff $x = g_1.g_2 \ldots g_k$ $(k \geq 1)$, and v is a receiving node and none of its outgoing edges is labeled $+g_1$. A state $[v, w, x, y]$ is an *unspecified reception state for N* iff $y = g_1.g_2 \ldots g_k$ $(k \geq 1)$, and w is a receiving node and none of its outgoing edges is labeled $+g_1$. If no reachable state of $[M, N]$ is an unspecified reception state for M or N, then the communication of $[M, N]$ is said to be *free from unspecified receptions.*

The communication from M to N (or from N to M) in a network [M, N] is said to be bounded by K iff for every reachable state [v, w, x, y] of [M, N], $|y| \leq K$ ($|x| \leq K$), where $|y|$ is the number of

messages in string y. The communication of [M, N] is said to be bounded by K iff each of the communications from M to N, and from N to M is bounded by K. If the communication from M to N (or from N to M) is bounded by K but not by K-1, then K is called the *smallest possible capacity* for the input channel of N(M).

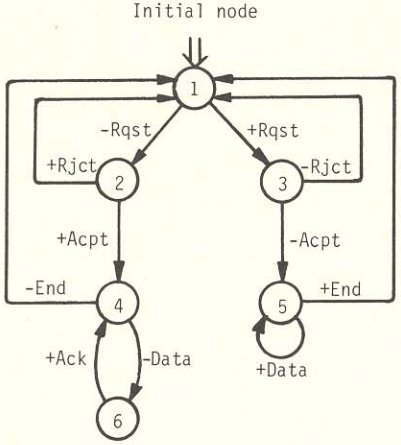
3. Fair Reachability Graph (FRGs)

A state $[v, w, x, y]$ of network $[M, N]$ is *fair* iff $|x| = |y|$, where $|x|$ is the number of messages in string x. Obviously, the initial state of $[M, N]$ is fair. Also, any deadlock state of $[M, N]$ is fair.

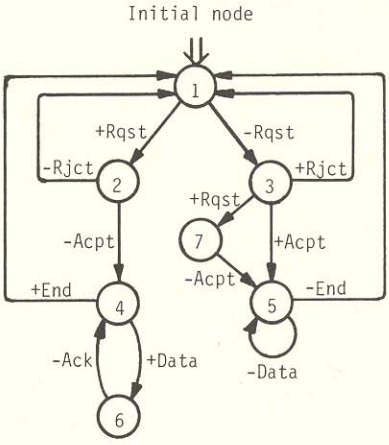
Let s_1 and s_2 be two fair states of $[M, N]$, and let e and f be two edges in M and N respectively. s_2 *fairly follows* s_1 *over* (e, f) iff there exists a state s such that *either* s follows s_1 over e and s_2 follows s over f, *or* s follows s_1 over f and s_2 follows s over e.

The *fairly reachable states* of a network $[M, N]$ are defined recursiveley as follows:

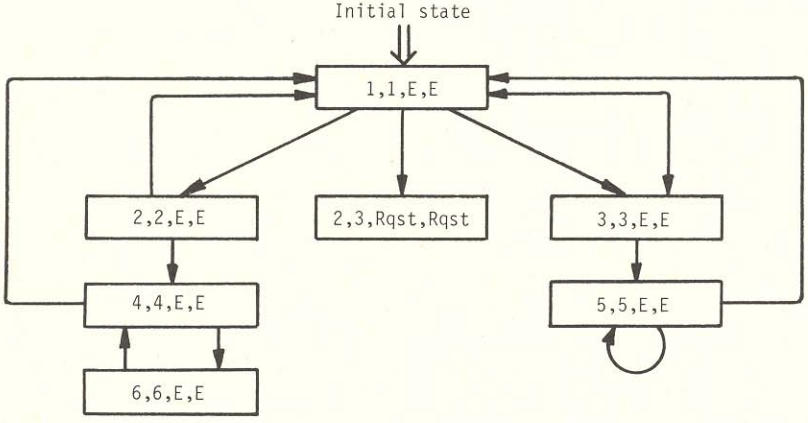
i. The initial state of $[M, N]$ is fairly reachable.
ii. If a state s is fairly reachable, and if a state s' fairly follows s over (e, f) for some edges e and f in M and N respectively, then s' is fairly reachable.
iii. No state, other than those defined by i and ii, is fairly reachable.



(a) M



(b) N



(c) The fair reachability graph of network [M,N].

Fig. 1. An Example.

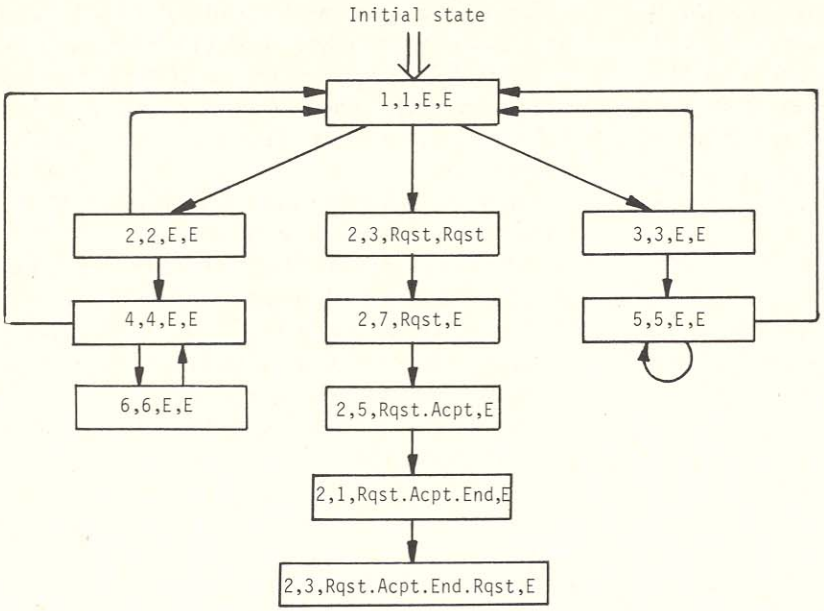


Fig. 1 (cont'd). (d) The augmented fiar reachability graph G_m^* of $[M, N]$ with respect to N.

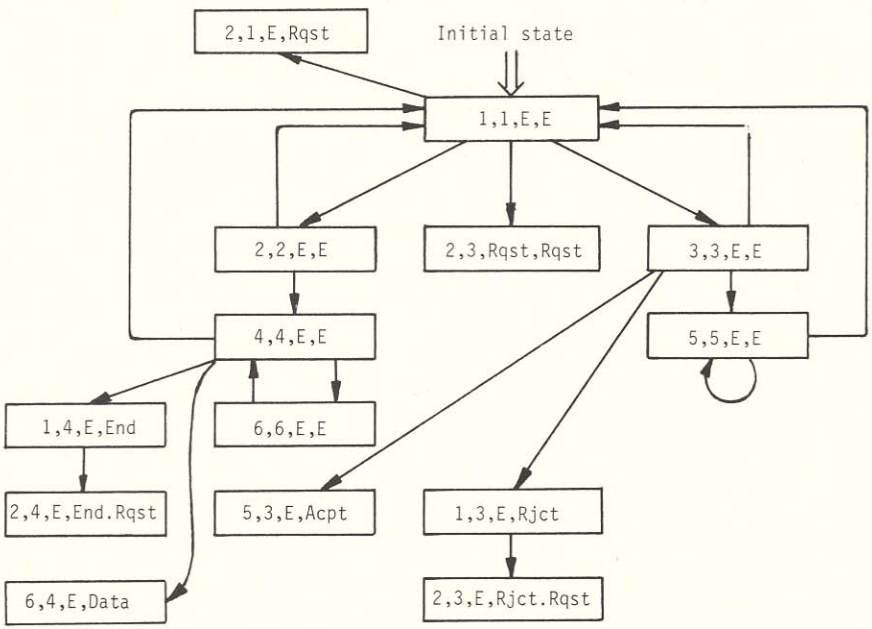


Fig. 1 (cont'd). (e) The extended fair reachability graph G_M^+ of $[M, N]$ with respect to M.

The fairly reachable states of a network $[M, N]$, and the relationship of "fairly follow" over them can be represented by a labeled directed graph G, called the *fair reachability graph* of $[M, N]$, as follows:

i. For each fairly reachable state s of $[M, N]$, add a vertex, labeled with state s, to G. (A vertex labeled with a state s in G is referred to as vertex s.)

ii. If two fairly reachable states s_1 and s_2 of $[M, N]$ are such that s_2 fairly follows s_1 over (e, f), then add an arc, labeled (e, f), from vertex s_1 to vertex s_2 in G.

(Notice that G has vertices and arcs to be dis-

tinguished from the nodes and directed edges of M or N.)

Rubin and West observed [5] that if the fair reachability graph G of a network $[M, N]$ is *finite* (i.e. has a finite number of vertices and arcs), then it can be used to decide whether the communication of $[M, N]$ is free from deadlocks and unspecified receptions. Their algorithm can be stated as follows:

i. The communication of $[M, N]$ is deadlock-free iff no vertex in G is labeled with a deadlock state.
ii. The communication of $[M, N]$ is free from unspecified receptions iff (a) no vertex in G is labeled with an unspecified reception state, and (b) no vertex in G is labeled with a state of the form $[v, w, E, E]$ that is "followed" by an unspecified reception state.

The termination of this algorithm follows from the fact that G is finite. (Later, Yu and Gouda [7] presented another algorithm that uses fair reachability to decide, in polynomial time, freedom of deadlocks for networks, where the two machines exchange one type of message.)

Example 1. Consider the network [M, N] whose communicating finite state machines M and N are shown in Figures 1a and 1b, respectively. The exchanged messages between the two machines have the following meanings;

Rqst	denotes a request to send data messages.
Acpt	denotes a positive acceptance of Rqst.
Rjct	denotes a rejection of the request Rqst.
Data	denotes a data message.
Ack	denotes an acknowledgement of receiving a data message.
End	denotes an end to transmitting data messages.

The fair reachability graph G for this network is shown in Fig. 1c. G has no deadlock states, but it has an unspecified reception state [2, 3, Rqst, Rqst] for M. Therefore, the communication of $[M, N]$ is free from deadlocks, but not from unspecified receptions. □

In their paper [5], Rubin and West contended that they couldn't extend their algorithm to decide boundedness. (In [8], Yu and Gouda presented an algorithm that uses fair reachability graphs to decide boundedness for networks, where the two machines exchange one type of message.) In this paper, we extend the Rubin and West algorithm to decide boundedness for any network whose fair reachability graph is finite. But before we discuss this algorithm, we need to discuss how to solve another problem using finite fair reachability graph. The solution to this other problem will be used later in our algorithm to decide boundedness.

4. Deciding Node Reachability Using FRGs

It is required to construct an algorithm to decide for any network $[M, N]$, whose fair reachability graph G is finite, and for any node v in machine M, whether v is reachable in $[M, N]$. In other words, decide whether there is a reachable state of $[M, N]$ of the form $[v, w, x, y]$, for some w, x, and y. (Obviously, an algorithm to decide whether a node w in N is reachable in $[M, N]$ is similar.)

The required algorithm uses an augmented version G^*_M of G; G^*_M is called the *augmented fair reachability graph* of $[M, N]$ *with respect* to M. It can be constructed from G by the following three-step procedure.

i. Initially, $G^*_M := G$.
ii. **For** every vertex s in G
  do if s′ is an unspecified reception state for N that follows s
    then add a vertex s′ to G^*_M;
    add an arc from vertex s to vertex s' in G^*_M.
iii. **For** every unspecified reception state s for N, in G^*_M
  do if there is a state $s' = [v', w', x', y']$ that follows s over an edge in M, and if G^*_M has no vertex s'' with a directed path to vertex s, where (a) $s'' = [v', w', x', y'']$, and (b) y'' is a prefix of y'.
  then add a vertex s′ to G^*_M; add an arc from vertex s to vertex s' in G^*_M.

Next, we sketch an argument that G^*_M resulting from the above algorithm is finite; this in turn implies that the algorithm terminates. Each vertex that is added to G^*_M in Step ii corresponds to an unspecified reception state for N that can be reached after M progresses over one edge from a state in G. Since G is finite, the number of vertices added to G^*_M in Step ii is finite. Moreover, every vertex that is added to G^*_M in Step iii corresponds to a state that is reached by the sole progress of M

while N remains in an unspecified reception state. This sole progress of M can either lead to a halting state (e.g. an unspecified reception state for M, or a state where M is at a receiving node and its input channel is empty) or lead to M reaching a directed cycle of all sending edges. However, the progress of M in Step iii is stopped before M completes a directed cycle of all sending edges. Hence, the number of vertices added to G^*_M in Step iii is finite, and the resulting G^*_M is finite.

The following theorem states a necessary and sufficient condition (based on G^*_M) for a node in M to be reachable in $[M, N]$.

Theorem 1. *Let [M, N] be a network whose augmented fair reachability graph G^*_M is finte, and let v be a node in M. Node v is reachable in [M, N] iff G^*_M has a vertex labeled with a state of the form [v, w, x, y], for some w, x, y.*

Proof. The if part is immediate; we sketch the proof for the only if part. Assume that node v in M is reachable in $[M, N]$, i.e. there exists a reachable state $[v, w, x, y]$ of $[M, N]$, for some w, x, y. Also assume that to reach this state, M and N progress over the directed paths P and Q. Let $|P|$ and $|Q|$ denote the number of directed edges in paths P and Q respectively. There are three cases to consider:

Case 1 ($|P| = |Q|$). In this case, $[v, w, x, y]$ is fairly reachable, and so it must label one of the vertices in G^*_M. Therefore, the condition is satisfied.

Case 2 ($|P| < |Q|$). In this case, there exists a proper prefix Q' of Q such that (a) $|P| = |Q'|$, and (b) as M and N progress over P and Q' (respectively), the network $[M, N]$ reaches a state $[v, w', x', y']$, for some w', x', y'. This state is fairly reachable, and so it must label one of the vertices in G^*_M. Therefore, the condition is satisfied.

Case 3 ($|P| > |Q|$). In this case, we try to extend the directed path Q in N into Q' such that (a) $|P| = |Q'|$, and (b) as M and N progress over P and Q' (respectively), the network $[M, N]$ reaches a state $[v, w', x', y']$, for some w', x', y'. There are two possible outcomes of this trial:

a. *Such an extension is possible:* In this case, the reached state $[v, w', x', y']$ is fairly reachable, and so must label one of the vertices in G^*_M. Therefore the condition is satisfied.

b. *Such an extension is impossible:* In this case, Q can be extended only to Q'' such that (a) $|P| > |Q''|$, and (b) as M and N progress over P and Q'' (respectively), the network reaches an unspecified reception state, for N, of the form $s'' = [v, w'', x'', y'']$.Let P'' be a proper prefix of P such that (a) either $|P''| = |Q''|$ or $|P''| = |Q''| + 1$, and (b) as M and N progress over P'' and Q'' (respectively), the network $[M, N]$ reaches an unspecified reception state s, for N. State s must label one of the vertices in G^*_M. State s'' is reachable from s. If s'' is reachable from s without M ever completing a directed cycle of all sending edges, then $[v, w'', x'', y'']$ must label one of the vertices in G^*_M and the condition is satisfied. If it is reachable from s only after M completes one or more directed cycles of all sending edges, then there is another state $s' = [v, w'', x'', y']$ that is reachable from s without M ever completing a directed cycle of all sending cycles. State s' must label one of the vertices in G^*_M, and the condition is satisfied. □

Theorem 1 suggests the following algorithm.

Algorithm 1. To decide whether a node v in a communicating finite state machine M is reachable in a network $[M, N]$:

i. Construct the augmented fair reachability graph G^*_M of $[M, N]$ with respect to M.
ii. **If** G^*_M has a vertex labeled with a state of the form $[v, w, x, y]$ for some w, x, and y
then v is reachable in $[M, N]$
else v is not reachable in $[M, N]$. □

A similar algorithm can use the augmented fair reachability graph G^*_N of $[M, N]$ to decide whether a node w in N is reachable in $[M, N]$.

Example 1 (continued). Consider the network $[M, N]$ in Figure 1, and assume that it is required to decide whether nodes 5 and 7 in machine N are reachable. Following an algorithm similar to Algorithm 1, we construct G^*_N as shown in Figure 1d. Since G^*_N has two vertices labeled with the states [5, 5, E, E] and [2, 7, Rqst, E], then both 5 and 7 in N are reachable. □

In the next section, we employ these algorithms to decide whether the communication of [M, N] is bounded.

5. Deciding Boundedness Using FRGs

It is required to construct an algorithm to decide for any network $[M, N]$, whose fair reachability graph G is finite, whether the communication from M to N is bounded. In other words, decide for any such network $[M, N]$ whether there exists a positive integer k such that for any reachable state $[v, w, x, y]$ of $[M, N]$, $|y| \leq k$. The required algorithm is based on the following theorem.

Theorem 2. *Let [M, N] be any network whose fair reachability graph G is finite. The communication from M to N is unbounded iff there exists a reachable state [v, w, x, y] of [M, N], where node v is in a directed cycle of all sending edges.*

Proof. The if part is immediate; we prove the only if part by contradiction. Assume that the communication from M to N is unbounded (i.e. for every positive integer k, there exists a reachable state $[v', w', x', y']$ such that $|y'| \geq k$), and that no reachable state $[v, w, x, y]$ of $[M, N]$ is such that v is in a directed cycle of all sending edges. We show that the fair reachability graph of $[M, N]$ is infinite, which contradicts the fact that G is finite.

To show that the fair reachability graph of $[M, N]$ is infinite, we show that for every positive integer k, there exists a reachable state $[v, w, x, y]$ of $[M, N]$ such that $|x| = |y| > k$.

Let k be any positive integer. Since the communication from M to N is unbounded, there exists a reachable state $[v, w, x, y]$ of $[M, N]$ such that $|y|$ is bigger than any positive integer. In particular, there exists a reachable state $[v, w, x, y]$ of $[M, N]$ such that $|y| > mk + m - 1$, where m is the number of nodes in machine M. Let P and Q be the two directed paths to be traversed by M and N (respectively) for the network $[M, N]$ to reach $[v, w, x, y]$. Let $|P|$ ($|Q|$) be the number of directed edges in path P (Q). There are three cases to consider: (We show that in each case the network can reach a state $[v', w', x', y']$, where $|x'| = |y'| > k$.)

Case 1 ($|P| = |Q|$). In this case, the reachable state $[v, w, x, y]$ is such that $|x| = |y| = mk + m - 1 > k$.

Case 2 ($|P| < |Q|$). In this case, there is a proper prefix Q' of Q such that (a) $|P| = |Q'|$, and (b) as M and N progress over P and Q', the network $[M, N]$ reaches a state $[v, w', x', y']$, where $|x'| = |y'| \geq |y| = km + m - 1 > k$.

Case 3 ($|P| > |Q|$). In this case, there is a proper prefix P' of P such that (a) $|P'| = |Q|$, and (b) as M and N progress over P' and Q, the network $[M, N]$ reaches a state $[v', w, x', y']$, where $|x'| = |y'|$ and $|y| - |y'| \leq (m-1)(|x'| - |x|) + m - 1$. (The last inequality is due to the fact that M cannot traverse a directed cycle of all sending edges as it progresses from P' to P.) Therefore, $|x'| = |y'| > k$. □

Theorem 2 suggests the following algorithm.

Algorithm 2. To decide whether the communication of a network $[M, N]$ whose fair reachability graph G is finite, is bounded:

i. **If** M has a directed cycle C whose edges are all sending, and if any node in C is decided (by Algorithm 1) to be reachable in $[M, N]$
   then the communication from M to N is unbounded
   else the communication from M to N is bounded
ii. **If** N has a directed cycle D whose edges are all sending, and if any node in D is decided to be reachable in $[M, N]$
   then the communication from N to M is unbounded
   else the communication from N to M is bounded. □

Example 1 (continued). Consider the network $[M, N]$ in Figure 1, and assume that it is required to decide whether its communication is bounded. From Theorem 2, the communication from M to N is bounded since M has no directed cycles of all sending edges. Following Algorithm 2, since the state [5, 5, E, E] is in G_N^* and since node 5 in N has a sending self loop, then the communication from N to M is unbounded. □

Once it is decided that the communication of a network [M, N] is bounded in one direction, i.e. from M to N or from N to M, the next question to ask is what is the smallest possible capacity for the channel in that direction. In the next section we show that this question can also be answered using finite fair reachability graphs.

6. Computing Channel Capacities Using FRGs

It is required to construct an algorithm to compute the smallest possible capacity for the output channel of M in any network $[M, N]$, whose fair reachability graph G is finite, and where the communication from M to N is decided (by Algorithm 2) to be bounded. In other words, compute the smallest possible integer k such that for any reachable state $[v, w, x, y]$ of $[M, N]$, $|y| \leq k$.

The required algorithm uses an extended version G_M^+ of G. G_M^+ is called the *extended fair reachability graph of* $[M, N]$ *with respect to* M. It can be constructed from G by the following two-step procedure.

i. Initially, $G_M^+ := G$.
ii. **For** every vertex s in G_M^+
do if there is a state s' that follows s over an edge in M
then add a vertex s' to G_M^+; add an arc from vertex s to vertex s' in G_M^+.

Next, we sketch an argument that G_M^+, resulting from the above procedure, is finite; this in turn implies that the procedure terminates. Every vertex that is added to G_M^+ in Step ii corresponds to a state that is reachable, by the sole progress of M, from a state in G (which is finite). Since the communication from M to N is bounded, then (by Theorem 2) M can never reach a directed cycle of all sending edges. Therefore, every sole progress of M in Step ii must lead to a halting state for M (i.e. either an unspecified reception state for M, or a state where M is at a receiving node and its input channel is empty). Hence, the number of vertices added to G_M^+ in Step ii is finite, and the resulting G_M^+ is finite.

The following theorem explains why a finite G_M^+ is useful in computing the smallest possible capacity for the output channel of M.

Theorem 3. *Let [M, N] be a network whose extended fair reachability graph* G_M^+ *is finite. For every reachable state [v, w, x, y] of [M, N], there exists a state [v', w', x', y'] that labels a vertex in* G_M^+ *such that* $|y'| \geq |y|$.

Proof. let $s = [v, w, x, y]$ be a reachable state of $[M, N]$, and let P and Q be the two directed paths to be traversed by M and N (respectively) for the network $[M, N]$ to reach s. Also let $|P|$ ($|Q|$) be the number of directed edges in path P (Q). There are three cases to consider: (We show that in each case, G_M^+ has a vertex labeled with a state $s' = [v', w', x', y']$ such that $|y'| \geq |y|$.)

Case 1 ($|P| = |Q|$)**:** In this case, state $s = [v, w, x, y]$ is fairly reachable, and so must label one vertex in G, and in G_M^+.

Case 2 ($|P| < |Q|$)**:** In this case, there is a proper prefix Q' of Q such that (a) $|P| = |Q'|$, and (b) as M and N progress over P and Q' (respectively), the network $[M, N]$ reaches a state $s' = [v, w', x', y']$, where $|x'| = |y'| \geq |y|$. State s' must label one vertex in G and in G_M^+.

Case 3 ($|P| > |Q|$)**:** In this case, there is a proper prefix P' of P such that (a) $|P'| = |Q|$, and (b) as M and N progress over P' and Q (respectively), the network $[M, N]$ reaches a state $s' = [v', w, x', y']$, where $|x'| = |y'|$. State s' must label one vertex in G, and so must label one vertex in G_M^+. Moreover, state s is reachable from s' by the sole progress of M; therefore state s' must label one vertex in G_M^+. □

Theorem 3 suggests the following algorithm.

Algoritm 3. To compute the smallest possible capacity k_M for the output channel of M in a network $[M, N]$, whose extended fair reachability graph G_M^+ is finite, and where the communication from M to N is decided to be bounded (by Algorithm 2):

i. Construct the extended fair reachability graph G_M^+ of $[M, N]$.
ii. Find the state $[v, w, x, y]$ that labels a vertex in G_M^+ such that for any state $[v', w', x', y']$ that labels a vertex in G_M^+, $|y| \geq |y'|$.
iii. $k_M := |y|$. □

A similar algorithm can use G_N^+ to compute the smallest possible capacity k_N for the output channel of N.

Example 1 (continued). Consider the network $[M, N]$ in Figure 1, and assume that it is required to compute the smallest possible capacity k_M for the channel from M to N. Following Algorithm 3, we construct G_M^+ as shown in Figure 1e. Since each state $[v, w, x, y]$ that labels a vertex in G_M^+ is such that $|y| \leq 2$, then $k_M = 2$. □

7. Concluding Remarks

We have presented three algorithms that decide the following three questions for any given network whose fair reachability graph is finite:

a. Whether a node in one of the two machines in the network is reachable,
b. whether the communication is bounded, and
c. whether the smallest possible capacity for one channel that can be bounded is k, for some given k.

Each of the three algorithms is based on the finite fair reachability graph of the given network or some augmented or extended version of it. Notice that if the given network is free from unspecified receptions, then the augmented versions of its fair reachability graph becomes identical to the fair reachability graph itself. In this case, the two algorithms to decide problems a and b above will use only the fair reachability graph of the given network.

In [3], Gouda, Chow, and Lam have considered the class of communicating finite state machine networks, where the communication is known to be bounded in one direction. They showed that the fair reachability graph of each network in this class is finite. This result along with the result of Rubin and West [5], and that in the current paper show that boundedness, and freedom of deadlocks and unspecified receptions can all be decided for this class of networks. This confirms the previous results concerning this class in [2] and [4], and provides different decidability algorithms to achieve these results.

Acknowledgement

We would like to thank Carl Sunshine and the referees for their careful reading of the manuscript and their comments.

References

[1] G.V. Bochmann, "Finite state description of communication protocols," *Computer Networks,* Vol. 2, pp. 361–371, 1978.

[2] D. Brand and P. Zafiropulo, "On communicating finite-state machines," *JACM,* Vol. 30, pp. 323–342, April 1983.

[3] M.G. Gouda, C.H. Chow, and S.S. Lam, "Livelock detection in networks of communicating finite state machines," Technical Report, TR-84-10, Dept. of Comp. Sciences, Univ. of Texas at Austin, March 1984. (Submitted for journal publication).

[4] M.G. Gouda, E.M. Gurari, T.H. Lai, and L.E. Rosier, "Deadlock detection in systems of communicating finite state machines," Technical Report, TR-84-11, Dept. of Comp. Sciences, Univ. of Texas at Austin, April 1984. (Submitted for journal publication.)

[5] J. Rubin and C.H. West, "An improved protocol validation technique," *Computer Networks,* Vol. 6, pp. 65–73, April 1982.

[6] C.A. Sunshine, "Formal techniques for protocol specification and verification," *Computer,* Vol. 12, No. 9, pp. 20–27, Sept. 1979.

[7] Y.T. Yu and M.G. Gouda, "Deadlock detection for a class of communicating finite state machines," *IEEE Trans. on Comm.,* Vol. COM-30, pp. 2514-2518, Dec. 1982.

[8] Y.T. Yu and M.G. Gouda, "Unbounded detection for a class of communicating finite state machines," *Information Processing Letters,* Vol. 17, pp. 235–240, Dec. 1983.

[9] P. Zafiropulo, C.H. West, H. Rudin, D.D. Cowan, and D. Brand, "Towards analyzing and synthesizing protocols," *IEEE Trans. on Comm.,* Vol. COM-28, pp. 651–661, April 1980.