ON THE PROGRESS OF COMMUNICATION BETWEEN TWO MACHINES

M. G. Gouda[1], E. G. Manning[2], Y. T. Yu[1]

TR-200 May 1982

DCS etc.

FIGS missing

[1]Department of Computer Sciences, University of Texas at Austin.

[2]Department of Computer Sciences, University of Waterloo.

put on prev. pg.

ABSTRACT

We consider the following problem concerning any two finite state machines M and N which exchange messages via two one-directional channels. "Is there a positive integer K such that M and N with K-capacity channels never reach a nonprogress state?" The problem is shown to be undecidable in general. For a reasonable class of communicating machines, the problem is shown to be decidable; and the decidability algorithm is polynomial. We also discuss some sufficient conditions for the problem to have a a positive answer; these sufficient conditions can be checked for the given M and N in polynomial time.

Keywords: Bounded communication, communicating finite state machine, communication deadlock, communication protocol, progress, unspecified reception.

1. INTRODUCTION

The model of communicating finite state machines is an abstraction of sequential processes which communicate exclusively by exchanging messages. The abstraction is achieved by suppressing the local data structures and internal operations of the processes, and representing each of them only by its sending and receiving operations with other processes. This abstract model has been useful in the specification [4, 8], analysis [1, 9, 10], and synthesis [6, 11] of communication protocols. But its major impact has been in characterizing some communication progress properties such as boundedness, and freedom of deadlocks and unspecified receptions [2, 11].

In this paper, we consider the general problem of communication progress between two machines, and discuss its relationship to the above progress properties. We also show that the problem is undecidable in general and present some special cases for which the problem is decidable by polynomial algorithms.

The paper is organized as follows. The communication progress problem is defined in Section II, and equivalent forms for the problem are presented in Section III. In Section IV, the problem is shown to be undecidable in general; and in Section V, it is shown to be decidable by a polynomial algorithm for a special class of communicating machines called alternating machines. In Section VI, we discuss a set of sufficient conditions to ensure that the problem has a positive answer; these conditions can be checked by a polynomial-time algorithm. A summary of the results is given in Section VII.

2. THE COMMUNICATION PROGRESS PROBLEM

A communicating machine M is a directed labelled graph with two types of nodes - sending and receiving nodes. 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.

Each node in M has at least one output edge. An output edge of a sending (or receiving) node is called a sending (or receiving) edge, and is labelled send(g) (or receive(g) respectively) for some message g from a finite set S of messages. No two outputs of the same node in M have identical labels.

Let M and N be two communicating machines with the same set S of messages and let K be a positive integer, K>0. A state of M and N with K-capacity channels is a four-tuple [m,n,x,y] where m and n are two nodes in M and N respectively and x and y are two strings of messages from the set S such that $|x| \leq K$ and $|y| \leq K$ where $|x|$ and $|y|$ are the numbers of messages in x and y respectively. Informally, a state [m,n,x,y] means that the execution of M has reached node m, and the execution of N has reached node n, while the input channel of M has the message sequence x, and the input channel of N has the message sequence y.

The <u>initial</u> <u>state</u> of M and N with K-capacity channels is $[m_0,n_0,E,E]$ where m_0 and n_0 are the initial nodes of M and N respectively, and E is the empty string.

A state [m,n,x,y] of M and N with K-capacity channels is called an <u>overflow</u> <u>state</u> iff either m is a sending node and $|y|=K$ or n is a sending node and $|x|=K$.

Let s=[m,n,x,y] be a state of M and N with K-capacity channels and let e be an output edge of node m or n. A state s' of M and N with K-capacity channels is said to <u>follow</u> s <u>over</u> e, denoted s--e-->s', iff s is not an overflow state and the following four conditions are satisfied:

i. <u>If</u> e is from m to m' in M <u>and</u> is labelled send(g),
<u>then</u> s'=[m',n,x,y.g], where "." is the concatenation operator.

ii. <u>If</u> e is from n to n' in N <u>and</u> is labelled send(g),
<u>then</u> s'=[m,n',x.g,y].

iii. <u>If</u> e is from m to m' in M <u>and</u> is labelled receive(g),
<u>and</u> x=g.x',
<u>then</u> s'=[m',n,x',y].

iv. <u>If</u> e is from n to n' in N <u>and</u> is labelled receive(g),
<u>and</u> y=g.y',
<u>then</u> s'=[m,n',x,y'].

Let s and s' be two states of M and N with K-capacity channels, s' <u>follows</u> s if there is a directed edge e in M or N such that s--e-->s'.

Let s and s' be two states of M and N with K-capacity channels. s' is <u>reachable</u> <u>from</u> s if either 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 s of M and N with K-capacity channels is <u>reachable</u> if it is reachable from the initial state of M and N. The set R_K of all reachable states of M and N with K-capacity channels is called the <u>reachable</u> <u>set</u> of M and N with K-capacity channels.

A state s of M and N with K-capacity channels is said to be a <u>nonprogress</u> <u>state</u> if no state follows s. For instance, an overflow state is a nonprogress state.

In this paper, we address the following <u>communicating</u> <u>progress</u> <u>problem</u>. "Given two communicating machines M and N, is there a positive integer K such that the reachable set R_K of M and N with K-capacity channels has no nonprogress states?" If an instance of this problem has a positive answer, then it is possible to determine the smallest K, denoted K_{min}, for which this instance has a positive answer. This is because, in this case, the set $R=\bigcup_{K\geq 1} R_K$ is finite as can be shown from Theorem 2 (below). Therefore K_{min} can be computed as follows.

$$K_{min}=\max_{[m,n,x,y] \text{ in } R}(|x|,|y|).$$

In the next section, we discuss two other equivalent forms for the communication progress problem; then in section IV, we prove that one of these forms, and hence all of them, are undecidable.

3. OTHER FORMS FOR THE COMMUNICATION PROGRESS PROBLEM

Let M and N be two communicating machines. A state s=[m,n,x,y] of M and N with K-capacity channels is a deadlock state if m and n are two receiving nodes in M and N respectively and $|x|=|y|=0$.

A state s=[m,n,x,y] with K-capacity channels is an unspecified reception state if one of the following two conditions is satisfied

i. $x=g_1.g_2. \ldots .g_k$, for some $k\geq 1$ and m is a receiving node with no output labelled receive(g_1) in M.

ii. $y=g_1.g_2. \ldots .g_k$, for some $k\geq 1$ and n is a receiving node with no output labelled receive(g_1) in N.

An overflow or a deadlock state is a nonprogress state; but an unspecified reception state is not necessarily a nonprogress state. Nevertheless, the following theorem implies that an unspecified reception state always leads to a nonprogress state.

Theorem 1: Let M and N be two communicating machines; and let R_K be the reachable set of M and N with K-capacity channels. R_K has a nonprogress state iff R_K has an overflow state, a deadlock state, or an unspecified reception state.

Proof: If Part: According to the definition of "follow", a deadlock state or an overflow state is a nonprogress state. It remains to show that an unspecified reception state leads to a nonprogress state. Assume that an unspecified reception state s=[m,n,x,y] is in R_K. There are two cases to consider:

i. $x=g_1.g_2. \ldots .g_k$, for some $k\geq 1$ and m is a receiving node with no output labelled receive(g_1) in M.

ii. $y=g_1.g_2. \ldots .g_k$, for some $k\geq 1$ and m is a receiving node with no output labelled receive(g_1) in N.

Since the proofs for the two cases are similar, only case i is considered. If a state s' is reachable from s then there exist states $s_1,\ldots,s_r$, such that $s=s_1$, $s'=s_r$ and s_i--e_i-->s_{i+1}, where e_i is an edge in N, for $i=1,\ldots,r-1$. (Notice that no e_i is in M from case i).

Let the number of sending (or receiving) edges in $\{e_i|i=1,\ldots,r-1\}$ be S (or R respectively). Therefore, $S\leq K-|x|$ and $R\leq|y|$. In other words, the number of states reachable from s is finite; and a nonprogress state must be reachable from s.

Only If Part: Assume that a nonprogress state $s=[m,n,x,y]$ is in R_K. There are two cases to consider.

i. <u>Both m and n are receiving nodes</u>: If x and y are empty then s is a deadlock state. Otherwise, let $x=g_1. \ldots .g_k$. Since $s=[m,n,x,y]$ is a nonprogress state, then no output of m is labelled receive(g_1); and s is an unspecified reception state.

ii. <u>Either m or n is a sending node</u>: If m (or n) is a sending node and $|y|<K$ (or $|x|<K$ respectively) then there is a state following s; contradiction. Therefore, $|y|\geq K$ (or $|x|\geq K$ respectively) and s is an overflow state. []

From Theorem 1, the communication progress problem can be equivalently stated as follows. "Given two communicating machines M and N, is there a positive integer K such that the reachable set R_K of M and N with K-capacity channels has no overflow, no deadlock, and no unspecified reception states?". This proves that overflows, deadlocks, and unspecified receptions are <u>the</u> causes of nonprogress between two machines which communicate via finite-capacity channels. Next we discuss a third equivalent form for this same problem.

Let M and N be two communicating finite state machines with the same set S of messages. A <u>state</u> of M and N with <u>infinite-capacity channels</u> is a four-tuple [m,n,x,y] where m and n are two nodes in M and N respectively and x and y are two strings of messages from the set S. Notice that in this definition the length of x or y is not required to be bounded by any constant. Hence, the concept of an overflow state is not present in this case.

The initial state of M and N with infinite-capacity channels is $[m_0,n_0,E,E]$ where m_0 and n_0 are the initial nodes of M and N respectively, and E is the empty string.

Let s=[m,n,x,y] be a state of M and N with infinite-capacity channels and let e be an output edge of node m or n. A state s' of M and N with infinite-capacity channels is said to follow s over e, denoted s--e-->s', iff the four conditions i, ii, iii, and iv of the above "follow-over" definition for K-capacity channels are satisfied.

From this "follow-over" definition, the definitions of "follow", "reachable from", "reachable", and "reachable set" for infinite-capacity channels are similar to their counter parts for K-capacity channels. Also the definitions for a "nonprogress state", a "deadlock state" and an "unspecified reception state" are defined for infinite capacity channels in a similar way as for K-capacity channels.

Let R be the reachable set of M and N with infinite-capacity channels; and let K be a positive integer. The communication between M and N is said to be bounded by K iff each state [m,n,x,y] in R is such that $|x| \leq K$ and $|y| \leq K$; the communication is bounded iff it is bounded by K, for some positive integer K. The communication between M and N is said to be deadlock-free iff R has no deadlock states. The communication between M and N is said to be without unspecified receptions iff R has no unspecified reception states.

Theorem 2: Let M and N be two communicating machines, and let R_K be the reachable set for M and N with K-capacity channels. The following two statements are equivalent.

i. There is a positive integer K such that R_K has no overflow, deadlock, or unspecified reception states.

ii. The communication between M and N is bounded, deadlock-free, and without unspecified receptions.

Proof: i==>ii: Let K be a positive integer such that R_K has no overflow, deadlock, or unspecified reception states. Since R_K has no overflow states, each state in R is in R_K; and the communication between M and N is bounded by K. Also, no deadlock state or unspecified reception state is in R ;

and the communication between M and N is deadlock-free and has no unspecified receptions.

ii==>i: The communication between M and N is bounded; so there is a positive integer K such that for each state [m,n,x,y] in R , $|x| \leq K$ and $|y| \leq K$. We show by contradiction that R_K has no overflow, deadlock, or unspecified reception states. Let s=[m,n,x,y] be an overflow state in R_K; i.e. m is a sending node and $|y|=K$. Since each state in R_K is also in R, there is a state s'=[m',n',x',y'] which follows s in R such that $|y'|=K+1$; contradiction. Therefore, R_K has no overflow state. Similarly, we can show that R_K has no deadlock or unspecified reception states. []

From Theorem 2, the communication progress problem can be equivalently stated as follows. "Given two communicating machines M and N, is their communication bounded, deadlock-free, and without unspecified receptions?" In the next section we prove that this problem is undecidable.

4. UNDECIDABILITY OF THE COMMUNICATION PROGRESS PROBLEM

To prove the undecidability of the communication progress problem, we need first to establish a mapping from Post machines into pairs of communicating finite state machines. (A similar mapping from Turing machines into a slightly different model of communicating machines is discussed in [2].)

Theorem 3: For any Post machine P there are two communicating machines M and N which satisfy the following two conditions:

i. There is a node f in N such that P halts over the empty string iff there is a state of the form [m,f,x,y] in the reachable set R of M and N with infinite-capacity channels.

ii. For any state s in R, if s is not of the form [m,f,x,y], then s is neither a deadlock nor an unspecified reception state. (Informally, this implies that M and N cannot reach a deadlock or an unspecified reception state before the execution of N reaches node f.)

Proof: A Post machine P is a finite directed graph with one variable z, whose value can be any string over the symbols {0,1,#} [7]. Each vertex in the graph corresponds to a statement which has one of the forms shown in Figure 1.

The ASSIGNMENT statements allowed in P are to concatenate a symbol, namely 0, 1, or # to the right of z.

The TEST statement checks the leftmost symbol of z, namely head(z), and deletes it after making the decision.

A Post machine is said to halt over the empty string iff the computation of P starting with z=E (where E denotes the empty string) eventually reaches the HALT statement.

Given a Post machine P, we show how to construct two communicating machines M and N with the set {0,1,#,$} of messages such that conditions i and ii are satisfied. Machine M is shown in Figure 2. Informally, machine M sends every message it receives. Notice that the nodes of machine M are labelled 0, 1, #, and E (for the empty string); later we prove that variable z of P can be written as a string y.m.x where y and x are two strings over {0,1,#,$} and m is a level of a node in M.

Machine N is the finite labelled directed graph constructed by applying four <u>transformation</u> <u>rules</u> to the different types of vertices in the given Post machine P.

Rule T_1, illustrated in Figure 3a, transforms the START statement in P to an arrow which indicates the initial node in N.

Rule T_2, illustrated in Figure 3b, transforms the HALT statement in P into the special node f in N. Notice that node f can be selected later as a sending or a receiving node.

Rule T_3, illustrated in Figure 3c, transforms an assignment z<--z.a, where "a" is in {0,1,#}, into a sending node with an output labelled send(a) in N.

Rule T_4 is illustrated in Figure 3d. Informally, N simulates the test empty(z)? by sending the special symbol **$** to M, then it waits to receive from M. If it receives the same symbol **$**, it recognizes that z is empty. If it receives 0, 1, or #, then it recognizes that z is not empty. In this case, M removes the symbol **$** then waits to receive the next symbol and depending on its type, the execution of N proceeds along one of the three output branches.

Let T(v) be the resulting subgraph after applying the appropriate transformation rule T to a vertex v in P. And let the entry node of the subgraph T(v) in N be labelled v also.

Using induction, the following statement can be proven from the transformation rules T_1 to T_4. A vertex v in P is reached with the value of z being w iff a state [m,v,x,y], where y.m.x=w, is in the reachable set R of M and N with infinite-capacity channels. From this statement, conditions i and ii can be proven as follows:

i. Node f in N is the one which corresponds to the HALT statement in P. So, P halts over the empty string iff there is a state [m,f,x,y] in the reachable set R.

ii. Since each receiving node in M or N, except possibly node f, has an output labelled receive(g) for each message g in {0,1,#}, then if [m,n,x,y] is in R and $n \neq f$ then [m,n,x,y] cannot be an unspecified reception state. Also, since the Post machine P only stops at the HALT statement, so if [m,n,x,y] is in R and $n \neq f$ then [m,n,x,y] is not a deadlock state. []

Since the halting problem for Post machines is undecidable [7], the communication progress problem is also undecidable as shown in the next theorem.

Theorem 4: Given two communicating machines; the problem of whether their communication is bounded, deadlock-free, and without unspecified receptions is undecidable.

Proof: (by contradiction) Assume that there is an algorithm, say algorithm A, to decide whether the communication between two given machines is bounded, deadlock-free, and without unspecified receptions. We show that algorithm A can be used to decide whether any Post machine halts over the empty string.

Let P be a Post machine; and let M and N be the two communicating machines constructed from P as discussed in Theorem 3. Also let R be the reachable set of M and N with infinite-capacity channels. From Theorem 3, P halts over the empty string iff a state [m,f,x,y] is in R, where f is the special node in N, discussed in Theorem 3.

Let N' be the resulting machine from N by replacing node f by a construct shown in Figure 4 which continuously receives a message then sends it. Clearly, the communication between M and N' is deadlock-free and without unspecified receptions.

Apply algorithm A to machines M and N' to decide whether their communication is bounded, deadlock-free, and without unspecified receptions. If the answer is "no" implying that the communication between M and N is unbounded, then no state of the form [m,f,x,y] is in R and P does not halt over the empty string. On the other hand, if the answer is "yes" implying that the communication between M and N is bounded, then the set R is finite and all its states can be generated to check whether or not it has a state of the form [m,f,x,y]; this in turn can decide whether or not P halts over the empty string.

[]

From Theorem 4, there is no algorithm to answer the communication progress problem in general. Still, there are two approaches to bypass this negative result. First, identify special classes of communicating machines for which the problem is decidable. One example of this approach is discussed in the next section. The second approach is based on the observation that in most instances one is more interested in proving a positive answer for the problem. Therefore, in section VI we discuss some sufficient conditions which if satisfied by two communicating machines then the communication progress problem for them has a positive answer.

5. ALTERNATING COMMUNICATING MACHINES

A communicating machine M is called alternating if each sending node in M is immediately followed by receiving nodes only. We show in the next two theorems that the communication progress problem for alternating machines is decidable, and that its decidability algorithm is polynomial.

Theorem 5: The communication between any two alternating machines is bounded by two.

Proof: Let M and N be two alternating communicating machines and let R be the reachable set of M and N with infinite-capacity channels. We prove that the communication between M and N is bounded by two. Let $[m_0,n_0,E,E]$ be the initial state. there are four cases to consider:

i. **m_0 and n_0 are receiving nodes:** R contains only the initial state. Therefore the communication is bounded by one (and so by two).

ii. **m_0 is a sending node and n_0 is a receiving node:** As shown in Figure 5a, if the initial state is of the form [s,r,E,E], where s (or r) denotes a sending (or receiving respectively) node, then any state in R is in any one of the following five forms: [s,r,E,E], [r,r,E,E], [r,s,E,E] [r,r,g,E] or [r,r,E,g], where g denotes a string which consists of only one message. In this case, the communication is bounded by one (and so by two).

iii. **m_0 is a receiving node and n_0 is a sending node:** Using an argument similar to that in case ii, it can be shown that the communication is bounded by one (and so by two).

iv. **m_0 and n_0 are sending nodes:** As shown in Figure 5b, the initial state is of the form [s,s,E,E], where s denotes a sending node, then any state in R is in any one of the 13 forms in Figure 5b, where r denotes a receiving node, g denotes a string which consists of one message, gg denotes a string which consists of two messages. In this case, the communication is bounded by two. []

Theorem 6: There is a polynomial-time algorithm to solve the communication progress problem for any two alternating communicating machines.

Proof: Let M and N be two alternating communicating machines over a set S of messages. And let R be the reachable set of M and N with infinite-capacity channels. From Theorem 5, any state [m,n,x,y] in R is such that $|x| \leq 2$ and $|y| \leq 2$. Thus, the number of states in R is $O(uvw^4)$, where u is the number of nodes in machine M, v is the number of nodes in machine N, and w is the number of messages in set S. Therefore, each state in R can be generated and checked for being a deadlock or an unspecified reception state. Clearly, this algorithm can solve the communication progress problem for M and N and it requires polynomial time. []

In the next section, we discuss a set of sufficient conditions to ensure that the communication progress problem has a positive answer.

6. COMPATIBLE COMMUNICATION

Let M and N be two communicating machines; and let p and q be two directed paths which start with the initial nodes in M and N respectively. Paths p and q are said to be <u>compatible paths</u> if for i=1,2,..., the ith edge in p is labelled send(g) (or receive(g)) and the ith edge in q is labelled receive(g) (or send(g) respectively). The communication between M and N is said to be <u>compatible</u> if for any directed path p which starts with the initial node in M, there exists exactly one directed path q which starts with the initial node in N, and vice versa, such that p and q are compatible.

The reason for our interest in compatible communication is two fold. First, compatibility is a sufficient condition to ensure that the communication is deadlock-free and without unspecified receptions as we prove in Theorem 7. Second, it is decidable whether the communication between two machines is compatible, as we prove in Theorem 8.

Theorem 7: Let M and N be two communicating machines. If the communication between M and N is compatible then it is deadlock-free and without unspecified receptions.

Proof: Let M and N be two machines whose communication is compatible. And let R be the reachable set of M and N with infinite-capacity channels. We show by contradiction that no nonprogress state is in R. Assume that a nonprogress state s is in R, there are two cases to consider; each of which leads to a contradiction:

i. s is a deadlock state: Since s is in R, there exist states $s_0,\ldots,s_r$, such that s_0 is the initial state, $s_r=s$, and s_{i-1}--e_i-->s_i for $i=1,\ldots,r$. The set of edges $\{e_i|i=1,\ldots,r$ and e_i is in M$\}$ corresponds to a directed path p which starts with the initial node in M. Similarly, the set of edges $\{e_i|i=1,\ldots,r$ and e_i is in N$\}$ corresponds to a directed path q which starts with the initial node in N. Since s is a deadlock state, then $|p|=|q|$. There are two cases to consider:

   a. p and q are compatible: In this case, if path p is extended in any way into p' in M, then no directed path q' which starts with the initial node in N is compatible with p'. This contradicts the assumption that the communication between M and N is compatible.

   b. p and q are not compatible: Since the communication between M and N is compatible, there is a directed path $\tilde{p}$ which starts with the initial node in N such that p and $\tilde{p}$ are compatible. Clearly, $|q|=|p|=|\tilde{p}|$ and paths q and $\tilde{p}$ are not identical. let e and $\tilde{e}$ be the first different edges in q and $\tilde{p}$ respectively. Edges e and $\tilde{e}$ have the same tail node; and they are either the ith sending edges or the ith receiving edges in their respective paths. Therefore, they correspond to the ith receiving edge or the ith sending edge of path p in M; i.e., they have identical labels; this contradicts the fact that no two outputs of the same node in N have identical labels.

ii. s is an unspecified reception state: Using a similar argument as in case i can lead to a contradiction. []

Theorem 8: There is a polynomial-time algorithm to decide whether the communication between two communicating machines is compatible.

Proof: Let M and N be two communicating machines with a set S of messages. Construct machine $\overline{N}$ from N by replacing each sending node by a receiving node and vice versa, and by replacing each label "send(g)" by "receive(g)" and vice versa. View machine M and $\overline{N}$ as two finite automata over the alphabet {send(g),receive(g)|g is in S}; and assume that each node in M or $\overline{N}$ is an accepting state. Each path p (or q) which starts with the initial node in machine M (or $\overline{N}$) corresponds to a word in the regular language L(M) (or L($\overline{N}$)) accepted by the automaton M (or $\overline{N}$ respectively). Therefore, the communication between M and N is compatible iff L(M)=L($\overline{N}$). Since whether L(M)=L($\overline{N}$) is decidable in polynomial time [5], the problem of whether the communication between M and N is compatible is decidable in polynomial time. []

From Theorem 7, compatibility guarantees freedom of deadlocks and unspecified receptions; however, it does not guarantee boundedness. Therefore, compatibility alone does not ensure a positive answer to the communication progress problem; and an additional condition is needed for that purpose.

Theorem 9: Let M and N be two communicating machines. Assume that the communication between M and N is compatible and that each directed cycle in M or N has at least one sending and one receiving nodes. Then, there is a positive integer K such that the reachable set R_K for M and N with K-capacity channels has no nonprogress states.

Proof: Since any directed cycle in M or N contains at least one sending and one receiving nodes, define K to be the maximum number of successive nodes of the same type (sending or receiving) in M or N. We show that any reachable state s=[m,n,x,y] of M and N with infinite-capacity channels satisfies the following three conditions:

i. s is not a deadlock state.

ii. s is not an unspecified reception state.

iii. $|x| \leq K$ and $|y| \leq K$.

Conditions i and ii are satisfied since the communication between M and N is compatible (Theorem 7). It remains to prove condition iii. Let p and q be two compatible infinite paths in M and N respectively. Neither p nor q contains an infinite number of successive sending nodes or an infinite number of successive receiving nodes. Therefore, p or q consists of a finite number of sending nodes followed by a finite number of receiving nodes, followed by a finite number of sending nodes, etc. All these numbers are bounded by K. Without loss of generality, assume that p begins with a number of sending nodes; hence q begins with the same number of receiving nodes. p can send up to K messages before q receives any of them. Then p waits until q receives all of them and starts sending to p. In this case, q can send up to K messages, and then it waits until p receives all these messages and starts sending to q. The same argument can be applied to the rest of p and q. Therefore any reachable state [m,n,x,y] along p and q is such that $|x| \leq K$ and $|y| \leq K$. Since any state must be reached along compatible paths, any reachable state [m,n,x,y] is such that $|x| \leq K$ and $|y| \leq K$. This completes the proof that $R_K = R$ and that no state in R_K is a nonprogress state. []

7. SUMMARY OF RESULTS

We have addressed the following communication progress problem. "Given two communicating machines M and N, is there a positive integer K such that the reachable set of M and N with K-capacity channels has no nonprogress states?" We have shown the following

i. Nonprogress between two machines which communicate via finite-capacity channels is caused by overflows, deadlocks, or unspecified receptions (Theorem 1).

ii. The communication progress problem is equivalent to the problem, "Is the communication between M and N bounded, deadlock-free, and without unspecified receptions?" (Theorem 2).

iii. The problem is undecidable in general (Theorem 4).

iv. The problem is decidable for a special class of communicating machines

REFERENCES

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

[2] D. Brand and P. Zafiropulo, "On communicating finite-state machines," IBM Research Report, RZ1053(#37725), Jan. 1981.

[3] P. R. Cunha and T. S. Maibaum, "A synchronization calculus for message oriented programming," Res. Rep. CS-80-43, Dept. of Comp. Sc., Univ. of Waterloo, Sep. 1980.

[4] A. Danthine, "Protocol representation with finite state models," IEEE Trans. Comm., Vol. COM-28, No. 4, April 1980, pp. 632-643.

[5] M. R. Garey and D. S. Johnson, Computers and interactability, a guide to the theory of NP-completeness, W. H. Freeman and Company, San Francisco, 1979.

[6] M. G. Gouda and Y. T. Yu, "Designing deadlock-free and bounded communication protocols," Tech. Rep. 179, Dept. of Comp. Sc., Univ. of Texas at Austin, June 1981. submitted to IEEE Trans. on Comm..

[7] Z. Manna, Mathematical theory of computation, McGraw-Hill Book company, 1974.

[8] C. A. Sunshine, "Formal modeling of communication protocols," USC/Inform. Sc. Institute, Res. Rep. 81-89, March 1981.

[9] C. H. West, "An automated technique of communication protocol validation," IEEE Trans. Comm., Vol. COM-26, pp.1271-1275, Aug. 1978.

[10] Y. T. Yu and M. G. Gouda, "Deadlock-detection for a class of communicating finite state machines," Tech. Rep. 193, Dept. of Computer Sciences, Univ. of Texas at Austin, Feb. 1982. Submitted to IEEE Trans. on Comm..

[11] P. Zafiropulo, et. al., "Towards analyzing and synthesizing protocols," IEEE Trans. Comm., Vol. COM-28, No. 4, April 1980, pp. 651-661.

called alternating machines; and the decidability algorithm is polynomial (Theorem 6).

v. The problem has a positive answer if the communication between M and N is compatible and each directed cycle in M or N has at least one sending and one receiving nodes (Theorem 9). These two conditions can be detected in polynomial time (Theorem 8).

We have also derived two results concerning K_{min}, the smallest K for which the problem has a positive answer if at all.

i. If the given M and N are alternating machines then $k_{min}=2$ (Theorem 5).

ii. If the given M and N satisfy the two cited conditions in Theorem 9 then K_{min} = the maximum number of successive nodes of the same type (i.e., sending or receiving) in M or N (Theorem 9).

Other results related to the communication progress problem have appeared in [3,6]. In [3], the communication progress problem is shown to be decidable if M and N exchange one type of messages. The complexity of this decidability algorithm is yet to be determined. In [6], a synthesis approach to the problem is taken; in particular, the following problem is addressed. "Given one communicating machine M, it is required to synthesize another communicating machine N such that the communication progress problem for M and N has a positive solution." A synthesis algorithm is discussed in [6] along with an algorithm to compute K_{min} for M and N. Both algorithms are polynomial in the number of nodes in M and N.