Deadlock Detection in Communicating Finite State Machines by Even Reachability Analysis

Wuxu Peng
Department of Computer Science
Southwest Texas State University
San Marcos, TX 78666

Abstract

Fair reachability *is a very useful technique in detecting errors of deadlocks and unspecified receptions in networks of communicating finite state machines (CFSMs) consisting of two machines. The paper extends the classical fair reachability technique, which is only applicable to the class of two-machine CFSMs, to the general class of CFSMs. For bounded CFSMs, the extended fair reachability technique reduces by more than one half the total number of reachable global states that have to be searched in verifying freedom from deadlocks. The usefulness of the new reachability technique, called* even reachability, *is demonstrated through two examples.*

1 Introduction

A network of *communicating finite state machines* (CFSM) consists of a set of finite state machines which communicate asynchronously with each other over (potentially) unbounded FIFO channels by sending and receiving typed messages. As a concurrency model, CFSMs has been widely used to specify and validate communications protocols [1, 4, 11, 12, 8].

A central issue in this model is to verify whether a network of CFSMs is free of progress errors such as *deadlocks, unspecified receptions, unbounded communications* [1, 6, 9]. However, because the total number of global states that have to be explored grows exponentially as a function of size of the network, most existing verification techniques suffer from the notorious *space explosion* problem. Various other techniques have been proposed to relieve the state space explosion problem [4, 6, 5, 7]. Among them the *fair reachability* technique proposed by Gouda and Han in 1985 is particularly interesting [4]. For networks of CFSMs with bounded communications, which is the case of all practical communication protocols, the fair reachability technique proposed in [4] can reduce by more than one half the total number of global states that have to be searched to verify freedom of deadlocks. This is a significant improvement and it has been often used in combination with other techniques to further relieve the state explosion problem.

However, the fair reachability technique can only be applied to network of CFSMs with two machines. This constraint severely limits its applicability.

In this paper, we extend the classical fair reachability technique to networks of CFSMs with arbitrary number of machines. Intuitively, the extended technique, called *even reachability*, forces two actions to be taken during verification.

We start by giving relevant definitions in Section 2. Section 3 is the core of this paper. There we first introduce the concept of *even reachability* and then show that even reachability is sufficient to verify freedom of deadlocks for networks of CFSMs with any number of machines. Two example applications of this new reachability technique are given in Section 4. Section 5 concludes the paper with some remarks.

2 Definitions

The definitions given here are similar to those in [7, 8].

Let $I = \{1, 2, \cdots, n\}$, where $n \geq 2$ is some constant (denoting the number of processes in a network).

Definition 2.1 (*Communicating Finite State Machines*) *A communicating finite state machine P_i is a four-tuple $(S_i, \Sigma_i^{\pm}, \delta_i, p_{0i})$, where*

- *S_i is the set of local states of machine P_i.*
- *$p_{0i} \in S_i$ is the start state of machine P_i.*
- $$\Sigma_i^{\pm} = \sum_{1 \leq j \leq n} \Sigma_{i,j} \bigcup \sum_{1 \leq j \leq n} \Sigma_{j,i}$$

0-8186-7180-7/95 $04.00

where $\Sigma_{i,j}, 1 \leq j \leq n$ is the alphabet of messages that P_i can send to P_j, and $\Sigma_{j,i}, 1 \leq j \leq n$ is the alphabet of messages that P_i can receive from P_j.

- *$\delta_i : S_i \times \pm\Sigma_i \times I \rightarrow 2^{S_i}$ is the transition function. $\delta_i(p, -m, j)$ is the set of states that process P_i could move to from state p after sending a message m to process P_j. $\delta_i(p, +m, j)$ is the set of states that process P_i could move to from state p after receiving a message m sent by process P_j.*

We shall use $c_{i,j}$ ($i \neq j$, since it is assumed that a machine cannot send messages to or receive messages from itself) to denote the (potentially unbounded) FIFO channel from machine P_i to P_j. For each $m \in \Sigma_{i,j}$, $-(m,j)$ denotes the event of process P_i sending a message m to P_j (i.e. appending message m to $c_{i,j}$), while $+(m,i)$ denotes the event of process P_j receiving a message m sent by process P_i (i.e. removing a message from $c_{i,j}$).

A transition $p' \xrightarrow{-(m,j)} p$ ($p' \xrightarrow{+(m,j)} p$, resp.) in P_i is called a *send edge* (*receive edge*, resp.). A state p in P_i is said to be a *send* (*receive*, resp.) *state* if all of its outgoing edges are send (receive, resp.) edges. p is said to be a *mixed state* if it has both outgoing send and receive edges.

Definition 2.2 (*Networks of Communicating Finite State Machines*) *A network of communicating finite state machines is a tuple $\langle P_1, P_2, \cdots, P_n \rangle$ where each P_i is a CFSM. The semantics of a NCFSMs can be captured by the concepts of* global states *and* global state transitions. *A global state is a tuple $[\vec{v}, \vec{c}]$, where $\vec{v} = \langle p_i \rangle_{i \in I}$ and $\vec{c} = \langle c_{i,j} \rangle_{i,j \in I}$. The initial global state is $[\langle p_{0i} \rangle_{i \in I}, \langle c_{i,j} \rangle_{i,j \in I}]$, where $c_{i,j} = \varepsilon$ $(i \neq j)$. A global state $[\langle p'_k \rangle_{k \in I}, \langle c'_{k,l} \rangle_{k,l \in I}]$ is* one-step reachable *from a global state $[\langle p_k \rangle_{k \in I}, \langle c_{k,l} \rangle_{k,l \in I}]$, if $\exists i, j \in I$ such that:*

- *either $p_i \xrightarrow{-(m,j)} p'_i$ is in P_i, $\forall k \in I (k \neq i \rightarrow p_k = p'_k)$; and $c'_{i,j} = c_{i,j}.m$, $\forall k, l \in I (k \neq i \vee l \neq j \rightarrow c'_{k,l} = c_{k,l})$; or*

- *$p_i \xrightarrow{+(m,j)} p'_i$ is in P_i, $\forall k \in I (k \neq i \rightarrow p_k = p'_k)$; and $m.c'_{j,i} = c_{j,i}$, $\forall k, l \in I (k \neq j \vee l \neq i \rightarrow c'_{k,l} = c_{k,l})$.*

Although the above notion of one-step reachability appears complicated, it actually defines two simple operations. In the first case, machine P_i, in local state p_i, sends a message m to machine P_j and then moves to a local state p'_i. Hence only channel $c_{i,j}$ is changed. In the second case, machine P_i, in local state p_i, receives a message from channel $c_{j,i}$, and then moves to another local state p'_i.

A global state $[\vec{v}', \vec{c}']$ is *reachable* from another global state $[\vec{v}, \vec{c}]$, notated as

$$[\vec{v}, \vec{c}] \stackrel{*}{\Longrightarrow} [\vec{v}', \vec{c}']$$

if the former is reachable from the latter in zero or more steps. The reachability set, denoted by $RS(N)$, is defined as the least set that contains all the global states reachable from the initial global state $[\vec{v_0}, \vec{c_0}]$. We shall take the freedom of using $\vec{c_0}$ to denote an all-empty channel. For a state tuple $v = [p_1, \cdots, p_m]$, we say that v is a *receive state tuple* if each local state p_i in v is a local receive state in machine P_i.

The task of verifying a NCFSMs modeling a communication protocol is to determine if the NCFSMs possesses some undesirable properties that reflects some logic errors in the original protocol. Among the most interested and investigated undesirable properties are *Deadlocks, unspecified receptions*, and *unbounded communications* [1, 3, 5, 6, 8, 11, 12]. We are not going to give rigorous formal definitions for these properties here as informal definitions will well serve our needs. Informally, a NCFSMs is in a (total) deadlock states if each of the machines is in local receive state and the every channel is empty. An unspecified reception occurs when one or more machines in the network are in their local receive states but none of the first messages in their incoming FIFO channels matches the type of messages desired in those local receive state (recall that messages are typed). The communications of a NCFSMs are bounded if there exists some predefined constant K such that the total number of pending messages in any FIFO channel at any time is less than or equal to K.

The *topology graph* of a NCFSMs is a directed graph where the nodes are the CFSMs in the network and an edge from P_i to P_j indicates that there is a unidirectional channel from the former to the latter. A global state $[\vec{v}, \vec{c}]$ is said to be *stable* if $\vec{c} = \vec{c_0}$, i.e. all channels are empty. Apparently, a deadlock state is also a stable state, but the reverse is not true.

We shall use $\Sigma_{i,j}$ to denote the set of messages that can be transmitted over the channel $P_i \rightarrow P_j$; Σ to denote the set of messages that can be transmitted in a NCFSMs N.

3 Even reachability and deadlock detection

In this section, we introduce the notion of *even reachability* of general networks of CFSMs. We first

present the new definition and then show how it can be used for detecting deadlocks.

Definition 3.1 (*Even Reachability*) *Let $N = \langle P_1, P_2, \cdots, P_n \rangle$ be a NCFSM, and $[\vec{v}, \vec{c}]$ a reachable global state of N. A global state $[\vec{v'}, \vec{c'}]$ is one-step* even reachable *from $[\vec{v}, \vec{c}]$ if*

(1) $[\vec{v'}, \vec{c'}]$ is resulted from $[\vec{v}, \vec{c}]$ by two actions from any two different machines P_i and P_j such that the action from P_i is a send action to P_j, and the action from P_j is a receive action from P_i; or

(2) $[\vec{v'}, \vec{c'}]$ is resulted from $[\vec{v}, \vec{c}]$ by two actions from any two different machines P_i and P_j such that the action from P_i is a send action, and the action from P_j is a receive action; or

(3) $[\vec{v'}, \vec{c'}]$ is resulted from $[\vec{v}, \vec{c}]$ by two consecutive actions from the same machine.

In addition, it is also required that the above three cases are considered in the order. In other words, if $[\vec{v'}, \vec{c'}]$ is reachable from $[\vec{v}, \vec{c}]$ by the first type of derivation while $[\vec{v''}, \vec{c''}]$ is reachable from $[\vec{v}, \vec{c}]$ by the second type of derivation, then $[\vec{v''}, \vec{c''}]$ is not *considered even reachable from $[\vec{v}, \vec{c}]$. Similar restriction applies between second and third type derivations, and between the first and third type derivations.*

As usual, a global state $[\vec{v'}, \vec{c'}]$ is said to be *even reachable* from another global state $[\vec{v}, \vec{c}]$ if the former is even reachable from the latter by zero or more steps.

It is clear that the total number of messages (the sum of pending messages in all channels) in every even reachable state is even. It is also not difficult to see that for two-machine NCFSMs, even reachability is not equivalent to the classical fair reachability. In fact, even reachability is weaker than fair reachability: if a global state is fair reachable, it is also even reachable. The reverse, however, is not true.

It is obvious from the above definitions that the total number of pending messages in all channels in any even reachable state is an *even* number.

With the definition of even reachability, we can define the concept of *even reachability graph* (ERG). The ERG G of a NCFSM is a (possibly infinite) directed graph where nodes in G are even reachable states and an edge from $[\vec{v}, \vec{c}]$ to $[\vec{v'}, \vec{c'}]$ is in G if only if the latter is one-step even reachable from the former.

The following lemma states the relationship between even reachable global states and the reachability set $RS(N)$. To aid the presentation, we shall use the following notations:

- The notion s_{i_1,i_2} denotes an action of machine P_{i_1} sending a message to P_{i_2}, and r_{i_1,i_2} an action of machine P_{i_1} receiving a message sent by P_{i_2}.
- The symbol e_i will be used to denote the sequence of actions from machine P_i.

Lemma 3.1 *Let $N = \langle P_1, P_2, \cdots, P_n \rangle$ be a NCFSM. If $[\vec{v}, \vec{c}]$ is an even reachable global state, then it is a valid reachable global state in* RS(N).

Proof: We show by induction on the number of steps that any even reachable global state $[\vec{v}, \vec{c}]$ is a valid global state in $RS(N)$.

Apparently the initial global state $[\vec{v_0}, \vec{c_0}]$ is an even reachable state. Assume that $[\vec{v}, \vec{c}]$ is an even reachable global state derived by the following sequence:

$$[\vec{v_0}, \vec{c_0}] \stackrel{*}{\Longrightarrow} [\vec{v}, \vec{c}]$$

in $2k$ steps for $k > 0$. Let us consider the first pair of actions that should be taken. There are three cases:

(1) If there exist in the original derivation sequence two actions s_{i_1,i_2} and r_{i_2,i_1} which are the first actions from machine P_{i_1} and P_{i_2} respectively, we take these two actions as the first pair of actions in the even reachability sequence;

(2) If there are two send actions s_{i_1,i_2} and s_{j_1,j_2} from two different machines P_{i_1} and P_{j_1}, we take these two actions as the first pair of actions in the even reachability sequence;

(3) If there are two send actions s_{i_1,i_2} and s_{i_1,i_3} from a single machine P_{i_1}, we take these two send actions as the first pair of actions in the even reachability sequence.

Clearly, in either case, the so obtained new even reachable state is a valid reachable global state. We observe in particular that if the first two cases are not true, then Case (3) above must true. Otherwise, a deadlock or unspecified reception state would have occurred.

Assume that all even reachable global states derived within $2k$ steps, $k > 1$, are valid global states. Consider an even reachable global state derived with $2(k+1)$ steps:

$$[\vec{v_0}, \vec{c_0}] \stackrel{2k}{\Longrightarrow} [\vec{v'}, \vec{c'}] \Longrightarrow [\vec{v}, \vec{c}]$$

By induction hypothesis, $[\vec{v'}, \vec{c'}] \in RS(N)$. It is then easy to see that, for each of the three possible pairs of actions in the definition of even reachability, the global

state $[\vec{v}, \vec{c}]$ is validly reachable from $[\vec{v'}, \vec{c'}]$. Hence the conclusion follows. ■

We now present the main theorem of this paper.

Theorem 3.1 *Let $N = \langle P_1, P_2, \cdots, P_n \rangle$ be a NCFSM. If a global state $[\vec{v}, \vec{c_0}]$ is stable, then it is even reachable.*

Proof: We again show by induction, on the number of steps, that any stable reachable global state $[\vec{v}, \vec{c}]$ is also an even reachable global state.

For the base cases, apparently the initial global state $[\vec{v_0}, \vec{c_0}]$ is both stable and even reachable. If a stable global state $[\vec{v}, \vec{c}]$ is derived with two steps, the two actions involved must consist of a send action from some machine P_i to P_j and a matching receive action from P_j to P_i. It is clearly to see that such a stable state is even reachable by the definition of even reachability.

Assume that all stable reachable global states derived within $2k$ steps, $k > 1$, are even reachable. Consider a stable reachable global state $[\vec{v}, \vec{c}]$ derived with $2(k+1)$ steps:

$$[\vec{v_0}, \vec{c_0}] \stackrel{2k}{\Longrightarrow} [\vec{v'}, \vec{c'}] \Longrightarrow [\vec{v}, \vec{c}]$$

We show by contradiction that $[\vec{v}, \vec{c}]$ is even reachable. As in the proof of Lemma 3.1, we can always choose first the pair of actions from the above derivation sequence to arrive at an even reachable state $[\vec{v_1}, \vec{c_1}]$:

$$[\vec{v_0}, \vec{c_0}] \Longrightarrow [\vec{v_1}, \vec{c_1}]$$

We then continue from the global state $[\vec{v_1}, \vec{c_1}]$ to derive even reachable global states, using only actions from the original derivation sequence given above, until we arrive at a global state $[\vec{v_2}, \vec{c_2}]$:

$$[\vec{v_1}, \vec{c_1}] \stackrel{*}{\Longrightarrow} [\vec{v_2}, \vec{c_2}]$$

from which we cannot move further by even reachability.

By Lemma 3.1, $[\vec{v_2}, \vec{c_2}] \in RS(N)$. Because we cannot derive any new even reachable global state from $[\vec{v_2}, \vec{c_2}]$, all the three types of pairs of actions in the definitions of even reachability are impossible. Since in deriving $[\vec{v_2}, \vec{c_2}]$ from $[\vec{v_0}, \vec{c_0}]$ only actions from the original derivation sequence are used, $[\vec{v_2}, \vec{c_2}]$ must not be a deadlock or unspecified reception state. Hence, the only possibility left is that only one action from some machine P_i is executable. Let us consider two cases:

Case 1. This action is a send action $s_{i,j}$. Notice that in this case, P_j must have no executable send or receive action. Either there is no action following $s_{i,j}$ from P_i left, in this case we have an unspecified reception error, or the action following $s_{i,j}$ is a receive action that cannot be executed. In either case, a contradiction arises.

Case 2. This action is a receive action $r_{i,j}$. Again either there is no action following $s_{i,j}$ from P_i left, in this case we have an unspecified reception error, or the action following $s_{i,j}$ is a receive action that cannot be executed. Again, in either case, a contradiction arises.

We have shown that in all the possible situations, there always exists an even reachable step. This contradicts to the claim that we cannot move forward by executing even reachable steps. Hence such a step always exists until we exhaust all the actions from the original derivation sequence. This concludes the proof. ■

Let us call a stable global state that is even reachable as a *stable even reachable* state. We already know that every even reachable state is a valid reachable global state. Therefore, an immediate corollary of the above theorem is the following, which is the basis of applying the even reachability concept in deadlock detections.

Corollary 3.1 *A NCFSM N is free of deadlocks if and only if all the stable even reachable states in N are free from deadlocks.*

According to Theorem 3.1 and Corollary3.1, to check for deadlocks, we need only search for stable even reachable states. It is not difficult to see that it is possible for a global state with an even number of total messages not reachable through even reachability. We state this fact in the following proposition.

Proposition 3.1
For any given NCFSM with bounded communications, the total number of even reachable states is less than or equal to half of the total number of reachable global states.

4 Example applications

Based on Corollary 3.1, to verify if a given well-formed NCFSM is free from deadlocks, we need only to examine all those stable even reachable states. As indicated by Proposition 3.1, since the total number of even reachable states is less than or equal to half of the

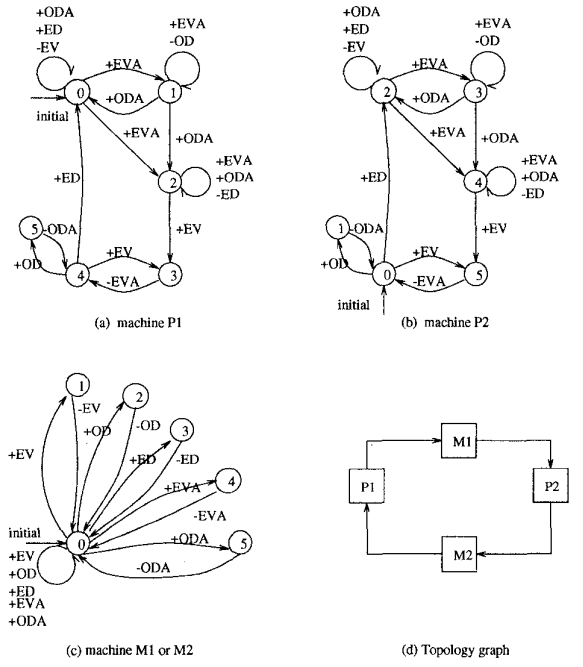

Figure 1: The alternating bit protocol

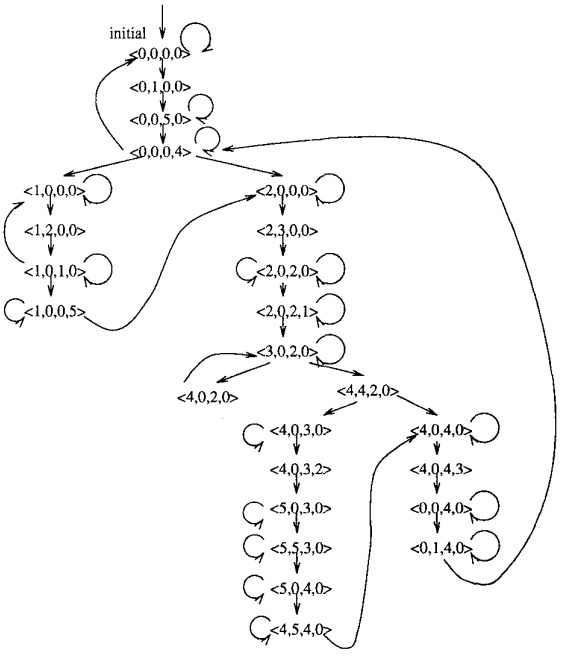

Figure 2: The simplified even reachability graph for the alternating bit protocol

total reachable global states, a significant amount of space saving is achieved. Moreover, for many practical protocols, the total number of stable even reachable states may be a very small portion of the total number of all reachable global states. The two examples in this section well demonstrate this point.

To simplify the presentation, we use the following conventions. First the pair of actions in each step in the even reachability graph are omitted. In fact, these action pairs can be easily seen by comparing the reachability graph with its original network. Second, we shall omit the components for channels in a global state if all channels in it are empty.

Example 1: Fig. 1 is a NCFSM modeling an alternating bit protocol [6]. The unreliable communication channels between two machines are modeled by two additional machines M_1 and M_2. Due to self looping send edges in machine P_1 and P_2, the communications are unbounded. However, applying the even reachability concept, there are only twenty-five even reachable states. Fig. 2 shows a simplified version of its even reachability graph. Notice that all the channels from every even state are empty. ■

Example 2: Fig. 3 is a NCFSM modeling the CSMA/CD [2] (the machine M is slightly modified here). As the machine M modeling the communication media should respond to all possible combinations of event sequences from machines A and B, this protocol can be modeled perfectly by the CFSM model. Fig. 4 shows a simplified version of its even reachability graph. Again, although the communication is unbounded, the even reachability graph for the network is bounded and there are only nineteen even reachable states. We also notice that all the channels in every even states are empty. ■

5 Concluding remarks

The notion of *even reachability* introduced in this paper extends the classical fair reachability concept, which is only applicable to two-machine NCFSMs, to general NCFSMs with any number of machines. We showed that this new notion can relieve the state explosion problem that frequently occurs in verifying communication protocols. Specifically, CFSMs with bounded communications, which is the case for most practical communication protocols, this technique can reduce by more than one half the number of reachable global states that have to be searched in verifying freedom from deadlocks.

References

[1] D. Brand and P. Zafiropulo. On communicat-

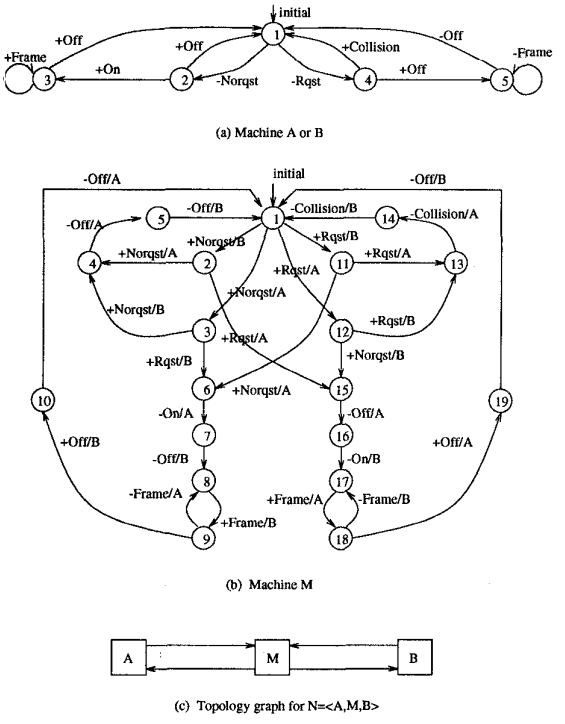

Figure 3: The CSMA/CD protocol

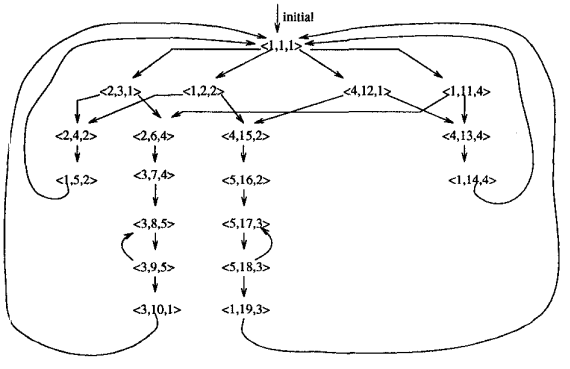

Figure 4: The simplified even reachability graph for the CSMA/CD protocol

ing finite-state machines. *JACM*, **30**(2):323-342, 1983.

[2] M. Gouda and C. K. Chang. Proving liveness for networks of communicating finite state machines. *ACM Trans. on Programming Lang. and Sys.*, Vol. **8**(1), pp.154-182, 1986.

[3] M. Gouda, E. Gurari, T.-H. Lai and L. E. Rosier. On deadlock detection in systems of communicating finite state machines. *Computers and Artificial Intelligence*, **6**(3):209-228, 1987.

[4] M. Gouda and J. Han. Protocol validation by fair progress state exploration. *Computer Networks and ISDN System*, North-Holland, Vol. **9**, pp.353-361, 1985.

[5] G.J. Holzmann. An improved protocol reachability analysis technique. *Software Practice and Experience*, Vol.**18**(2), pp.137-161, Feb. 1988.

[6] J. Pachl. Protocol description and analysis based on a state transition model with channel expressions. *Protocol Specification, Testing, and Verification, VII*, H.Rubin and C.H.West (editors), Elsevier Science Publishers B.V. (North-Holland), pp.207-219, 1987.

[7] W. Peng and S. Purushothaman. Data flow analysis of communicating finite state machines. *ACM Trans. on Programming Lang. and Sys.*, Vol. **13**(3), pp.399-442, 1991.

[8] W. Peng and S. Purushothaman. Analysis of a class of communicating finite state machines. *Acta Informatica*, **29**, pp.499-522, 1992.

[9] W. Peng. Single-link and time communicating finite state machines. *Proc. of 1994 International Conference on Network Protocols*, Boston, Oct. 1994, pp.126-133.

[10] T. Räuchle and S. Toueg. Exposure to deadlock for communicating processes is hard to detect. *Information Processing Letters*, **21**, pp.63-68, 1985.

[11] J. Rubin and C. H. West. An Improved Protocol Validation Technique. *Computer Networks*, Vol. **6**(2), pp.65-73, Apr. 1982.

[12] Y. T. Yu and M. G. Gouda. Deadlock detection for a class of communicating finite-state machines. *IEEE Transaction on Communications*, COM-**30**(12):2514-2518, Dec. 1982.

[13] Y. T. Yu and M. G. Gouda. Unboundedness detection for a class of communicating finite state machines. *Information Processing Letters*, **17**, pp.235-240, 1983.