CLOSED COVERS: TO VERIFY PROGRESS FOR COMMUNICATING FINITE STATE MACHINES

Mohamed G. Gouda

Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712

TR-191 January 1982

ABSTRACT

Consider nonterminating finite state machines which communicate exclusively by exchanging messages. We discuss a technique to verify that the communication between a given pair of such machines will progress indefinitely; this implies that their communication is free of deadlocks and unspecified receptions. The technique is based on finding a set of global states for the communicating pair such that the following two conditions are satisfied: (i) the initial global state is in that set; and (ii) starting from any global state in that set, an "acyclic version" of the communicating pair must reach a global state in that set. We call such a set a closed cover; and prove that the existence of a closed cover for a communicating pair is sufficient to guarantee indefinite communication progress.

Keywords: Communicating finite state machines, communication progress, communication protocols, verification techniques.

I. INTRODUCTION

The model of communicating finite state machines is an abstraction of processes which communicate exclusively by exchanging messages. The abstraction is achieved by ignoring the internal data structures and internal operations of communicating processes and representing each process by its external behaviour; i.e., by all possible sequences of its sending and receiving operations with other processes. This abstract representation of a process is called a communicating finite state machine.

Communicating finite state machines are useful in the specification [3], [8], analysis [1], [2], and design [4], [6], [10] of communication protocols. So far, however, state exploration [7], [9] is the only available technique to verify progress properties for these machines. This technique has two apparent shortcomings. First, the number of generated global states is usually "large" making it difficult to use for practical protocols. Second, state exploration can be used only for bounded communications, i.e., when the channels between communicating finite state machines have finite capacities.

In this paper, we propose another technique to verify progress for two communicating finite state machines and show that the two shortcomings of state exploration are remedied to some degree by this new technique.

II. COMMUNICATING FINITE STATE MACHINES

A communicating finite state machine P is a directed labelled graph where each edge is labelled either "send(m)" or "receive(m)" for some message m from a finite set M. An edge labelled "send(m)" is called a sending edge; otherwise, it is called a receiving edge. One of the nodes in P is identified as its initial node; and all the nodes in P are reachable by directed paths from the initial node. P is assumed to be "nonterminating"; i.e., each node in P must have at least one output edge.

Let p be a direct path, in P, which starts from node i, ends at node j, and consists of the directed edges $e_1, e_2, \ldots, e_v$. And let node k and its input edge e_u and its output edge e_{u+1} be in p. Then p[i,k] denotes the directed path which consists of the edges $e_1, e_2, \ldots, e_u$, while p[k,j] denotes the path which consists of the edges $e_{u+1}, \ldots, e_v$. Therefore, path p can be denoted as p[i,j].

The sequence of sent messages along a directed path p whose edges are $e_1, \ldots, e_v$ in P is defined as $s_1 \cdot s_2 \cdot \ldots \cdot s_v$,

where . is the usual string concatenation operator, and

$s_i = m$ ...if e_i is labelled "send(m)"

$= E$ (the empty string) ...otherwise.

(Notice that x.E.y = x.y).

Similarly, the sequence of received messages along a directed path p whose edges are $e_1, \ldots, e_v$ in P is defined as $r_1 \cdot r_2 \cdot \ldots \cdot r_v$,

where r_i = m ...if e_i is labelled "receive(m)"
= E (the empty string) ...otherwise.

Let P and Q be two communicating finite state machines with the same set M of messages. A global state s of P and Q is an ordered tuple with four components: s=[i,j,x,y] where i and j are two nodes in P and Q respectively, and x and y are two (possibly empty) strings over the set M. Informally, [i,j,x,y] defines a global state at which the execution of P has reached node i, the execution of Q has reached node j, and the contents of the input channels of P and Q are x and y respectively.

The initial global state s_0 of P and Q is of the form $s_0=[i_0,j_0,E,E]$ where i_0 and j_0 are the initial nodes of P and Q respectively and E is the empty string.

A global state $[i_2,j_2,x_2,y_2]$ is reachable from a global state $[i_1,j_1,x_1,y_1]$ over the two paths $p[i_1,i_2]$ and $q[j_1,j_2]$ in P and Q respectively iff the following two conditions R1 and R2 are satisfied:

R1: $x_1.s' = r.x_2$, and
$y_1.s = r'.y_2$

where s and s′ are the two sequences of sent messages along p and q respectively; and r and r′ are the two sequences of received messages along p and q respectively.

R2: There are no two nodes i and j with receiving output edges on the two paths p and q respectively such that

$x_1.u' = v$, and
$y_1.u = v'$

where u and u′ are the two sequences of sent messages along $p[i_1,i]$ and $q[j_1,j]$ respectively, and v and v′ are the two sequences of received messages along $p[i_1,i]$ and $q[j_1,j]$ respectively.

Notation: $[i_1,j_1,x_1,y_1]$---p,q-->$[i_2,j_2,x_2,y_2]$

denotes that the global state $[i_2,j_2,x_2,y_2]$ is reachable from the global state $[i_1,j_1,x_1,y_1]$ over the two paths $p[i_1,i_2]$ and $q[j_1,j_2]$ in P and Q respectively. []

A proof for the following lemma is in the Appendix.

Lemma 1: Let P and Q be two communicating finite state machines; and let $p[i_1,i_2]$ and $q[j_1,j_2]$ be any two paths in P and Q respectively.

If (i) $[i_1,j_1,x_1,y_1]$---p,q-->$[i_2,j_2,x_2,y_2]$, and

(ii) nodes i and j are any two nodes in paths p and q respectively, and

(iii) $[i_1,j_1,x_1,y_1]$---p,q-->$[i,j,x,y]$

then (iv) $[i,j,x,y]$---p,q-->$[i_2,j_2,x_2,y_2]$. []

A global state s_2 is reachable from a global state s_1 iff there are two directed paths p and q in P and Q respectively such that s_1---p,q-->s_2.

A global state is reachable iff it is reachable from the initial global state.

A global state s=[i,j,x,y] is called a blocking state for P (or for Q) iff any reachable state from s is of the form [i,k,w,z] (or [k,j,w,z] respectively). Informally, a blocking state for P is one in which no further execution, or progress, of P is possible. A blocking state for P (or Q) can either be a deadlock state or an unspecified reception state for P (or Q). The exact definitions of these states are irrelevant to this paper; but they can be found in [4].

The communication between two communicating finite state machines P and Q will progress indefinitely if no reachable global state is a blocking state for P or for Q. In this paper, we propose a new technique to verify that the communication between two given finite state machines will progress indefinitely. The technique is based on the concept of closed covers discussed next.

III. CLOSED COVERS

Let C be a set of global states of two communicating finite state machines P and Q. A node i in P (or node j in Q) is said to be covered by C iff C has a global state of the form [i,k,x,y] (or [k,j,x,y] respectively). Set C is called a cover for P and Q iff every directed cycle in P or Q has at least one node covered by C.

Let C be a cover for P and Q. Define AP to be the directed labelled graph constructed from P by partitioning every node i, covered by C, into two nodes i_1 and i_2 where i_1 has all the output edges of node i and i_2 has all the input edges of i. Node i_1 in AP is called the input version of node i in P; and i_2 is called the output version of i. Also, define AQ to be the directed labelled graph

constructed from Q in a similar way. Since C is a cover for P and Q, both AP and AQ are acyclic; hence AP and AQ are called the acyclic versions of P and Q with respect to C.

Except for being acyclic and for their lack of initial nodes, AP and AQ are two communicating finite state machines with the same set M of messages as P and Q. A global state of AP and AQ is of the form [i,j,x,y] where i and j are nodes in AP and AQ respectively, and x and y are two (possibly empty) strings over M. Let p and q be two directed paths in AP and AQ respectively; and let s_1 and s_2 be two global states of AP and AQ. Then s_2 is reachable from s_1 over p and q, denoted s_1---p,q-->s_2, iff the above two reachability conditions R1 and R2 are satisfied for s_1, s_2, p and q. Also, s_2 is reachable from s_1 iff there are two paths p and q in AP and AQ respectively such that s_1---p,q-->s_2.

Let C be a cover for P and Q; and let AP and AQ be the acyclic versions of P and Q with respect to C. A global state s=[i,j,x,y] in C is called closed iff the following condition is satisfied: Let i_1 and j_1 be the input versions of nodes i and j respectively; and let $p[i_1,k_2]$ and $q[j_1,l_2]$ be two directed paths in AP and AQ respectively.

If (i) $[i_1,j_1,x,y]$---p,q-->$[k_2,l_2,w,z]$, and

(ii) no other global state of AP and AQ is reachable from $[k_2,l_2,w,z]$

then (iii) k_2 is the output version of some node k in P, and

(iv) l_2 is the output version of some node l in Q, and

(v) [k,l,w,z] is in C.

A cover set C of P and Q is called a closed cover iff it satisfies the following two conditions: (i) the initial global state of P and Q is in C; (ii) each global state in C is closed.

Next, we show that the existence of a closed cover for two communicating finite state machines is sufficient to guarantee that their communication will progress indefinitely. In what follows, let C be a closed cover for two communicating finite state machines P and Q. Also, let p and q be two directed paths which start from the initial nodes i_0 and j_0 and end at some nodes i and j in P and Q respectively. The proofs of the following lemmas and theorem are in the Appendix.

Lemma 2:

If (i) $[i_0,j_0,E,E]$---p,q-->[i,j,x,y], and

(ii) nodes r and s are the Kth nodes covered by C in paths p and q respectively

then (iii) there exists a global state [r,s,w,z] in C, and

(iv) $[i_0,j_0,E,E]$---p,q-->[r,s,w,z]. []

Lemma 3:

If (i) $[i_0,j_0,E,E]$---p,q-->[i,j,x,y], and

(ii) path p has K nodes covered by C, and path q has L nodes covered by C such that $K \geq L$

then path q can be extended to some node s in Q such that the extended path q′ satisfies the following two conditions:

(iii) $[i_0,j_0,E,E]$---p,q-->[i,s,w,z], and

(iv) path q′ has K nodes covered by C. []

Lemma 3 is also true if the roles of paths p and q are reversed. It is straightforward to re-state Lemma 3 in this reversed form and to prove it using a similar proof to that of Lemma 3. From Lemmas 1, 2, and 3, the following theorem can be proved; the proof is in the Appendix.

Theorem 1: The communication between P and Q is guaranteed to progress indefinitely. []

IV. A METHODOLOGY TO VERIFY PROGRESS

From Theorem 1, to verify that the communication between two communicating finite state machines P and Q will progress indefinitely, it is sufficient to construct a set C of global states of P and Q then verify that C is a closed cover as follows:

(i) Show that the initial global state of P and Q is in C.

(ii) Then, show that each directed cycle in P or Q has at least one node covered by C.

(iii) Finally show that each global state [i,j,x,y] in C is closed as follows:

a. Construct the two acyclic versions AP and AQ of P and Q with respect to C; and let i_1 and j_1 be the input versions of nodes i and j in P and Q respectively.

b. Construct the set $S[i_1,j_1,x,y]$ of all global states of AP and AQ reachable from state $[i_1,j_1,x,y]$. This step is discussed later in detail.

c. Check that if a state $[k_2,l_2,w,z]$ is in $S[i_1,j_1,x,y]$ and if no other state in $S[i_1,j_1,x,y]$ is reachable from $[k_2,l_2,w,z]$ then k_2 and l_2 are the output versions of some nodes k and l in P and Q respectively such that [k,l,w,z] is in C.

To construct the set $S[i_1,j_1,x,y]$ of all global states reachable from $[i_1,j_1,x,y]$ in AP and AQ, usual state exploration techniques [9] can be used as follows:

a. $[i_1,j_i,x,y]$ is in $S[i_1,j_1,x,y]$.

b. If $[k,l,w,z]$ is in $S[i_1,j_1,x,y]$ and if there is an edge, labelled "send(m)", from node k (or l) to node r in AP (or AQ), then $[r,l,w,z.m]$ (or $[k,r,w.m,z]$ respectively) is in $S[i_1,j_1,x,y]$.

c. If $[k,l,w,z]$ is in $S[i_1,j_1,x,y]$, and there is an edge, labelled "receive(m)", from node k (or l) to node r in AP (or AQ), and if w=m.s (or z=m.s), then $[r,l,s,z]$ (or $[k,r,w,s]$ respectively) is in $S[i_1,j_1,x,y]$.

Notice that since AP and AQ are acyclic, set $S[i_1,j_1,x,y]$ is finite and can be constructed in a finite time.

V. EXAMPLES

Example 1: Figures 1a and 1b show two communicating finite state machines P_1 and Q_1 whose initial nodes are "a" and "e" respectively. Consider the following set C_1 of global states of P_1 and Q_1:

$$C_1 = \{[a,e,E,E], [c,g,E,E]\}$$

where E is the empty string. First, the initial global state [a,e,E,E] is in C_1. Second, the directed cycle of P_1 has two nodes "a" and "c" covered by C_1; and the directed cycle of Q_1 has two nodes "e" and "g" covered by C_1. Now, it remains to show that every global state in C_1 is closed. Figures 1c and 1d show the acyclic versions AP_1 and AQ_1 of P_1 and Q_1 with respect to C_1. To show that [a,e,E,E] is closed, Figure 1e shows all the global states of AP_1 and AQ_1 reachable from

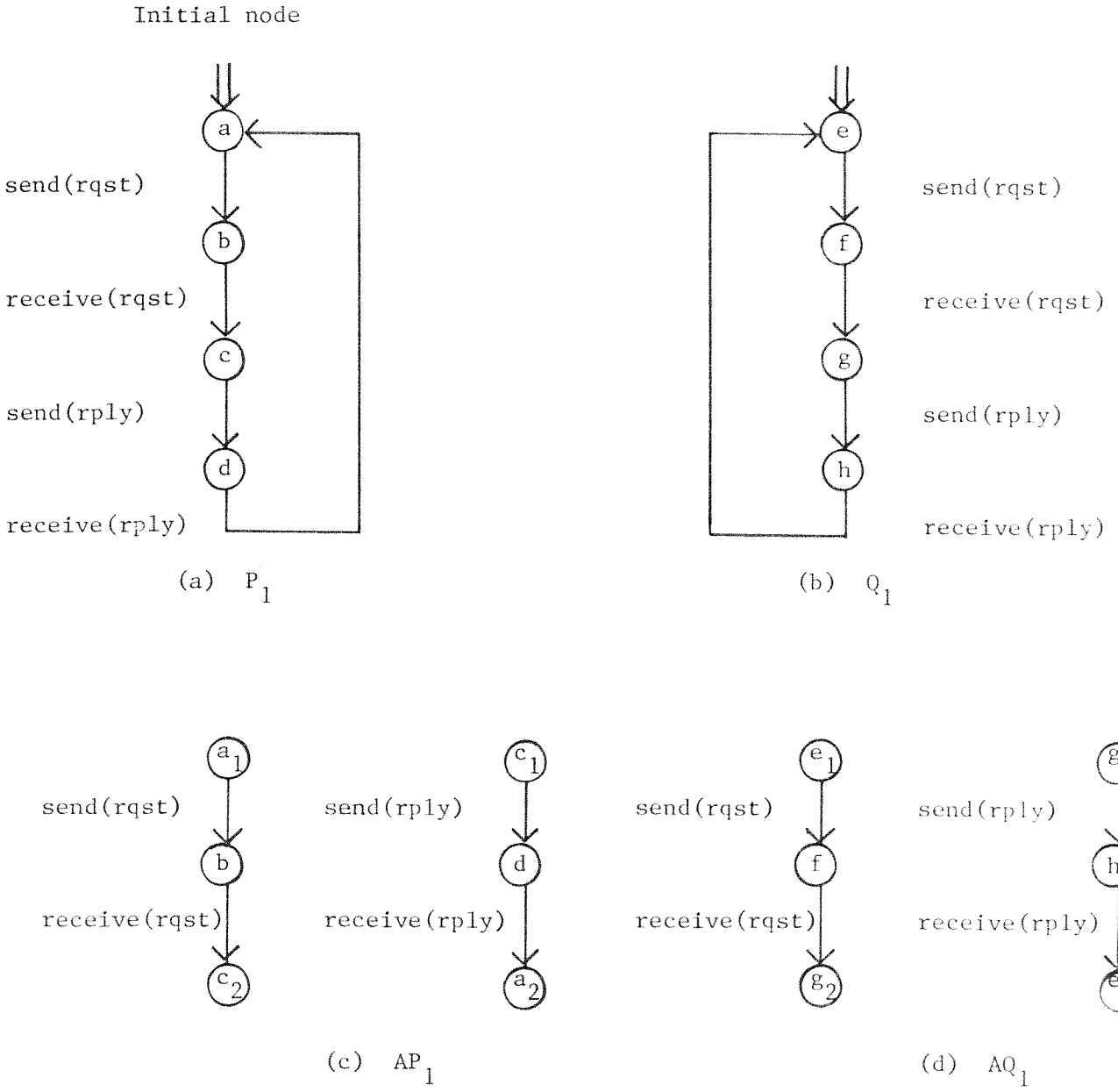
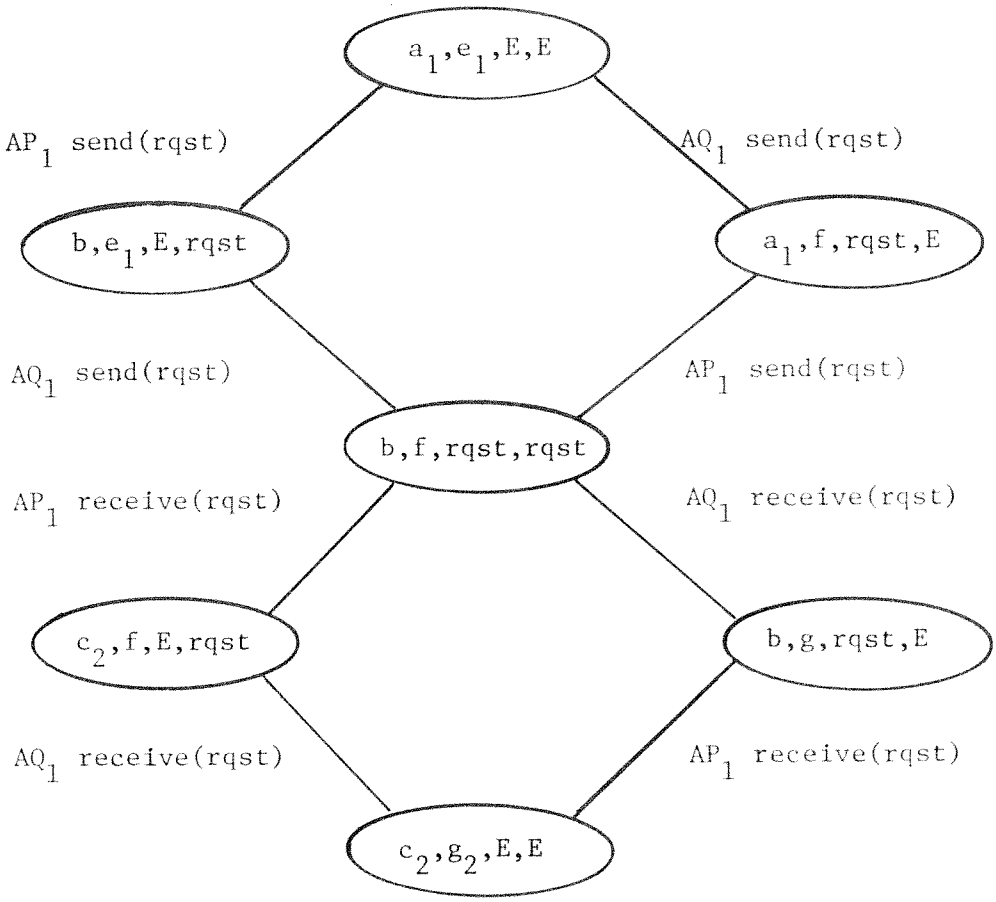


Figure 1 Example 1



(e) Proving that [a,e,E,E] is closed.

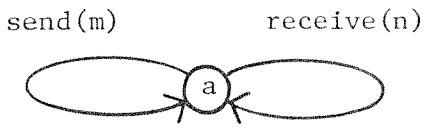
Figure 1 Example 1

$[a_1,e_1,E,E]$. Since $[c_2,g_2,E,E]$ is the only state with no other reachable state, and since [c,g,E,E] is in C_1, then [a,e,E,E] is closed. Similarly it can be shown that [c,g,E,E] is closed. Thus, C_1 is a closed cover for P_1 and Q_1; and the communication between P_1 and Q_1 is guaranteed to progress indefinitely by Theorem 1.

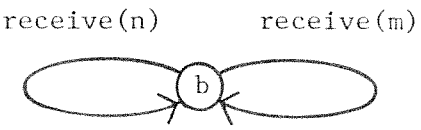
Notes: (i) Other closed covers can be found for this example. For instance, the two sets {[a,e,E,E]}, and {[a,e,E,E], [b,f,rqst,rqst], [c,g,E,E], [h,d,rply,rply]} are closed covers for P_1 and Q_1.

(ii) Communication progress in this example can be verified using usual exploration techniques [9] assuming that each of the two channels between P_1 and Q_1 has a capacity of two. On the other hand, using closed cover techniques can reduce the number of global states generated during the verification. For instance, in the above example all the global states at which a channel has two messages, which would have been generated using state exploration techniques, are not generated using the closed cover C_1. []

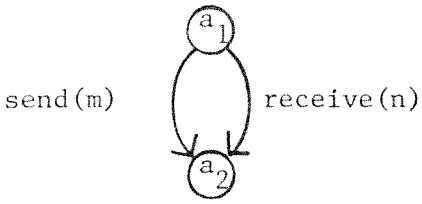
Example 2: Figures 2a and 2b show two communicating finite state machines P_2 and Q_2 whose initial nodes are "a" and "b" respectively. Consider the following set C_2={[a,b,E,E]} of global states of P_2 and Q_2 where E is the empty string. First, C_2 contains the initial global state. Second, each directed cycle in P_2 and Q_2 has one node covered by C_2; thus C_2 is a cover for P_2 and Q_2. It remains to show that [a,b,E,E] is closed. Figures 2c and 3d show the acyclic versions AP_2 and AQ_2 of P_2 and Q_2 with respect to C_2. Figure 2e shows all the global states of AP_2 and AQ_2 reachable from $[a_1,b_1,E,E]$; notice that the only state with no other reachable state is $[a_2,b_2,E,E]$. Thus,



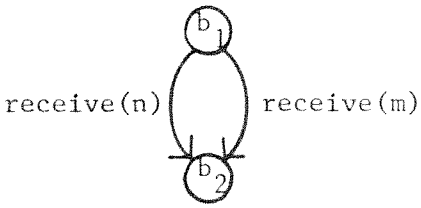
(a) P_2



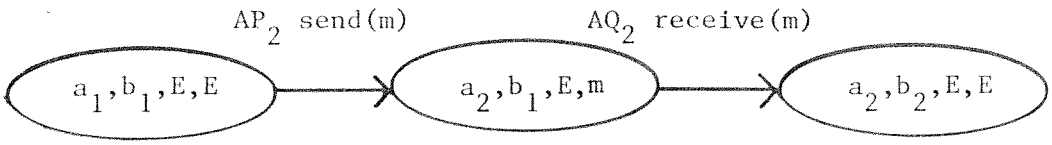
(b) Q_2



(c) AP_2



(d) AQ_2



(e) Proving that [a,b,E,E] is closed.

Figure 2 Example 2

[a,b,E,E] is closed; and C_2 is a closed cover for P_2 and Q_2; and the communication between P_2 and Q_2 is guaranteed to progress indefinitely.

Notice that the channel from P_2 to Q_2 must have an infinite capacity; hence usual state exploration techniques cannot be used to verify communication progress in this example. []

VI. INFINITE CLOSED COVERS

Theorem 1 is still applicable even if a closed cover has an infinite number of global states. One way to specify infinite closed covers is by introducing global state schemas. A global state schema of two communicating finite state machines P and Q with a set M of messages is an ordered tuple with four components: [i,j,X,Y] where i and j are two nodes in P and Q respectively, and X and Y are two regular expressions [5] over M. A global state [i,j,x,y] of P and Q is in a global state schema [i,j,X,Y] of P and Q iff x and y are two strings in the regular languages accepted by the regular expressions X and Y respectively.

Let H be a finite set of global state schemas of two communicating finite state machines. A global state s is in H iff s is in a global state schema in H. Because of this definition of global states being in a set H of global state schemas, H is a cover or a closed cover iff the global states in H satisfies the definitions in Section III. Also, Theorem 1 is still applicable to such a closed cover H.

To verify that a set H of global state schemas is a closed cover, it is not convenient to verify that each global state in H is closed since H can have an infinite number of global states. Rather, it is sufficient to verify that any global state schema [i,j,X,Y] in H is closed as follows:

a. Construct the acyclic versions AP and AQ of P and Q with respect to H; and let i_1 and j_1 be the input versions of nodes i and j respectively.

b. Construct the set $S[i_1,j_1,X,Y]$ of all global state schemas reachable from $[i_1,j_1,X,Y]$ in AP and AQ. This step is discussed in more detail later.

c. If $[k_2,l_2,W,Z]$ is in $S[i_1,j_1,X,Y]$ and if no other global state schema in $S[i_1,j_1,X,Y]$ is reachable from $[k_2,l_2,W,Z]$, then k_2 and l_2 must be the output versions of nodes k and l in P and Q respectively, and there must be a schema $[k,l,\underline{W',Z'}]$ in H such that the language accepted by W (or Z) is a subset of the language accepted by W′ (or Z′ respectively).

Notice that proving a global state schema h is closed implies that each global state in h is closed. It remains to show how to construct the set $S[i_1,j_1,X,Y]$:

a. $[i_1,j_1,X,Y]$ is in $S[i_1,j_1,X,Y]$

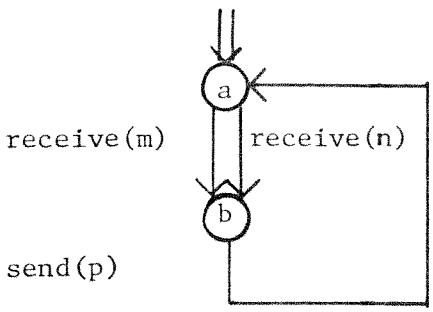
b. If [k,l,W,Z] is in $S[i_1,j_1,X,Y]$ and if there is an edge, labelled "send(m)", from node k (or l) to node r in AP (or AQ), then the global state schema [r,l,W,Z.m] (or [k,r,W.m,Z] respectively) is in $S[i_1,j_1,X,Y]$.

c. If [k,l,W,Z] is in $S[i_1,j_1,X,Y]$ and if there is an edge, labelled "receive(m)", from node k (or l) to node r in AP (or AQ), and if the regular languages accepted by W (or Z) has at least one string of the form m.s, then the global state schema [r,l,W/m,Z] (or [k,r,W,Z/m] respectively) is in $S[i_1,j_1,X,Y]$ where $W/m = \{s \mid m.s \text{ is in } W\}$ and $Z/m = \{s \mid m.s \text{ is in } Z\}$.

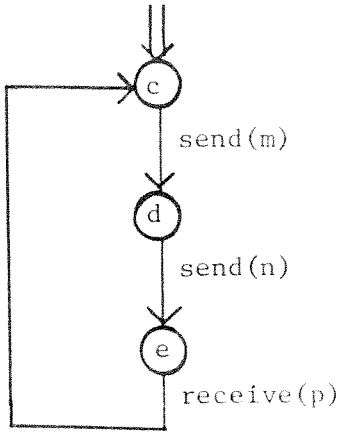
Example 3: Figures 3a and 3b show two communicating finite state machines P_3 and Q_3 whose initial nodes are "a" and "c" respectively. Consider the following set H of global state schemas of P_3 and Q_3.

$$H=\{[a,c,(m.n)^* + n.(m.n)^*,E],\ [b,e,(m.n)^+ + n.(m.n)^*,E]\}$$

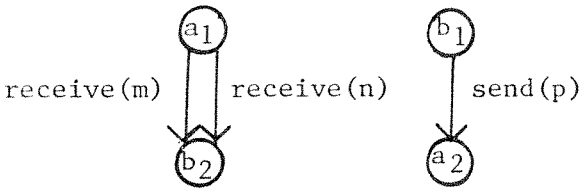
First, the initial global state [a,e,E,E] is in the schema $[a,c,(m.n)^* + n.(m.n)^*,E]$ and thus in H. Second, each directed cycle in P_3 and Q_3 has at least one node covered by the global states in H; hence H is a cover for P_3 and Q_3. It remains to show that each global state schema in H is closed. Figures 3c and 3d show the acyclic versions AP_3 and AQ_3 of P_3 and Q_3 with respect to H. Figure 2e shows the two schemas of AP_2 and AQ_2 reachable from $[b_1,e_1,(m.n)^+ + n.(m.n)^*,E]$; notice that the only schema with no other reachable schema is $[a_2,c_2,(m.n)^+ + n.(m.n)^*,E]$. Since $[a,c,(m.n)^* + n.(m.n)^*,E]$ is in H and since the language accepted by $(m.n)^+ + n.(m.n)^*$ is a subset of the language accepted by $(m.n)^* + n.(m.n)^*$, then $[b,e,(m.n)^+ + n.(m.n)^*,E]$ is closed. Similarly, it can be shown that $[a,c,(m.n)^* + n.(m.n)^*,E]$ is closed. Therefore, H is a closed cover for P_3 and Q_3; and the communication between P_3 and Q_3 will progress indefinitely by Theorem 1. []



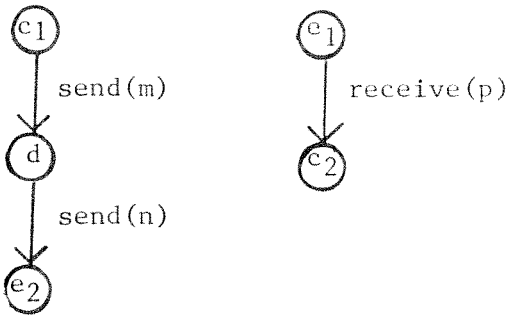
(a) P_3



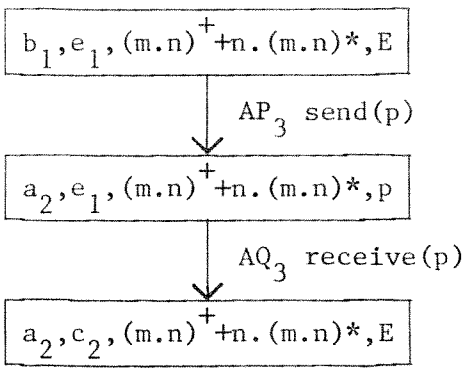
(b) Q_3



(c) AP_3



(d) AQ_3



(d) Proving that $[b_1,e_1,(m.n)^{+}+n.(m.n)^{*},E]$ is closed.

Figure 3 Example 3

VII. CONCLUDING REMARKS

The closed cover technique can be extended in a straightforward manner to verify progress for more than two communicating finite state machines. A comparison between this technique and usual state exploration is as follows.

The closed cover technique has two advantages over state exploration. First, the total number of global states generated when using a closed cover is usually less than those generated during state exploration (Example 1). The amount of saving depends on the communicating machine pair and on the selected closed cover. Second, the closed cover technique can be used to verify progress for machines with unbounded communications (Examples 2 and 3) whereas state exploration cannot be used in these cases.

On the other hand, state exploration has two advantages over the closed cover technique. First, to use state exploration one should determine or "guess" the capacities of the channels between the communicating machines. This seems much simpler than guessing a whole closed cover as required by the closed cover technique. Second, state exploration can be used to verify nonprogress, e.g., by showing deadlock states, while the closed cover technique cannot verify nonprogress.

There is an analogy between the closed cover technique and the assertion techniques to verify safety properties of sequential programs. A global state [i,j,x,y] in a closed cover can be viewed as an assertion stating that "if the execution of P reaches node i and

the execution of Q reaches node j, then the input channels of P and Q have x and y respectively". Also the condition that each directed cycle in P or Q must have at least one node covered by the closed cover is analogous to the condition that each cycle in a sequential program must have at least one assertion. Finally, the requirement that each global state in a closed cover must be closed is analogous to the requirement that each assertion before a block of statements must be sufficient to ensure the assertion after the block. This analogy between the two techniques should encourage further research to "blend" both techniques together to prove safety and progress properties for communicating sequential processes.

ACKNOWLEDGEMENT: The author is thankful to K. F. Carbone for her careful typing.

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

[4] M. G. Gouda and Y. Yu, "Designing deadlock-free and bounded communication protocols," Tech. Rep. 179, Dept. of Computer Sciences, Univ. of Texas at Austin, June 1981.

[5] J. E. Hopcroft and J. D. Ullman, Formal languages and their relation to automata, Addison-Wesley, Reading, MA, 1969.

[6] P. Merlin, and G. V. Bochmann, "On the construction of communication protocols and module specifications," Pub. 352 dept. dinformatique et de recherche operationnelle, Universitie de Montreal, Jan. 1980.

[7] C. A. Sunshine, "Interprocess communication protocols for computer networks," Ph.D. dissertation, Dept. of Comp. Sci., Stanford Univ., Stanford, CA, 1975.

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

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

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

APPENDIX: PROOFS

Proof of Lemma 1: We prove that the two reachability conditions R_1 and R_2 are satisfied.

Proof of R1: From (i), we have

(v) $x_1.s' = r.x_2$, and
(vi) $y_1.s = r'.y_2$

where s and s′ are the two sequences of sent messages along p and q respectively, and r and r′ are the two sequences of received messages along p and q respectively. From (iii), we have

(vii) $x_1.u' = v.x$
(viii) $y_1.u = v'.y$

where u and u′ are the two sequences of sent messages along $p[i_1,i]$ and $q[j_1,j]$ respectively, and v and v′ are the two sequences of received messages along $p[i_1,i]$ and $q[j_1,j]$ respectively. From (ii), we have

(ix) $s = u.t$,
(x) $s' = u'.t'$,
(xi) $r = v.w$, and
(xii) $r' = v'.w'$

where t and t′ are the two sequences of sent messages along $p[i,i_2]$ and $q[j,j_2]$ respectively, and w and w′ are the two sequences of received messages along $p[i,i_2]$ and $q[j,j_2]$ respectively.

$v.x.t' = x_1.u.t'$ from (vii)
$= x_1.s'$ from (x)
$= r.x_2$ from (v)
$= v.w.x_2$ from (xi)

$x.t' = w.x_2$

Similarly, we can show that $y.t = w'.y_2$ from (viii), (ix),(vi), and (xii). This proves R1.

Proof of R2: (by contradiction) Assume that there are two nodes i′ and

j′ with receiving outputs on the two paths $p[i,i_2]$ and $q[j,j_2]$ such that

(xiii) $x.f' = g$, and
(xiv) $y.f = g'$

where f and f′ are the two sequences of sent messages along p[i,i′] and q[j,j′] respectively, and g and g′ are the two sequences of received messages along p[i,i′] and q[j,j′] respectively.

$x_1.u.f' = v.x.f'$ from (vii)
$= v.g$ from (xiii)
and $y_1.u.f = v'.g'$ from (viii) and (xiv)

This contradicts (i). Thus R2 is valid. []

Proof of Lemma 2: (by induction on K)

Initial step (K=1): The lemma is correct since $[i_0,j_0,E,E]$ is in C.

Induction hypothesis (K=k-1): Let m and n be the (k-1)th covered nodes on paths p and q respectively. Then, there exist w′ and z′ such that

(v) [m,n,w′,z′] is in C, and
(vi) $[i_0,j_0,E,E]$---p,q-->[m,n,w′,z′]

From (i), (vi), and Lemma 1 we have

(vii) [m,n,w′.z′]---p,q-->[i,j,x,y]

Induction step (K=k): Let m_1 and n_1 be the input versions of m and n; and let r_2 and s_2 be the output versions of r and s respectively. Let $p_1[m_1,r_2]$ denote the path in AP which corresponds to p[m,r] in P, and $q_1[n_1,s_2]$ denote the path in AQ which corresponds to q[n,s] in Q. Now (vii) can be rewritten for AP and AQ as follows.

(viii) $[m_1,n_1,w',z']$---p_1,q_1-->[k,l,w",z"], and
(ix) no other global state of AP and AQ is reachable from [k,l,w",z"].

If [k,l,w",z"] ≠ $[r_2,s_2,w,z]$ then [m,n,w′,z′] is not closed

contradicting that C is closed. Therefore $[k,1,w'',z''] = [r_2,s_2,w,z]$ and (viii) must be rewritten as:

(x) $[m_1,n_1,w',z']$---p_1,q_1-->$[r_2,s_2,w,z]$

From (x) since $[m,n,w',z']$ is closed, then $[r,s,q,z]$ must be in C which proves (iii). Also, (x) can be rewritten for P and Q as follows.

(xi) $[m,n,w',z']$---p,q-->$[r,s,w,z]$

Thus (iv) is immediate from (vi) and (xi). []

Proof of Lemma 3: (by induction on K)

Initial step (K=1): The lemma is true since any path q starting from the initial node j_0 has at least one node covered by C, namely node j_0.

Induction step (K=k): (Proof is by contradiction) Assume that path q cannot be extended to node s which satisfies (iii) and (iv). In other words, there is a node r on the extended path q′ such that

(v) $[i_0,j_0,E,E]$---p,q'-->$[i,r,w,z]$, and

(vi) no other global state of P and Q is reachable from $[i,r,w,z]$, and

(vii) the path $q'[j,r]$ has exactly k-1 nodes covered by C. (This is because path p contains exactly K=k nodes and the lemma is assumed true when K=k-1 by the induction hypothesis.)

Let u and v be the (k-1)th and the kth covered nodes in p; and let u_1 and v_2 be the input and output versions of u and v respectively. Let $p_1[u_1,v_2]$ denote the path in AP which corresponds to the path p[u,v] in P. Similarly, let t be the (k-1)th covered node in q', and t_1 be the input version of t. Also, let $q_1'[t_1,r]$ denote the path in AQ which corresponds to path q'[t,r] in Q. Notice that since C is closed, the global state [u,t,x',y'] is in C by Lemma 2. From (v), (vi), and (vii), we have:

(viii) $[u_1,t_1,x',y']$---p_1,q_1-->$[v_2,r,w',z']$, and

(ix) no other global state of AP and AQ is reachable from $[v_2,r,w',z']$, and

(x) node r is not covered by C.

These three conditions contradict the fact that [u,t,x',y'] and hence C are closed. []

Proof of Theorem 1: (by contradiction) Let i_0 and j_0 be the initial nodes of P and Q respectively; and assume that the following two conditions are satisfied.

(i) $[i_0,j_0,E,E]$---p,q-->[i,j,x,y]

(ii) All reachable states from [i,j,x,y] are of the form [i,k,w,z]. In other words [i,j,x,y] is a blocking state for P.

Path p must have more covered nodes than path q, otherwise according to Lemma 3 (in its reversed form) path p can be extended in violation of condition (ii). Assume that path p has K covered nodes and that its Kth covered node is m. If path q does not have K covered nodes then according to Lemma 3, it can be extended to node n such that the extended path q' satisfies the following two conditions.

(iii) $[i_0,j_0,E,E]$---p,q'-->[i,n,x',y'], and

(iv) node n is the Kth covered node in path q′.

From (i), (iii), and Lemma 1 we have,

(v) [i,j,x,y]---p,q′-->[i,n,x′,y′]

And from (ii) and (v) we have

(vi) All reachable states from [i,n,x′,y′] are of the form [i,t,w,z].

Since m and n are the Kth covered nodes in paths p and q′, then from Lemma 2 we have

(vii) $[i_0,j_0,E,E]$---p,q′-->[m,n,w′,z′]

From (vii), and Lemma 1 we have

(viii) [m,n,w′,z′]---p,q′-->[i,n,x′,y′]

Let m_1 and n_1 be the input versions of m and n respectively. Let $p_1[m_1,i]$ denote the path in AP which corresponds to path p[m,i] in P. Similarly, let $q_1'[n_1,n_1]$ denote the path in AQ which corresponds to path q′[n,n] in Q. The two conditions (vi) and (viii) can now be rewritten for AP and AQ as follows:

(viii) $[m_1,n_1,w',z']$---p_1,q_1'-->$[i,n_1,x',y']$

(ix) All reachable states from $[i,n_1,x',y']$ are of the form [i,t,w,z].

Since i is not covered by C (otherwise path p has K+1 covered nodes), then [m,n,w′,z′] is not closed implying that C is not closed. Contradiction. []