Closed Covers: To Verify Progress for Communicating Finite State Machines

MOHAMED G. GOUDA, MEMBER, IEEE

Abstract—Consider communicating finite state machines which exchange messages over unbounded FIFO channels. We discuss a technique to verify that the communication between a given pair of such machines will progress indefinitely; this implies that the communication is free from 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 (along with other conditions) are satisfied: 1) the initial global state is in that set; and 2) 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 show that the existence of a closed cover for a communicating pair is sufficient to guarantee indefinite communication progress. We also show that in many practical instances, if the communication is guaranteed to progress indefinitely, then the existence of a closed cover is necessary.

Index Terms—Communicating finite state machines, communication progress, communication protocols, verification techniques.

I. Introduction

A communicating finite state machine is an *abstraction* of a process which communicates, with the environment or with other processes, by sending and receiving messages over unbounded, one-directional, FIFO channels. The abstraction is achieved by ignoring the internal data structures and internal operations of the process, and representing the process only by its external behavior, i.e., by all possible sequences of its sending and receiving operations.

Communicating finite state machines are useful in modeling [12], analysis [1], [2], [7], [14], [15], and synthesis [3], [4], [6], [10], [16] of communication protocols. They are also useful in modeling and analysis of some resource sharing schemes [9], and in distributed systems.

The problem of whether the communication between an arbitrary pair of communicating finite state machines will continue to progress indefinitely is undecidable in general [2]. So far, state exploration [11], [13] is the only available technique to verify indefinite progress of communication between two (or more) machines. Unfortunately, this technique has two apparent shortcomings. First, the number of generated global states is usually "large," making it difficult to use for practical purposes. Second, state exploration can be used only for bounded communications, i.e., when there is an upper bound on the number of messages which can exist simultaneously in each channel.

Manuscript received January 29, 1982; revised December 16, 1983.
The author is with the Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712.

In this paper, we propose another technique to verify indefinite progress for two communicating finite state machines. We also show that the two shortcomings of state exploration are remedied to some degree by this new technique. (For simplicity, we limit the discussion to the case of two communicating machines. The proposed technique, however, can be used to verify indefinite progress for any number of communicating machines.)

II. Communicating Finite State Machines

A *communicating finite state machine* P is a directed labeled graph where each edge is labeled either "send(m)" or "receive(m)" for some *message* m from a finite set M. An edge labeled "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 directed path in P, which starts from node i, ends at node j, and consists of the directed edges $e_1, e_2, \cdots, e_v$. 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 $e_1, e_2, \cdots, e_u$, while $p[k, j]$ denotes the path which consists of the edges $e_{u+1}, \cdots, e_v$. Therefore, path p can be denoted as $p[i, j]$.

The *sequence of sent messages* along a directed path p whose edges are $e_1, \cdots, e_v$ in P is defined as $s_1 \cdot s_2 \cdot \cdots \cdot s_v$, where "$\cdot$" is the usual string concatenation operator, and

$s_1 = m$ if e_1 is labeled "send(m)"
$= E$ (the empty string) otherwise.

(Notice that $x \cdot E \cdot y = x \cdot y$.)

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

$r_1 = m$ if e_1 is labeled "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

0098-5589/84/1100-0846$01.00

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)

$$x_1 \cdot s' = r \cdot x_2$$

and

$$y_1 \cdot s = r' \cdot y_2$$

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

$$x_1 \cdot u' = v$$

and

$$y_1 \cdot u = v'$$

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 $p[i_i, i]$ and $q[j_j, j]$, respectively.

Notation:

$$[i_1, j_1, x_1, y_1] - p, q \rightarrow [i_2, j_2, x_2, y_2]$$

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) $[i_1, j_1, x_1, y_1] - p, q \rightarrow [i_2, j_2, x_2, y_2]$, and
ii) nodes i and j are any two nodes in paths p and q, respectively, and
iii) $[i_1, j_1, x_1, y_1] - p, q \rightarrow [i, j, x, y]$

then

iv) $[i, j, x, y] - p, q \rightarrow [i_2, j_2, x_2, y_2]$. □

are two directed paths p and q in P and Q, respectively, such that $s_1 - p, q \rightarrow s_2$.

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

A global state s is called a *blocking state for* P iff any state s' reachable from s is such that $s - p, q \rightarrow s'$ where p is an empty directed path; i.e., it has no directed edges. Similarly, a blocking state for Q can be defined. Informally, a blocking state for P is one after 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 [6].

The communication between two communicating finite state machines P and Q will *progress indefinitely* iff 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 labeled 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 directed labeled 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 $s_1 - p, q \rightarrow s_2$, 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 $s_1 - p, q \rightarrow s_2$.

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 two conditions are 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.

a) If

i) $[i_1, j_1, x, y] - p, q \rightarrow [k_2, l_2, w, z]$, 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.

b) If

i) k_2 is the output version of some node k in P, and
ii) l_2 is the output version of some node l in Q

then either

$$[(x \cdot s_q < r_p) \text{ and } (y \cdot s_p < r_q)]$$

or

$$[\text{not } (x \cdot s_q < r_p) \text{ and not } (y \cdot s_p < r_q)]$$

where "$\cdot$" is the string concatenation operator, "$<$" denotes "is a proper prefix of," s_p and s_q are the two sequences of sent messages along p and q, respectively, and r_p and r_q are the two sequences of received messages along p and q, respectively.

A cover set C of P and Q is called a *closed cover* iff it satisfies the following two conditions: 1) the initial global state of P and Q is in C; 2) 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) $[i_0, j_0, E, E] - p, q \rightarrow [i, j, x, y]$, 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) $[i_0, j_0, E, E] - p, q \rightarrow [r, s, w, z]$. □

Lemma 3: If

i) $[i_0, j_0, E, E] - p, q \rightarrow [i, j, x, y]$, and
ii) path p has K nodes covered by C, and path q has L nodes covered by C such that $K > L$

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

iii) $[i_0, j_0, E, E] - p, q' \rightarrow [i, s, w, z]$, 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 restate Lemma 3 in this reversed form and to prove it using a similar proof to that in 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.

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

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

3) 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.

d) Prove that for any path p that starts at the input version i_1 of i and ends at some output version in AP, and for any path q that starts at the input version j_1 of j and ends at some output version in AQ, either

$$[(x \cdot s_q < r_p) \text{ and } (y \cdot s_p < r_q)]$$

or

$$[\text{not } (x \cdot s_q < r_p) \text{ and not } (y \cdot s_p < r_q)]$$

where s_p and s_q are the sequences of sent messages along p and q, respectively, and r_p and r_q are the sequences of received messages along p and q, respectively.

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, labeled "send(m)," from node k (or l) to node r in AP (or AQ), then $[r, l, w, z \cdot m]$ (or $[k, r, w \cdot m, z]$, 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, labeled "receive(m)," from node k (or l) to node r in AP (or AQ), and if $w = m \cdot s$ (or $z = m \cdot s$), 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: Fig. 1(a) and (b) shows 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. Fig. 1(c) and (d) shows the acyclic versions AP_1 and AQ_1 of P_1 and Q_1 with respect to C_1.

The following two steps are needed to show that $[a, e, E, E]$ is closed.

a) Fig. 1(e) shows all the global states of AP_1 and AQ_1 reachable from $[a_1, e_1, E, E]$. $[c_2, g_2, E, E]$ is the only state with no other reachable state, and $[c, g, E, E]$ is in C_1.

b) There exists one path p that starts from a_1 and ends at some output version (c_2) in AP_1. The sequences s_p and r_p of

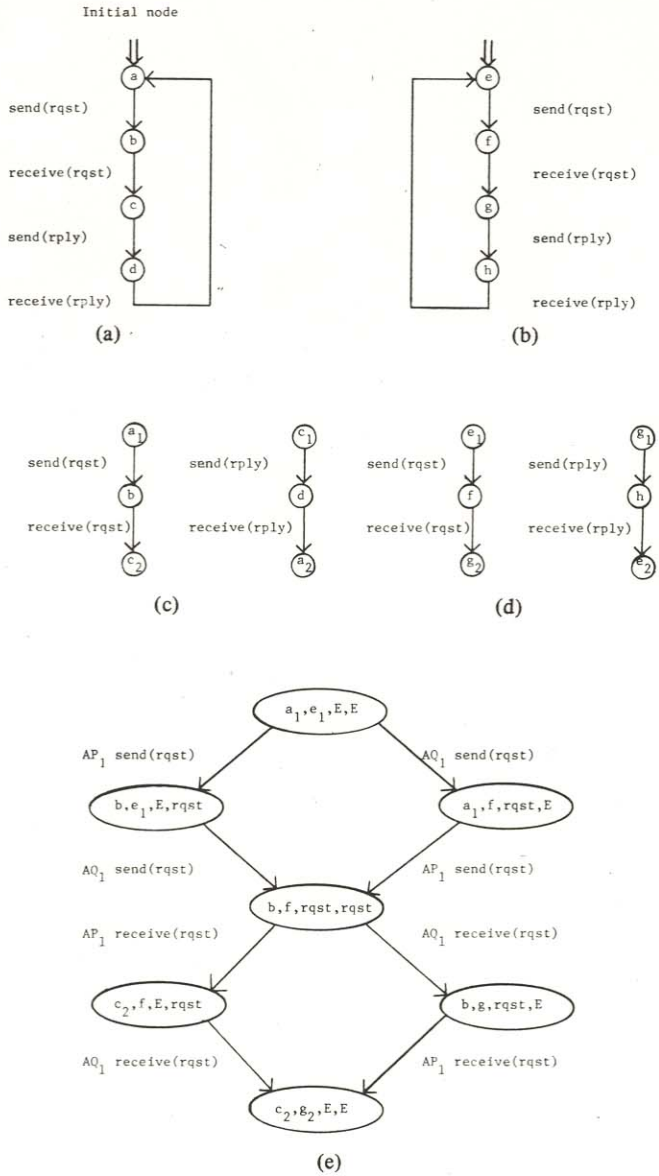

Fig. 1. Example 1. (a) P_1. (b) Q_1. (c) AP_1. (d) AQ_1. (e) Proving that $[a, e, E, E]$ is closed.

sent and received messages along p are "rqst" and "rqst." Also there exists one path q that starts from e_1 and ends at some output version (g_2) in AQ_1. The sequences s_q and r_q of sent and received messages along q are "rqst" and "rqst." Therefore,

$$\text{not } (s_q < r_p) \text{ and not } (s_p < r_q).$$

Similarly, we can show that $[c, g, E, E]$ is closed. Therefore, 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:

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

2) 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.

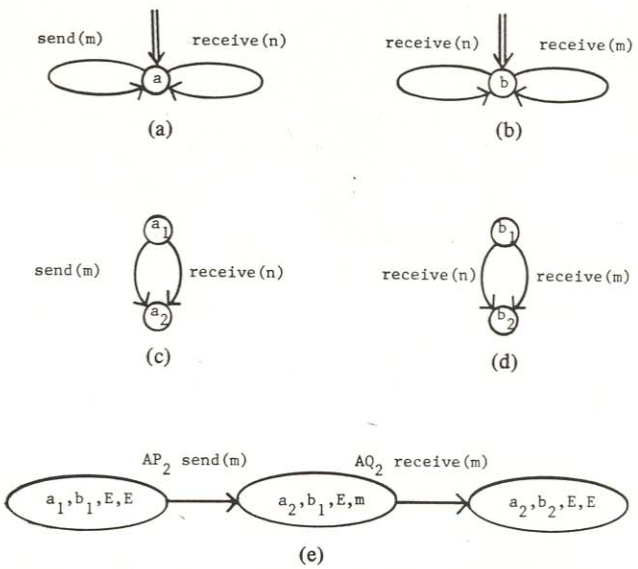

Fig. 2. Example 2. (a) P_2. (b) Q_2. (c) AP_2. (d) AQ_2. (e) Proving that $[a, b, E, E]$ is closed.

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: Fig. 2(a) and (b) shows the 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.

a) Fig. 2(c) and (d) shows the acyclic versions AP_2 and AQ_2 of P_2 and Q_2 with respect to C_2. Fig. 2(e) shows all the global states of AP_2 and AQ_2 reachable from $[a_1, b_1, E, E]$; the only state with no other reachable state is $[a_2, b_2, E, E]$.

b) There are two paths p_1 and p_2 that start from a_1 and end at some output version (a_2) in AP_2. There are also two paths p_3 and p_4 that start from b_1 and end at some output version (b_2) in AQ_2. Let s_i (r_i) be the sequence of sent (received) messages along path p_i, $i = 1, \cdots, 4$. Hence

$$s_1 = m, \quad r_1 = E \text{ (the empty string)}.$$

$$s_2 = E, \quad r_2 = n,$$

$$s_3 = E, \quad r_3 = n,$$

$$s_4 = E, \quad r_4 = m.$$

Therefore,

$$\text{not } (s_1 < r_3) \text{ and not } (s_3 < r_1),$$

$$\text{not } (s_1 < r_4) \text{ and not } (s_4 < r_1),$$

$$(s_2 < r_3) \text{ and } (s_3 < r_2),$$

$$(s_2 < r_4) \text{ and } (s_4 < r_2).$$

This shows that $[a, b, E, E]$ is closed, and C_2 is a closed cover for P_2 and Q_2. Hence, 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 case. □

VI. Necessity of Closed Covers

Theorem 1 implies that the existence of a closed cover is a *sufficient* condition for two communicating machines to progress indefinitely. In this section we show that the existence of a closed cover is also *necessary* for indefinite progress in many practical instances. These practical instances are characterized by the requirement that each node in either machine be "reachable"; the concept of reachable nodes is defined next.

Let P and Q be two communicating finite state machines with the same set M of messages. A node i in P is called *reachable* iff there are one node j in Q, and two message sequences x and y over M such that the global state $[i, j, x, y]$ of P and Q is reachable. Similarly, a reachable node in Q can be defined.

The requirement that each node in P or Q be reachable is not severe at all. This is because unreachable nodes can be removed along with their input and output edges without affecting the communication between P and Q. Moreover, anyone who constructs two communicating machines P and Q is expected to prove that each node k in P or Q is reachable. This can be done by showing a finite sequence $<s_0, s_1, \cdots, s_r>$ of reachable states of P and Q such that s_0 is the initial state of P and Q, and for $i = 0, \cdots, r-1$, s_{i+1} is reachable from s_i over one edge in P or Q, and s_r is either $[k, j, x, y]$ or $[i, k, x, y]$, depending on whether node k is in P or Q, respectively. A proof for the following theorem is the Appendix.

Theorem 2: Let P and Q be communicating finite state machines whose nodes are all reachable. If the communication between P and Q is guaranteed to progress indefinitely, then P and Q have a closed cover. □

The proof of Theorem 2, in the Appendix, explains how to construct a closed cover for two communicating machines where all the nodes are reachable. In particular, it shows the following.

a) Each node in the two machines should be covered by the closed cover.

b) Each directed edge in the acyclic versions of the two machines should be from an input version to an output version; i.e., the longest path is of length one.

c) Each state in the closed cover is of the form $[i, j, x, y]$ where $|x| = |y|$, and $|x|$ is the number of messages in string x.

These ideas are illustrated in the following example.

Example 3: Fig. 3 shows two communicating finite state machines P_3 and Q_3 that model the call establishment/clear procedure in X.25 [6].

P_3 models the DTE.
Q_3 models the DCE.
Message *cr* models a call request.
Message *cc* models a call connected.
Message *ci* models an incoming call.

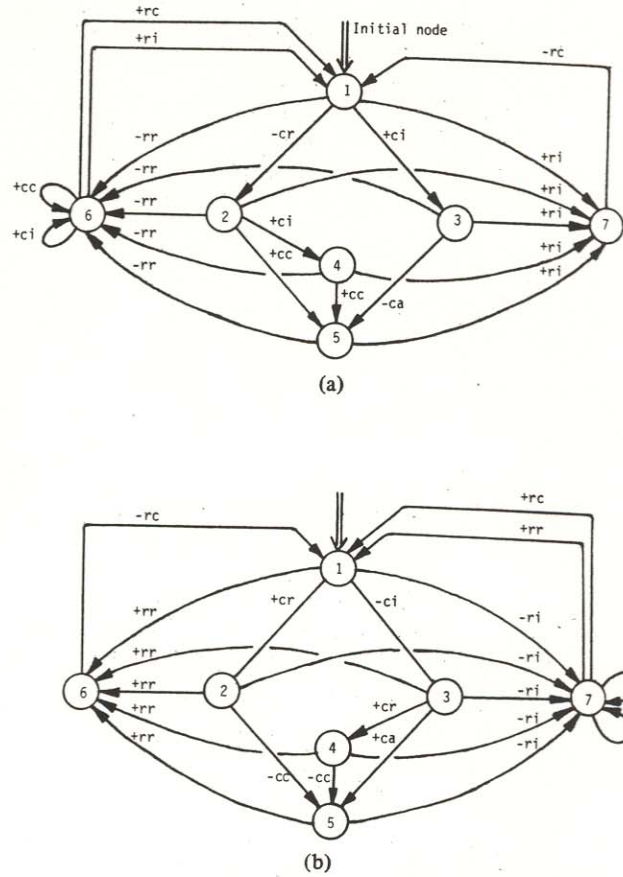

Fig. 3. The call establishment/clear protocol in X.25. (Notation: $-m$ = send(m), $+m$ = receive(m).) (a) P_3. (b) Q_3.

Message *ca* models a call accepted.
Message *rr* models a clear request.
Message *rc* models a clear confirm.
Message *ri* models a clear indication.

It is straightforward to show that the following set is a closed cover for P_3 and Q_3:

$$C = \{[1,1,E,E], [2,2,E,E], [2,3,ci,cr], [2,7,ri,cr], [3,3,E,E], [6,3,ci,rr], [6,6,E,E], [6,7,ri,rr], [7,7,E,E], [5,5,E,E], [6,5,cc,rr], [4,4,E,E], [4,7,ri,cr], [6,4,ci,rr], [6,7,ci \cdot ri, cr \cdot rr], [5,7,ri,ca]\}.$$ □

Based on Theorem 2, if one fails to construct a closed cover for two communicating finite state machines P and Q, then one of the following two reasons is responsible for this failure.

a) One of the reachable states of P and Q is a blocking state, and the communication between P and Q is not guaranteed to progress indefinitely. In this case, the failure to find a closed cover should be appreciated as a failure to prove a wrong theorem.

b) Some of the nodes in P or Q are unreachable. In this case, the failure to find a closed cover should be regarded as a signal that either P or Q contains redundant nodes and edges.

Actually, there is a third, but less obvious, reason for failing to find a closed cover. If a closed cover consists of an infinite number of global states, then the above straightforward techniques cannot be used to represent it and to verify that it is

indeed a closed cover. To remedy this, we discuss next a technique to represent and verify infinite closed covers.

VII. Infinite Closed Covers

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 [7] 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 state *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 $[k, l, W', Z']$ in H such that the language accepted by W (or Z) is a subset of the language accepted by W' (or Z', respectively).

d) Prove that for any path p that starts at the input version i_1 of i and ends at some output version, and for any path q that starts at the input version j_1 of j and ends at some output version in AQ, and for any string x in X and any string y in Y, either

$$[(x \cdot s_q < r_p) \text{ and } (y \cdot s_p < r_q)]$$

or

$$[\text{not } (x \cdot s_q < r_p) \text{ and not } (y \cdot s_p < r_q)]$$

where s_p and s_q are the sequences of sent messages along p and q, respectively, and r_p and r_q are the sequences of received messages along p and q, 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, labeled "send(m)," from node k (or l) to node r in AP (or AQ), then the global state schema $[r, l, W, Z \cdot m]$ (or $[k, r, W \cdot m, Z]$, respectively) is in $S[i_1, j_1, X, Y]$.

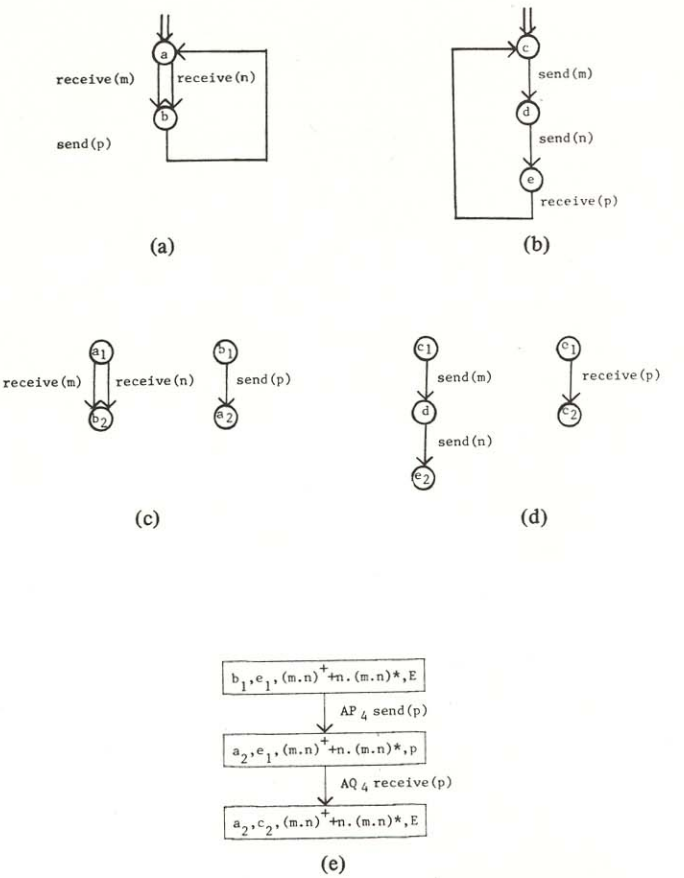

Fig. 4. Example 4. (a) P_4. (b) Q_4. (c) AP_4. (d) AQ_4. (e) Proving that $[b_1, e_1, (m \cdot n)^+ + n \cdot (m \cdot n)^*, E]$ is closed.

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

Example 4: Fig. 4(a) and (b) shows two communicating finite state machines P_4 and Q_4 whose initial nodes are "a" and "c," respectively. Consider the following set H of global state schemas of P_4 and Q_4.

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

First, the initial global state $[a, c, E, E]$ is in the schema $[a, c, (m \cdot n)^* + n \cdot (m \cdot n)^*, E]$ and, thus, in H. Second, each directed cycle in P_4 and Q_4 has at least one node covered by the global states in H; hence, H is a cover for P_4 and Q_4. It remains to show that each global state schema in H is closed. Fig. 4(c) and (d) shows the acyclic versions AP_4 and AQ_4 of P_4 and Q_4 with respect to H.

To show that $[b, e, (m \cdot n)^+, n \cdot (m \cdot n)^*, E]$ is closed, we follow the next two steps.

a) Fig. 4(e) shows the two schemas of AP_4 and AQ_4 reachable from $[b_1, e_1, (m \cdot n)^+ + n \cdot (m \cdot n)^*, E]$. Notice that the only schema with no other reachable schema is $[a_2, c_2, (m \cdot n)^+ + n \cdot (m \cdot n)^*, E]$, and that $[a, c, (m \cdot n)^* + n \cdot (m \cdot n)^*, E]$ is in H, where $(m \cdot n)^+ + n(m \cdot n)^*$ is a subset of $(m \cdot n)^* + n \cdot (m \cdot n)^*$.

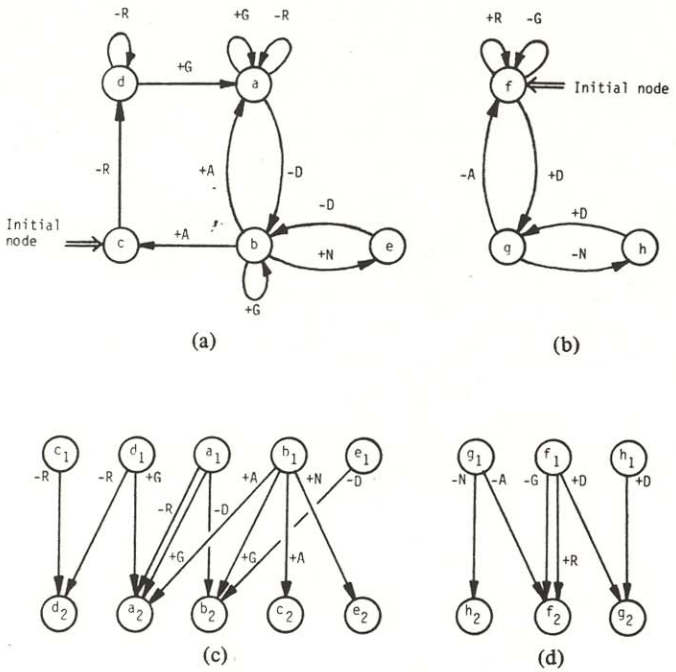

Fig. 5. Example 5. (a) P_5. (b) Q_5. (c) AP_5. (d) AQ_5.

b) There exists one path q_1 that starts from the input version b_1 of b, and ends at some output version (a_2) in AP_4. The sequences s_1 and r_1 of sent and received messages along a_1 are "p" and "E," respectively. Also there exists one path q_2 that starts from the input version e_1 of e, and ends at some output version (c_2) in AQ_4. The sequences s_2 and r_2 of sent and received messages along q_2 are "E" and "p," respectively. Therefore,

$$\text{not}(((m \cdot n)^+ + n(m \cdot n)^*) \cdot s_2 < r_1) \quad \text{and not} \quad (s_1 < r_2).$$

Similarly, we can show that $[a, c, (m \cdot n)^* + n \cdot (m \cdot n)^*, E]$ is closed. Hence, H is a closed cover for P_4 and Q_4, and the communication between P_4 and Q_4 will progress indefinitely by Theorem 1. □

It is also possible to use the characterization suggested in the proof of Theorem 2 to construct infinite closed covers. One example of this is discussed next.

Example 5: Consider the two communicating finite state machines P_5 and Q_5 in Fig. 5(a) and (b). P_5 models a sender that sends data messages to a receiver Q_5. For each sent data message, the receiver Q_5 should respond by either a negative or a positive acknowledgment. When the sender receives a negative acknowledgment, it resends the last data message; when it receives a positive acknowledgment, it sends a new data message. But before the sender can send a new data message it must send a request-to-send to the receiver, and receive a grant-to-send. The exchanged messages between P_5 and Q_5 in Fig. 5 have the following meaning.

D denotes a data message.
A denotes a positive acknowledgment.
N denotes a negative acknowledgment.
R denotes a request-to-send message.
G denotes a grant-to-send message.

Following the characterization of closed covers in the proof of Theorem 2, each node in the two machines should be covered by the closed cover. This yields the acyclic versions AP_5 and AQ_5 in Fig. 5(c) and (d), respectively. It is straightforward to show that the following set is an infinite closed cover of P_5 and Q_5:

$$H = \{[c, f, E, E], [d, f, G^n, R^n], [a, f, G^n, R^n], [b, f, G^n, R^{n-1} \cdot D], [b, g, E, E], [e, h, E, E]\}.$$

Notice that each element in H is a global schema of P_5 and Q_5. For instance, the element $[d, f, G^n, R^n]$ is the infinite set $\{[d, f, E, E], [d, f, G, R], [d, f, G^2, R^2], \cdots\}$ of states of P_5 and Q_5. Similarly, the element $[c, f, E, E]$ is the set $\{[c, f, E, E]\}$. □

VIII. 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, 4, and 5), 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. (Recall that the proof of Theorem 2 has suggested a characterization that can help in constructing closed covers in some cases. See Examples 3 and 5.) Second, state exploration can be used to verify nonprogress, e.g., by showing deadlock states, while the closed cover technique cannot verify nonprogress. (On the other hand, the more one fails to construct a closed cover for a given network of communicating machines, the more he should question his belief in the network's indefinite progress.)

The technique of closed covers has already proven useful in constructing communicating machines with guaranteed indefinite progress. Two construction methodologies that are based on this technique have been discussed in [3], [4]. Also, this technique has been used successfully to prove liveness for networks of communicating finite state machines [5].

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, 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.

APPENDIX
PROOFS

Proof of Lemma 1: We prove that the two reachability conditions $R1$ and $R2$ are satisfied.

Proof of R1: From i), we have

v) $x_1 \cdot s = r \cdot x_2$, and
vi) $y_1 \cdot s' = r' \cdot y_2$

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) $x_1 \cdot u = v \cdot x$
viii) $y_1 \cdot u' = v' \cdot y$

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) $s = u \cdot t$,
x) $s' = u' \cdot t'$,
xi) $r = v \cdot w$, and
xii) $r' = v' \cdot w'$

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.

$$\begin{aligned} v \cdot x \cdot t &= x_1 \cdot u \cdot t && \text{from vii)} \\ &= x_1 \cdot s && \text{from ix)} \\ &= r \cdot x_2 && \text{from v)} \\ &= v \cdot w \cdot x_2 && \text{from xi)} \end{aligned}$$

$x \cdot t = w \cdot x_2$.

Similarly, we can show that $y \cdot t' = w' \cdot y_2$ from vii), x), 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) $x \cdot f = g$, and
xiv) $y \cdot f' = g'$

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.

$$\begin{aligned} x_1 \cdot u \cdot f &= v \cdot x \cdot f && \text{from vii)} \\ &= v \cdot g && \text{from xiii)} \end{aligned}$$

and

$y_1 \cdot u' \cdot f' = v' \cdot g'$ from vii), 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) $[i_0, j_0, E, E] - p, q \rightarrow [m, n, w', z']$.

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

vii) $[m, n, w', z'] - p, q \rightarrow [i, j, x, y]$.

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 let $q_1[n_1, s_2]$ 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) $[m_1, n_1, w', z'] - p_1, q_1 \rightarrow [k, l, w'', z'']$, and
ix) no other global state of AP and AQ is reachable from $[k, l, w'', z'']$.

If $[k, l, w'', z''] \neq [r_2, s_2, w, z]$ then $[m, n, w', z']$ is not closed, contradicting that C is closed. Therefore, $[k, l, w'', z''] = [r_2, s_2, w, z]$ and viii) must be rewritten as

x) $[m_1, n_1, w', z'] - p_1, q_1 \rightarrow [r_2, s_2, w, z]$.

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) $[m, n, w', z'] - p, q \rightarrow [r, s, w, z]$.

Thus, iv) is immediate from vi) and xi). □

Proof of Lemma 3 (by induction on K):

Initial step (K = 2) (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) $[i_0, j_0, E, E] - p, q' \rightarrow [i, r, w, z]$, and
vi) state $[i, r, w, z]$ is a blocking state for Q, and
vii) the path $q'[j_0, r]$ has exactly one node covered by C. (This is because $2 = K > L = 1 > 0$.)

Let v be the second covered node in p, and let i_1 and v_2 be the input and output versions of i_0 and u_2, respectively. Let $p_1[i_1, v_2]$ denote the path in AP which corresponds to the path $p[i_0, v]$ in P. Similarly, let j_1 be the input version of j_0; also, let $q_1[j_1, r]$ denote the path in AQ which corresponds to path $q'[j_0, r]$ in Q. Notice that since C is closed, the global state $[i_0, j_0, E, E]$ is in C. From v), vi), and vii), we have

viii) $[i_1, j_1, E, E] - p_1, q_1 \rightarrow [v_2, r, w, z']$, 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.

This contradicts the fact that $[i_0, j_0, E, E]$ is closed.

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

xi) $[i_0, j_0, E, E] - p, q' \rightarrow [i, r, w, z]$, and
xii) state $[i, r, w, z]$ is a blocking state for Q, and
xiii) the path $q'[j_0, r]$ has exactly $k - 1$ nodes covered by C. (This is because path p contains exactly $K = k$ covered 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 xi), xii), and xiii), we have

xiv) $[u_1, t_1, x', y'] - p_1, q_1 \rightarrow [v_2, r, w', z']$, and
xv) no other global state of AP and AQ is reachable from $[v_2, r, w', z']$, and
xvi) 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): We show that no reachable state is a blocking state for P. (Proving that no reachable state is a blocking state for Q is similar). 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) $[i_0, j_0, E, E] - p, q \rightarrow [i, j, x, y]$.
ii) State $[i, j, x, y]$ is a blocking state for P.

From Lemma 3 (in its reversed form), path p cannot have fewer covered nodes than path q; otherwise, p can be extended in violation of condition ii). Hence, there are two cases to consider.

a) p has more covered nodes than q: Assume that path p has K covered nodes and that its Kth covered node is m. Since q does not have K covered nodes, then according to Lemma 3, q can be extended to node n such that the extended path q' satisfies the following two conditions:

iii) $[i_0, j_0, E, E] - p, q' \rightarrow [i, n, x', y']$, and
iv) node n is the Kth covered node in path q'.

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

v) $[i, j, x, y] - p, q' \rightarrow [i, n, x', y']$

and from ii) and v) we have

vi) State $[i, n, x', y']$ is a blocking state for P.

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

vii) $[i_0, j_0, E, E] - p, q' \rightarrow [m, n, w', z']$.

From vii) and Lemma 1 we have

viii) $[m, n, w', z'] - p, q' \rightarrow [i, n, x', y']$.

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 empty path in AQ which corresponds to the empty path $q'[n, n]$ in Q. The two conditions vi) and viii) can now be rewritten for AP and AQ as follows.

viii) $[m_1, n_1, w', z'] - p_1, q_1 \rightarrow [i, n_1, x', y']$
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.

b) p has the same number of covered nodes as q: Assume that each of p and q has K covered nodes, and let m and n be the Kth covered nodes in p and q, respectively. Then from Lemma 2, we have

x) $[i_0, j_0, E, E] - p, q \rightarrow [m, n, w, z]$.

From x) and Lemma 1, we have

xi) $[m, n, w, z] - p, q \rightarrow [i, j, x, y]$.

A similar argument as in the above case can be used to establish that $[m, n, w, z]$ (and, hence, C) is not closed; contradiction. □

Proof of Theorem 2: Let C be the set of all reachable states $[i, j, x, y]$ of P and Q where $|x| = |y|$, and $|x|$ is the number of messages in string x. We show that C satisfies the three conditions of a closed cover.

i) The initial global state of P and Q is in C.

ii) To show that each cycle in P or Q has at least one node covered by C, we show that each node in P or Q is covered by C. Let i be a node in P. Since i is reachable, then there exists a reachable state $[i, j, x, y]$; i.e.,

$$[i_0, j_0, E, E] - p, q \rightarrow [i, j, x, y],$$

where i_0 and j_0 are the initial nodes of P and Q, respectively. Let $|p|$ be the number of edges in path p; there are three cases to consider.

a) $|p| = |q|$: In this case, $|x| = |y|$ and $[i, j, x, y]$ is in C; i.e., node i is covered by C.

b) $|p| < |q|$: Let q' be the prefix of q such that $|p| = |q'|$. Thus, $[i_0, j_0, E, E] - p, q' \rightarrow [i, j', x', y']$, where $|x'| = |y'|$ and $[i, j', x', y']$ is in C; i.e., node i is covered by C.

c) $|p| > |q|$: Extend q to q' in any possible way such that $|p| = |q'|$, and $[i_0, j_0, E, E] - p, q' \rightarrow [i, j', x', y']$. (This is possible since the communication between P and Q is guaranteed to progress indefinitely.) Then $|x'| = |y'|$ and $[i, j', x', y']$ is in C; i.e., node i is covered by C.

iii) It remains now to show that each state $[i, j, x, y]$ in C is closed.

a) We have shown in part ii) above that each node in P or Q is covered by C. The acyclic version AP of P with respect to C is constructed by replacing each node i in P with an input version i_1 and an output version i_2; hence, each directed edge in AP is from an input version to an output version. Similarly, the acyclic version AQ of Q with respect to C is constructed

by replacing each node j in Q with an input version j_1 and an output version; hence, each directed edge in AQ is from an input version to an output version. Since $[i, j, x, y]$ is in C, then $|x| = |y|$. If AP and AQ start at state $[i_1, j_1, x, y]$, and since M and N are guaranteed to progress indefinitely, then AP and AQ must reach a state $[k_2, l_2, w, z]$ where $|w| = |z|$. Therefore, $[k, l, w, z]$ is in C and $[i, j, x, y]$ is closed.

b) From a, any path p that starts from i_1 and ends at some output version in AP consists of one directed edge. Similarly any path q that starts from j_1 and ends at some output version in AQ consists of one directed edge. Let s_p and r_p be the two sequences of sent and received messages along p; exactly one of s_p and r_p is the empty sequence "E." Let s_q and r_q be the sequences of sent and received messages along q; exactly one of s_q and r_q is "E." There are two cases to consider.

1) $s_p = s_q = E$:

If $x = E = y$ then $(x \cdot s_q < r_p)$ and $(y \cdot s_p < r_q)$.

If $x \neq E \neq y$ then not $(x \cdot s_q < r_p)$ and not $(y \cdot s_p < r_q)$.

2) $s_p \neq E$ or $s_q \neq E$:

not $(x \cdot s_q < r_p)$ and not $(y \cdot s_p < r_q)$.

This completes the proof that $[i, j, x, y]$ is closed. □

Acknowledgment

The author is thankful to one referee for pointing out a missing case in the proof of Theorem 1. He is also thankful to K. F. Carbone for her careful typing.

References

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

[2] D. Brand and P. Zafiropulo, "On communicating finite-state machines," *J. Ass. Comput. Mach.*, vol. 30, pp. 361–371, Apr. 1983.

[3] C. H. Chow, M. G. Gouda, and S. S. Lam, "A discipline for constructing multi-phase communication protocols," Dep. Comput. Sci., Univ. Texas, Tech. Rep.-233, June 1983.

[4] M. G. Gouda, "An example for constructing communicating machines by step-wise refinement," in *Proc. 3rd IFIP Workshop Protocol Specif., Testing, Verif.*, H. Rudin and C. H. West, Eds. New York: North-Holland, 1983, pp. 63–74.

[5] M. G. Gouda and C. K. Chang, "A technique for proving liveness of communicating finite state machines with examples," presented at 3rd ACM Symp. Principles Distrib. Comput., Aug. 1984.

[6] M. G. Gouda and Y. T. Yu, "Synthesis of communicating finite state machines with guaranteed progress," *IEEE Trans. Commun.*, vol. COM-32, pp. 779–788, July 1984.

[7] ——, "Protocol validation by maximal progress state exploration," *IEEE Trans. Commun.*, vol. COM-32, pp. 94–97, Jan. 1984.

[8] J. E. Hopcroft and J. D. Ullman, *Formal Languages and Their Relation to Automata.* Reading, MA: Addison-Wesley, 1969.

[9] D. A. Measce, G. J. Popek, and R. R. Muntz, "A locking protocol for resource coordination in distributed databases," *ACM Trans. Database Syst.*, vol. 4, pp. 103–138, June 1980.

[10] P. Merlin and G. V. Bochmann, "On the construction of submodule and communication protocols," *ACM TOPLAS*, vol. 5, pp. 1–25, Jan. 1983.

[11] C. A. Sunshine, "Interprocess communication protocols for computer networks," Ph.D. dissertation, Dep. Comput. Sci., Stanford Univ., Stanford, CA, 1975.

[12] ——, "Formal modeling of communication protocols," U.S.C./Inform. Sci. Inst., Marina Del Rey, CA, Res. Rep. 81-89, Mar. 1981.

[13] C. H. West, "An automated technique of communications protocol validation," *IEEE Trans. Commun.*, vol. COM-26, pp. 1271–1275, Aug. 1978.

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

[15] ——, "Unboundedness detection for a class of communicating finite state machines," *Inform. Processing Lett.*, vol. 17, pp. 235–240, Dec. 1983.

[16] P. Zafiropulo *et al.*, "Towards analyzing and synthesizing protocols," *IEEE Trans. Commun.*, vol. COM-28, pp. 651–661, Apr. 1980.

Mohamed G. Gouda (S'76–M'77) 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.