Hierarchy-Based Incremental Deadlock Detection for Communication Protocols

K. C. Tai

Prammod V. Koppol

Center for Communications and Signal Processing
Department of Computer Science
North Carolina State University

TR-93/5
March 1993

Hierarchy-Based Incremental Deadlock Detection for Communication Protocols

K. C. Tai and Pramod V. Koppol

Person for correspondence:
Prof. K. C. Tai
Department of Computer Science
Box 8206
North Carolina State University
Raleigh, NC 27695-8206, USA
Tel: (919) 515-7146
e-mail: kct@csc.ncsu.edu

Abstract

In this paper, we consider communication protocols modeled as sets of communicating finite state machines (CFSM) with synchronous communication. For a set M of CFSMs, reachability analysis of M is to derive a composite CFSM describing the behavior of M and verify safety properties such as freedom from deadlocks or livelocks. The conventional approach to reachability analysis of M is to compose all CFSMs in M at the same time and derive all reachable global states of M. This approach suffers from the state explosion problem.

In this paper we present an incremental strategy for reachability analysis. A set of CFSMs is organized into a hierarchy. We present an algorithm that, for a given hierarchy of a set M of CFSMs, incrementally composes and reduces subsets of CFSMs in M and finally produces a minimum CFSM describing the external behavior of M. We also show that this incremental reachability analysis guarantees the detection of global deadlocks and may also detect local deadlocks. Also, we provide an algorithm for selecting a hierarchy for a set of CFSMs.

Our incremental strategy for reachability analysis may take less time and space than the conventional approach. Also, it can be easily incorporated into an incremental development of communication protocols. Furthermore, if some components of a communication protocol are modified or replaced, the effort for re-analysis can be significantly reduced by applying this incremental approach.

Keywords: Communication protocols, communicating finite state machines, reachability analysis, incremental analysis, hierarchy selection, deadlock detection

1 Introduction

We consider communication protocols modeled as sets of communicating finite state machines with synchronous communication. The reachability analysis of a communication protocol P is to identify the behavior of P and verify safety properties, such as freedom from deadlocks or livelocks, without having P's specification. The external behavior of P can also be used to determine the satisfaction of P's specifications [CES86].

For a set M of CFSMs, the conventional approach to reachability analysis, referred to as the *all-at-once* approach, is to compose all CFSMs in M at the same time and construct the reachability graph (RG) of M, which contains all reachable global states of M. The number of states in the RG of M may be as high as the product of the numbers of states of individual CFSMs in M. This state explosion problem makes the all-at-once approach impractical for analyzing large communication protocols.

Let M = {M1,M2,...,Mn}, n>0, be a set of CFSMs, where Mi, 0<i<=n, denotes a CFSM. The CFSMs in M can be organized into a hierarchy, which defines a hierarchical structure of M1, M2, ..., and Mn. For example, ((M1,M3), M4, (M2,M5)) is a hierarchy of M1 through M5. According to this hierarchy, M1 and M3 are combined and then minimized to produce the composite CFSM M13. Similarly, M2 and M5 are combined and then minimized to produce the composite CFSM M25. Finally, M13, M4 and M25 are combined and minimized to produce a CFSM describing the external behavior of M1 through M5.

The organization of this paper is as follows. Section 2 gives basic definitions. Section 3 presents an algorithm that, for a given hierarchy of a set M of CFSMs, incrementally composes and reduces subsets of CFSMs in M and finally produces a minimum CFSM describing the external behavior of M. Section 4 shows our algorithm guarantees the detection of global deadlocks and may also detect local deadlocks. Section 5 addresses the problem of

how to select a hierarchy for a set of CFSMs. Section 6 provides empirical results. Section 7 concludes this paper with a comparison of different strategies for incremental analysis.

2 Preliminaries

In this paper, we consider communication protocols modeled as sets of finite state machines which use synchronous communication with direct-naming. In other words, send and receive are blocking and the destination (source) of a send (receive) command is a process. Thus a pair of sender and receiver defines a channel. A send (receive) in process P1 and a receive (send) in process P2 match if the destination (source) of the send (receive) is P2. A matching operation between a send and receive is referred to as a synchronization operation.

Let (channel name, message) be denoted as an event. For an event u, a send operation for u is denoted as u, a receive operation for u as $\bar{u}$, and a synchronization operation for u as $*u$. A CFSM is a 5-tuple $(\Sigma, V, \sigma, s, t)$, where Σ consists of send, receive, and synchronization operations, V is a finite set of states, σ is a nondeterministic state transition function that maps a state in $(V - t)$ and an element in Σ into a subset of V, s is the initial state, and t is a set of final states, each indicating a termination of the CFSM. In this paper, a CFSM $(\Sigma, V, \sigma, s, t)$ is represented as a directed graph (V, E), where E is the set of transitions, each labeled by a send, receive, or synchronization operation. A transition labeled by a send (receive, synchronization) operation is referred to as a send (receive, synchronization) transition. A transition of a state refers to a transition leaving the state.

Fig. 1 shows three CFSMs using direct-naming. A transition labeled by "-(i,j,m)" indicates a send operation with Mi as the sender, Mj the receiver, and m the message. Similarly, a transition labeled by "+(i,j,m)" indicates a receive operation with Mi as the sender, Mj the receiver, and m the message. Also, a transition labeled by "*(i,j,m)" indicates a synchronization between Mi and Mj with delivery of message m from Mi to Mj.

In this paper, we use the notation of CCS (Calculus of Communication Systems)[Mil89] to formally define incremental reachability analysis. A synchronization operation in a CFSM is referred to as an internal operation and is denoted as τ (Thus, a synchronization operation for event u in a CFSM is denoted either as $*u$ or as τ). Let M be a set of CFSMs. A CFSM M' is referred to as the *minimum CFSM* for M if M' is the minimal machine that is observational equivalent [Mil89] to M. In this paper, $\sim$ denotes observational equivalence between two CFSMs. A brief introduction to the theory of CCS including observational equivalence can be found in [CPS91].

Let M be a set of CFSMs. A channel in M is referred to as an internal channel in M if it involves CFSMs in M, but not CFSMs outside M. Since direct-naming is used, a channel in M involves two specific CFSMs and thus the set of internal channels of M can be determined directly from M. As an example, for two CFSMs M1 and M2 communicating with each other by using direct-naming, {(1,2),(2,1)} is the set of internal channels. Let L be a set of channels. RG(M,L) denotes the reachability graph of M with each channel in L that involves some CFSMs in M as an internal channel (L may contain channels not involving any CFSM in M). More precisely, RG(M,L) is the minimum CFSM that defines the set of sequences of (1) synchronization operations among CFSMs in M and (2) send and receive operations of CFSMs in M that involve channels not in L. The minimum CFSM that is observational equivalent to RG(M,L) is denoted by MRG(M,L).

A hierarchy of CFSMs denotes a set of CFSMs with a hierarchical structure. Formally, a hierarchy H of CFSMs M1, M2, ... and Mn, n>1, is denoted as (H1,H2,...,Hm), where m>0, each Hj, 0<j<=m, is either a hierarchy or Mi for some 0<i<=n, and each Mk, 0<k<=n, occurs only once in H. Each Hj, 0<j<=m, is said to be a component of H. H is said to be sub-hierarchy of itself. For each Hj, 0<j<=m, (1) Hj is said to be a sub-hierarchy of H, (2) a sub-hierarchy of Hj, if it exists, is also said to be sub-hierarchy of H, and (3) H is

said to be the parent-hierarchy of Hj. For examples, ((M1,M2),M3,M4), (M1,(M2,M4),M3), (((M1,M3),M4),M2), and ((M1,M4),(M2,M3)) are possible hierarchies of M1 through M4. (((M1,M3),M4),M2) has (M1,M3), ((M1,M3),M4), and (((M1,M3),M4),M2) as its subhierarchies.

The level of a hierarchy H of CFSMs with respect to itself is one. For each sub-hierarchy of H, its level with respect to H is one plus the level of its parent-hierarchy with respect to H. Thus, the levels of (M1,M3) and ((M1,M3),M4) with respect to (((M1,M3),M4),M2) are 3 and 2, respectively. The depth of a hierarchy H of CFSMs, or D(H), is defined as the maximum level of sub-hierarchies of H. A top-down traversal of a hierarchy H of CFSMs is to visit H first, then the level 2 hierarchies of H, and so on. A bottom-up traversal of a hierarchy H is to visit the level D(H) sub-hierarchies of H, then the level (D(H)-1) sub-hierarchies of H, and so on.

3 Hierarchy-Based Incremental Reachability Analysis

For a set M of CFSMs with IntChan(M) as the set of internal channels, the all-at-once approach is to construct RG(M,IntChan(M)), and then reduce it to a minimum CFSM. This approach is very time- and space- consuming. An alternative is to select a hierarchy of M and incrementally compose and reduce the CFSMs in M according to this hierarchy. In this section, we present an algorithm for this incremental approach.

Algorithm INCR_DN for Incremental Reachability Analysis

Let H be a hierarchy of a set M of CFSMs M1, M2, ..., and Mn, n>1, with synchronous communication and direct naming. Algorithm INCR_DN is as follows:

- **Step 1:** For each sub-hierarchy B of H, let IntChan(B) be defined as {channels involving two or more components of B, but not any CFSM outside B}. IntChan(M) and the IntChan sets for sub-hierarchies of H are constructed as follows:

  - Let IntChan(M) = {(i,j) | (Mi has a *send* to Mj) or (Mj has a *receive* from Mi)}, 1 <= i,j <= n.

  - Perform a bottom-up traversal of H to visit sub-hierarchies of H. For a sub-hierarchy B, let IntChan(B) = {(i,j) | Mi and Mj are in two different components of B, and (i,j) $\in$ IntChan(M)}.

- **Step 2:** For each CFSM Mi, i>0, in M, let IntChan(Mi) be the empty set. Thus, MRG(Mi,IntChan(Mi)) is Mi.

- **Step 3:** Perform a bottom-up traversal of H to visit sub-hierarchies of H. For a sub-hierarchy B, where B=(B1, B2, ..., Bm), m > 1,
  let RG(B,IntChan(B)) = RG((MRG(B1,IntChan(B1)), MRG(B2,IntChan(B2)), ..., MRG(Bm,IntChan(Bm))), IntChan(B)).
  Let MRG(B,IntChan(B)) be the minimum CFSM that is observational equivalent to RG(B,IntChan(B)).

When algorithm INCR_DN terminates, RG(H,IntChan(H)) and MRG(H,IntChan(H)) have been constructed. Since IntChan(M) denotes the set of internal channels for all of CFSMs in H, RG(H,IntChan(H)) and MRG(H,IntChan(H)) are also referred to as IRG(H,IntChan(M)) and MIRG(H,IntChan(M)), respectively, where IRG stands for *incremental reachability graph.* Below we show that MIRG(H,IntChan(M)) is the minimum CFSM describing the external behavior of M. The following lemma is derived from the restriction laws of CCS (on page 80 of [Mil89]) and is needed for the proof of theorem 2.

Lemma 1. Let M and M' be sets of CFSMs and L a set of channels.

- (a) RG(M,L) = RG(M,(L U L')), where L' is a set of channels not involving any CFSMs in M.
- (b) RG((RG(M,L), RG(M',L)), L*) = RG((M,M'), (L $\cup$ L*)), where L and L* are disjoint and no channel in L involves CFSMs in both M and M'.

Theorem 2. Let H be a hierarchy of a set M of CFSMs M1, M2, ..., and Mn, n>1, with synchronous communication and direct naming. When algorithm INCR_DN is applied to H,

- (1) IRG(H,IntChan(M)) defines the set of sequences of (i) synchronization operations involving components of H and (ii) send and receive operations that involve CFSMs not in M.
- (2) IRG(H,IntChan(M)) $\sim$ RG(M,IntChan(M)), (Note that RG(M,IntChan(M)) contains all synchronization transitions that correspond to channels in IntChan(M) which is a superset of IntChan(H).)
- (3) MIRG(H,IntChan(M)) = MRG(M,IntChan(M)). (Note that both contain send and receive operations that involve CFSMs not in M)

Proof. See Appendix.

Figures 2 through 4 illustrate our incremental approach by considering the hierarchy ((M1,M2),M3), where M1, M2, and M3 are the CFSMs shown in Fig. 1. Since M1, M2 and M3 use direct-naming, the determination of internal channels for incremental analysis is easy. We first compose M1 and M2 with (1,2) and (2,1) as internal channels. The resulting CFSM is RG((M1,M2),{(1,2),(2,1)}), which is shown in Fig. 2. Then we derive the minimum CFSM that has the same external behavior as RG((M1,M2),{(1,2),(2,1)}). This minimum CFSM, shown in Fig. 3, is MRG((M1,M2),{(1,2),(2,1)}). The next step is to

compose the CFSM in Fig. 3 and M3 with {(1,3),(2,3),(3,1),(3,2)} as the set of internal channels. The resulting CFSM of this composition is given in Fig. 4. Since the CFSM in Fig. 4 has synchronization transitions only, its minimum observational equivalent CFSM has only one state with no transitions.

In figures 5 through 11, we apply our incremental analysis algorithm to generate the CFSM representing the external behavior of the alternating bit protocol (ABP). ABP consists of three component processes sender(S), receiver(R) and medium(M). The sender accepts messages from a user entity E1 and transmits the message to the receiver, which in turn delivers the message through the medium to user entity E2.

4 Deadlock Detection in Incremental Reachability Analysis

In this section, we show that incremental reachability analysis using algorithm INCR_DN guarantees the detection of global deadlocks and may also detect local deadlocks. Let M be a set of CFSMs and L a set of channels. A state in RG(M,L) is called a global deadlock state if (i) S has no successor state, (ii) S is not a final state (i.e., at least one CFSM in M is not in a final state when M is in state S), and (iii) at least one path from the initial state of RG(M,L) to S contains only synchronization transitions. Condition (iii) is required because if every path from the initial state to S contains some send or receive transitions involving channels not in L, then whether S will be entered depends upon the interaction between M and other CFSMs which constitute the *environment*. A global deadlock state s of M is said to involve a CFSM in M if this CFSM is not in a final state when M is in state s. Assume that in Fig. 1, the label associated with the transition from state S33 to state S34 is changed from "+(2,3,e)" to "+(2,3,g)". Then the state (S14,S23,S33) in Fig. 4 becomes a global deadlock state.

Since observational equivalence is used in algorithm INCR_DN and in the proof for theorem 2, one important question is whether two observational equivalent CFSMs have the same result on deadlock. Consider RG(M,L) and MRG(M,L). If RG(M,L) contains some send or receive transitions involving channels not in L, then RG(M,L) contains a global deadlock state if and only if MRG(M,L) contains a global deadlock state. (The deadlock state in MRG(M,L) is entered from the initial state through the internal operation τ). If RG(M,L) has only synchronization transitions, then RG(M,L) has empty external behavior and MRG(M,L) contains only one state and no transitions. The only state in MRG(M,L) satisfies the definition of a global deadlock state, although RG(M,L) does not necessarily contain a global deadlock state. Thus, RG(M,L) and MRG(M,L) have the same result on deadlock only if they have non-empty external behavior. To deal with this situation, whenever MIRG(M,L) has empty external behavior, we perform deadlock detection using IRG(M,L) instead of MIRG(M,L). From theorem 2 and the above discussion, we have the following theorem.

Theorem 3. Let H be a hierarchy of a set M of CFSMs M1, M2, ..., and Mn, n>1, with synchronous communication and direct naming.

- (a) IRG(H,IntChan(M)) contains a global deadlock state if and only if RG(M,IntChan(M)) contains a global deadlock state.
- (b) If MIRG(M,IntChan(M)) has non-empty external behavior, MIRG(H,IntChan(M)) contains a global deadlock state if and only if RG(M,IntChan(M)) contains a global deadlock state.

Let M be a set of CFSMs M1, M2, ..., and Mn, n>1, and L a set of channels. A state s in RG(M,L) is said to be a local deadlock state if (1) s has at least one successor state, (2) at least one path from the initial state of RG(M,L) to s contains only synchronization transitions, and (3) there exists at least one CFSM Mi, 0<i<=n, such that (a) Mi has not

entered its final state and (b) each of s and its reachable states has no transitions involving Mi (i.e., Mi cannot make any progress after entering state S). A local deadlock state s of M is said to involve a CFSM if each of s and its reachable states has no transitions involving this CFSM. Following the definitions of global deadlock and local deadlock states, we have the following theorem.

Theorem 4. Let H be a hierarchy of a set M of CFSMs M1, M2, ..., and Mn, n>1, with synchronous communication and direct naming. Let B be a sub-hierarchy of H that is not H and let M' be the set of CFSMs in B. If MIRG(B,IntChan(M')) has non-empty external behavior and contains a global deadlock state, then RG(M,IntChan(M)) contains a local deadlock or global deadlock state involving all CFSMs in M'.

Consider the deadlock example used earlier. If the modified CFSM in Fig. 4 is combined with other CFSMs, the resulting CFSM contains a local deadlock or global deadlock state involving M1, M2, and M3 and this state is entered from the initial state via the sequence of transitions labeled "*(2,3,b)", "*(1,3,c)", and "*(1,3,f)".

5 The Hierarchy Selection Problem

One major issue in incremental reachability analysis is the selection of a hierarchy for a set of CFSMs. If an arbitrary hierarchy is used, incremental analysis may take more space than all-at-once analysis. For example, for the set of CFSMs in Fig 1, the space required for incremental reachability analysis using the hierarchy ((M1,M2),M3) is more than that required for all-at-once reachability analysis. In this section, we provide an algorithm for selecting a hierarchy for a set of CFSMs with synchronous communication and direct naming.

A Hierarchy H of a set M of CFSMs may be viewed as a tree, with each leaf corresponding to a distinct CFSM in M, and each internal node corresponding to a sub-hierarchy of H. Our incremental analysis algorithm is as follows: assuming that the tree representing the hierar-

chy H has depth d, the CFSMs corresponding to internal nodes at level d-1 are generated first and then those at level d-2 are generated and so on. The CFSM corresponding to each internal node is generated by performing the composition of the CFSMs which are the children of that node and then minimizing the composite CFSM with respect to observational equivalence. The algorithm terminates when the CFSM corresponding to the root node is generated.

Let Si represent the number of states in the CFSM (before minimization) at node i in the tree. The *cost* of a *node* is defined as follows: If i is a leaf node:

$$Cost(i) = Si$$

Otherwise:

$$Cost(i) = Max\{Si, Max\{Cost(j) \mid j \textit{ is a child of } i\}\}$$

The *cost* of a *hierarchy* H is defined as the cost of the root node of the tree representing the hierarchy H. For a given set of CFSMs, the hierarchy with the minimum cost is said to be *optimal*. Given a set of CFSMs, the *hierarchy selection problem* is to select an optimal hierarchy. The straightforward approach for selecting an optimal hierarchy is to compute the costs of all possible hierarchies and then choose the one with the minimal cost. However, the cost of an internal node Si can only be computed by performing the composition of its children and hence, the straightforward approach, or any other approach that requires computing the cost for internal nodes, is impractical. In section 5.1, we define several metrics for measuring the complexity of synchronization for a set of CFSMs. In section 5.2 we apply these metrics to develop an algorithm for selecting a hierarchy for a set of CFSMs.

5.1 Metrics for Measuring Complexity of Synchronization

Let M be a set of CFSMs with synchronous communication and direct naming. We define the following metrics for M.

Send/Receive Count (SRC): The number of send and receive transitions in M is considered as a metric for measuring the complexity of synchronization operations that may take place during the composition of the CFSMs in M. SRC for M is defined as,

$$SRC(M) = \#\ of\ send/receive\ commands\ involving\ CFSMs\ in\ M$$

Normalized Send/Receive Count (NSRC): The NSRC for M is the proportion of the total number of send and receive transitions in M that may be involved in synchronization operations. It is defined as,

$$NSRC(M) = \frac{SRC(M)}{Total\ \#\ of\ transitions\ in\ M}$$

Send/Receive Density (SRD): The SRD for M is the average contribution of each CFSM in M towards the send/receive count of M. SRD for M is defined as,

$$SRD(M) = \frac{SRC(M)}{\mid M \mid}$$

Normalized Send/Receive Density (NSRD): The NSRD set M is the average contribution of each CFSM in M towards the normalized send/receive count of M. It is defined as,

$$NSRD(M) = \frac{NSRC(M)}{\mid M \mid} = \frac{SRD(M)}{Total\ \#\ of\ transitions\ in\ M}$$

5.2 An Algorithm for Hierarchy Selection

For a set M of CFSMs, our approach to selecting a hierarchy is as follows: we select a subset of M which has the maximum NSRD and generate the composite CFSM for this subset and find its minimum CFSM. We then replace the subset with the minimum CFSM and repeat the above procedure for the new set of CFSMs. Thus, in our approach, hierarchy selection and incremental reachability analysis proceed simultaneously.

The rationale behind using the normalized send/receive density for selecting the subset is the intuition that the subset with maximum normalized send/receive density would generate

the maximum number of synchronization transitions and hence provide scope for a larger gain with respect to reduction in state space size. This intuition is consistent with the core idea behind incremental analysis which is to hide as much internal detail of subsystems as possible in the early stages of analysis.

Below, we provide a high level description of our simultaneous approach to hierarchy selection and incremental analysis. Incremental analysis was discussed in section 3, hence we omit details related to incremental analysis.

- **Step 1**: Let M' = M
- **Step 2**: Select a subset S of M' which has the maximum normalized send/receive density
- **Step 3**: Find the composite CFSM for the CFSMs in S, and minimize it with respect to observational equivalence. Let the minimum machine be M"
- **Step 4**: Let M* = (M' - S) $\cup$ M"
- **Step 5**: If M* = M", M" is the required incremental reachability graph; else, let M' = M* and repeat steps 2, 3, 4 and 5

To find a subset of M with the maximum normalized send/receive density, we represent M as a matrix C of size n $\times$ n, where n is the number of CFSMs in M. C is referred to as the *interaction matrix* of M and its contents are as follows:

$$C[i,i] = 0$$

and for i $\neq$ j,

$$C[i,j] = (\#\ of\ sends\ from\ Mi\ to\ Mj) + (\#\ of\ receives\ in\ Mj\ from\ Mi)$$

Let *Trans* be a vector such that,

$$Trans[i] = \#\ of\ transitions\ in\ Mi$$

A *sub-matrix* A' of a matrix A is defined as a matrix which is formed by deleting zero or more rows, and corresponding columns, from A. (Note that a matrix is a sub-matrix of itself) Thus, a sub-matrix of C is the interaction matrix for the corresponding subset of CFSMs in M.

Let MatSum(C) denote the sum of all elements in matrix C. For a k $\times$ k sub-matrix C' of an n $\times$ n matrix C,

$$SRD(C') = \frac{MatSum(C')}{k}$$

For the subset of CFSMs represented in a sub-matrix C', the normalized send/receive density (NSRD) is computed as follows

$$NSRD(C') = \frac{SRD(C')}{\Sigma\{Trans[i] \mid Mi\ is\ represented\ in\ C'\}}$$

For example, assume

$$C = \begin{pmatrix} 0 & y1 & z1 \\ x2 & 0 & z2 \\ x3 & y3 & 0 \end{pmatrix}$$

and vector Trans is,

$$Trans = [t1\ t2\ t3]$$

Then, for sub-matrix C' formed by deleting the third row and column from C,

$$SRD(C') = \frac{y1 + x2}{2}$$

and,

$$NSRD(C') = \frac{SRD(C')}{t1 + t2}$$

Given the interaction matrix C for a set of CFSMs, in order to select a subset of CFSMs with the maximum normalized send/receive density, we select a sub-matrix C' of C with the maximum normalized send/receive density.

Theorem 5. Given the interaction matrix C and vector $Trans$ for a set of CFSMs, the problem of selecting a sub-matrix C' with the maximum normalized send/receive density is NP-complete.

Proof. See Appendix.

Although the problem is NP-complete, we feel that our approach to selecting a hierarchy is more practical and easier than the straightforward approach or any other approach that requires computing the costs of intermediate nodes in the tree representing the hierarchy. In the following, we illustrate our hierarchy selection procedure using examples.

Consider the set of CFSMs of Fig 1. The interaction matrix for this set is,

$$C = \begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}$$

and vector Trans is,

$$Trans = [4,\ 4,\ 4]$$

For this set, the sub-matrix with the maximum normalized synchronization ratio is the matrix C itself. Thus, we consider all component CFSMs at the same time, i.e., the hierarchy with the maximum NSRD is (M1,M2,M3).

As a second example, consider the CFSMs for sender, receiver and medium in the alternating bit protocol of Fig 5, 6 and 7. The interaction matrix is (row 1 corresponds to Sender; row 2 to Medium; row 3 to Receiver),

$$C = \begin{pmatrix} 0 & 4 & 0 \\ 4 & 0 & 4 \\ 0 & 4 & 0 \end{pmatrix}$$

and vector Trans is,

$$Trans = [6,\ 8,\ 6]$$

The hierarchy selected for this set is ((Sender, Medium), Receiver).

6 Empirical Studies of Incremental Analysis and Deadlock Detection

We have compared algorithm INCR_DN and the all-at-once approach by applying them to the following six concurrent programs written in CCS:

- *Alternating Bit Protocol (ABP)*: We considered two versions of ABP. The difference between the two versions is that the sender and receiver in ABP-I have a time out mechanism while those in ABP-II do not. Both use a reliable medium, but ABP-I contains deadlocks while ABP-II does not.
- *Carrier Sense Multiple Access (CSMA) with Collision Detection* [Par86].
- *Dekker's Mutual Exclusion Algorithm (DME)* [Wal88]
- *Peterson's Mutual Exclusion Algorithm (PME)* [Wal88]

For each of these CCS programs, say M, we constructed the set of internal channels IntChan(M), and performed the following tasks:

- (a) selected a hierarchy using the strategy described in the previous section, and applied algorithm INCR_DN.
  - (a.1) determined the numbers of states in the resulting MIRG, which is the same as that of MRG. This number is referred to as **#EXT**.
  - (a.2) determined the maximum of total number of states at any time during the execution of algorithm INCR_DN. This number is referred to as **#IRG_max**.

- (b) applied the all-at-once approach to determine the number of states in RG(M,IntChan(M)). This number is referred to as **#all-at-once**.

These CCS program were analyzed using the Concurrency Workbench [CPS91] to determine the values of #EXT, #IRG_max, and #all-at-once, as well as the existence of global deadlocks. The results of this analysis are shown in the following table. The column titled **#_of_Procs** gives the number of component processes of the protocol. The number of states in each component process are provided under the title **#_of_States**.

Protocol	#_of_Proc	#_of_States	#EXT	#IRG_max	#all-at-once	Deadlock?
ABP-I	3	6,6,5	9	40	41	Yes
ABP-II	3	6,6,5	2	10	12	No
CSMA	3	7,10,7	12	31	36	No
DME	5	5,9,2,2,2	2	114	126	No
PME	5	6,6,2,2,2	2	31	31	No

For PME, the hierarchy chosen for incremental analysis is exactly the all-at-once approach and thus the value of #IRG_max is the same as that of #all-at-once. The reason is that during an intermediate stage of incremental analysis, the number of states may be larger than the value of #all-at-once. This indicates that the selection of a hierarchy for a set of CFSMs is critical. The above six programs are too small to show a significant reduction of space due to incremental analysis. We are currently using larger programs for empirical studies.

7 Conclusion

In this paper, we have developed a hierarchy-based approach to incremental detection of deadlocks in a set of CFSMs with synchronous communication and direct naming. We have presented an algorithm for the selection of a hierarchy for a set of CFSMs. Also, for a given hierarchy, we have described how to perform incremental analysis. Furthermore, we have proved that our incremental algorithm guarantees the detection of global deadlocks and may

also detect local deadlocks.

Incremental analysis of communication protocols has a number of advantages. First, it may take less time and space than all-at-once analysis. Second, incremental analysis can significantly reduce the effort for re-analysis of a large protocol due to correction or enhancement. Third, during an incremental development of a large communication protocol, incremental analysis can be incorporated into the development process. Assume that a module M of a protocol P is modified or replaced. If module M has the same MIRG as its original version, then there is no need to re-analyze P for verification of properties such as freedom from global or local deadlock. If module M has a different MIRG, then portions of P that are affected by module M can be incrementally re-analyzed in bottom-to-top order. The re-analysis of P stops at a sub-hierarchy of P if the MIRG of this sub-hierarchy is not changed due to the changes in module M.

When a deadlock state of a protocol is detected, it is necessary to identify at least one path leading to the deadlock state so that so that the cause of this deadlock can be analyzed. During incremental analysis of a set of CFSMs, since some of these CFSMs are combined and minimized, the generation of a complete path leading to a deadlock state becomes a problem. We are investigating how to solve this problem. We are also investigating the extension of our hierarchy selection algorithm and the incremental analysis algorithm to sets of CFSMs with port- and/or mailbox- naming. We plan to develop tools for hierarchy selection and incremental analysis.

Incremental analysis of a set of CFSMs with synchronous communication has been studied by some researchers. In [LSU89] algorithms for incremental composition and reduction of CFSMs with direct-naming were given. At each composition step, a pair of CFSMs were combined. No quantitative way of ranking CFSMs for incremental composition was used.

The paper presented three heuristic rules for reducing a CFSM to a smaller, observational equivalent one. However, the three rules do not necessarily produce a minimum, observational equivalent CFSM.

In [SKB90] an algorithm for incremental composition of CFSMs with direct-naming was given. Each transition of a CFSM may be associated with one input (receive command) and one or more outputs (send commands). At each composition step, a pair of CFSMs were combined into one by (1) matching pairs of transitions in these two CFSMs such that an output of one transition is the input of the other transition, and (2) keeping transitions that neither take input from nor produce output to the other CFSM. There was no discussion of whether the resulting CFSM is observational equivalent to the original set of CFSMs.

In [YY91] a prototype tool for incremental analysis of programs written in an Ada-like design language called PAL was described. For a PAL program, the tool transforms it into a set of process graphs, one for each task, and performs composition and simplification of these process graphs according to program structure. The tool can also determine the equivalence of process graphs. The axioms of composition, simplification, and equivalence used in this tool are based on ACP [BeK84], which is a theory of process algebra different from CCS and CSP [Hoa85].

Our paper is different from the above mentioned papers in several aspects. First, we show how to select a hierarchy for a set of CFSMs for incremental analysis. Second, we show how to perform incremental analysis of a given hierarchy of CFSMs using the notation of CCS. Third, we apply the theory of CCS to provide a formal proof that our incremental analysis guarantees the detection of deadlocks. Fourth, we discuss the relationship between global and local deadlocks in incremental analysis.

Acknowledgment

We would like to thank Dr. Rance Cleaveland for helpful discussion on CCS and Concurrency Workbench and for providing several CCS programs used in our empirical studies.

References

[And91] G. R. Andrews, *Concurrent Programming*, Benjamin/Cummings, 1991.

[BeK84] J. A. Bergstra, and J. W. Klop, *Process Algebra for Synchronous Communication*, Information and Control, 1984, 109-137.

[CES86] E. M. Clarke, E. A. Emerson, and A. P. Sistla, *Automatic verification of finite-state systems using temporal logic*, ACM TOPLAS, Vol. 8, No. 2, April 1986, 244-263.

[CPS91] R. Cleaveland, J. Parrow and B. Steffen, *The Concurrency Workbench: A Semantics Tool for the Verification of Concurrent Systems*, ACM Tran. Programming Languages and Systems, Vol 15, No. 1, Jan. 1993, pp. 36-72.

[Hel85] D. Helmbold, and D. Luckham, *Debugging Ada tasking programs*, IEEE Software, Vol. 2, No. 2, March 1985, 47-57.

[Hoa85] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice- Hall, 1985.

[LSU89] A. M. Lapone, K. K. Sabnani and M. U. Uyar, *An Algorithmic Procedure for Checking Safety Properties of Communication protocols*, IEEE Transactions on Communications, Sept. 1989, 940-948.

[Mil89] R. Milner, *Communication and Concurrency*, Prentice-Hall, 1989.

[Par86] J. Parrow, *Verifying a CSMA/CD Protocol with CCS*, Edinburgh University technical report ECS-LFCS-86-18.

[SKB90] B. Sarikaya, V. Koukoulidis and G. V. Bochmann, *Method of Analyzing Extended Finite-State Machine Specifications*, Computer Communications, Vol 13, No.2, March 1990, 83-92.

[Wal88] D. Walker, *Analysing mutual exclusion algorithms using CCS*, Edinburgh University technical report ECS-LFCS-88-45.

[YY91] W. J. Yeh, and M. Young, *Compositional Reachability Analysis Using Process Algebra*, Proc. ACM fourth Workshop on Software Testing, Analysis, and Verification, 1991, 49-59.

Appendix

Proof of Theorem 2. The proof for the theorem is by induction on the depth of a hierarchy. For a hierarchy with D(H) = 1, the theorem is obviously true. Assume that the theorem is true for any hierarchy with D(H) = n, n >= 1. Let H be a hierarchy of M with D(H) = n+1, and H = (H1, H2, ..., Hm), m > 1. Let Ni, 0<i<=m, be the set of CFSMs included in Hi and IntChan(Ni) be the set of internal channels in Ni according to algorithm. Then (a) M is the union of N1, N2, ...and Nm, and (b) IntChan(N1), IntChan(N2), ..., IntChan(Nm), and IntChan(H) are mutually exclusive and the union of them is IntChan(M). By definition, IRG(H,IntChan(M)) = RG(H,IntChan(H)). According to step 3 of algorithm INCR_DN,
IRG(H,IntChan(M)) = RG((MRG(H1,IntChan(H1)), MRG(H2,IntChan(H2)), ...,
MRG(Hm,IntChan(Hm))), IntChan(H)).
For 1<=i<=m, MRG(Hi,IntChan(Hi)) = MIRG(Hi,IntChan(Ni)). Therefore, MIRG(Hi,IntChan(Ni)) contains send and receive operations involving CFSMs not in Ni. Thus, IRG(H, IntChan(M)) contains synchronization operations involving components of H and send and receive operations involving CFSMs not in M. Hence, part(1) is proved.
Also, for 1<=i<=m,
MIRG(Hi,IntChan(Ni)) ~ IRG(Hi,IntChan(Ni)) ~ RG(Ni,IntChan(Ni)).
Thus,
IRG(H,IntChan(M)) ~ RG((RG(N1,IntChan(N1)), RG(N2,IntChan(N2)), ...,
RG(Nm,IntChan(Nm))), IntChan(H)).
Let IC be the union of IntChan(N1), IntChan(N2), ... and IntChan(Nm). Following lemma 1(a), we can replace each IntChan(Ni), $0 < i <= m$, in the right-hand side of the above equation with IC. Hence,
IRG(H,IntChan(M)) ~ RG((RG(N1,IC), RG(N2,IC), ..., RG(Nm,IC)), IntChan(H)).
Note that IC and IntChan(H) are disjoint and no channel in IC involves CFSMs in two or more of N1, N2, ..., and Nm. Following lemma 1(b), we have
IRG(H,IntChan(M)) ~ RG((N1, N2, ..., Nm),IC $\cup$ IntChan(H)) = RG(M,IntChan(M)).
Hence, part (2) is proved. The proof for part (3) is as follows. Since IRG(H,IntChan(M)) and RG(M,IntChan(M)) are observational equivalent, their minimum, observational equivalent CFSMs

are equal and therefore MIRG(H,IntChan(M)) = MRG(H,IntChan(M)).
Q.E.D.

Lemma 6. For an n × n interaction matrix C,

$$SRD(C') <= Cmax \times n$$

where, Cmax is the maximum element in C and C' is a sub-matrix of C.

Proof. For any k × k sub-matrix C' of C,

$$MatSum(C') <= Cmax \times k \times k$$

Therefore,

$$SRD(C') <= \frac{Cmax \times k \times k}{k} <= Cmax \times k$$

Since, k <= n,

$$SRD(C') <= Cmax \times n$$

Hence, proved.

Lemma 7. For an interaction matrix C and vector Trans,

$$NSRD(C') <= \frac{Cmax}{Tmin}$$

where Cmax is the maximum element of C, Tmin is the minimum element of Trans and C' is a sub-matrix of C.

Proof. For any k × k sub-matrix C' of C,

$$MatSum(C') <= Cmax \times k \times k$$

Let $Tsum$ be the such that,

$$Tsum = \Sigma\{Trans[i] \mid Mi \text{ is represented in } C'\}$$

Clearly,

$$Tsum >= Tmin \times k$$

Therefore,

$$NSRD(C') <= \frac{Cmax \times k \times k}{k} \times \frac{1}{Tmin \times k} = \frac{Cmax}{Tmin}$$

Hence, proved.

Proof of Theorem 5 To prove theorem 5, we first show that, given an interaction matrix C, selecting a sub-matrix C' with maximum send/receive density is NP-complete. The decision version of this problem is: *Given an interaction matrix C, is there a sub-matrix C' with SRD(C') = I?*

- The problem is in NP, because, we have the following non-deterministic algorithm:

Guesser: randomly, guess a sub-matrix C' of C

Checker: check if SRD(C') = I.

(both guessing and checking can be done in polynomial time)

- The problem is NP-Hard. The proof follows:

We use the subset-sum problem to prove that our problem is NP-hard. In the subset-sum problem, we are given a finite set S of positive integers and a target t (also a positive integer). The question is, is there a subset S' of S whose elements sum to t?

The reduction: we reduce the subset-sum problem to an instance of our problem as follows:

- Let $| S | = n$
- For the ith element $Ei \in S$, we construct a vector Vi of length n such that, Vi[i] = 0 and for k $\neq$ i, Vi[k] = Ei. We thus generate n vectors V1 through Vn
- We construct a matrix C such that, for 1 <= i,j <= n, C[i,i] = 0 and for i $\neq$ j, C[i,j] = ith element of S
- Let t' = $\frac{t \times (|S'|-1)}{|S'|}$

Clearly, the above reduction can be done in polynomial time. We now have to prove the following: The subset-sum problem gives an answer YES for set S and target t, iff, there is a sub-matrix C' of matrix C with SF(C') = t'.
part I: The subset-sum problem gives an answer YES for set S and target t, if there is a sub-matrix C' of matrix C with SRD(C') = t'.
Assume there is a sub-matrix C' of C with SRD(C') = t'. Assume that C' is of size k $\times$ k where k <= n. This sub-matrix has k rows each of which contains k-1 instances of some element and a 0. Therefore,

$$\frac{(k-1) \times \Sigma(\{i \mid i \textit{ is replicated over a row in } C'\})}{k} = t'$$

that implies,

$$\Sigma(\{i \mid i \textit{ is replicated over a row in } C'\}) = \frac{t' \times k}{k-1}$$

Let the set which is the parameter to Σ above be S'. Thus, $| S' | = k$. Therefore,

$$SubsetSum(S') = \frac{t' \times | S' |}{(| S' |) - 1} = t$$

Hence, part I is proved.
part II: There is a sub-matrix C' of matrix C with SRD(C') = t', if there is a subset S' of S whose elements sum to t.
If a subset S' exists whose elements sum to t, then there is a submatrix C' of size |S'| $\times$ |S'| in matrix C whose send/receive density is t'. This is because, we have |S'| rows in matrix C, each containing (|S|) $-$ 1 replicas of some element that belongs to S'. We can delete all rows whose element does not belong to S' (and also delete corresponding columns). Also, by definition of a sub-matrix, for every row i of C that exists in C', C[i,i] also exists in C'. Thus, every row in C' contains exactly one 0. This leaves a sub-matrix of size |S'| $\times$ |S'|, whose send/receive density is given as follows:

$$SRD(C') = \frac{(| S' | -1) \times \Sigma\{i \mid i \in S'\}}{| S' |}$$

$$= \frac{(| S' | -1) \times t}{| S' |} = t'$$

Hence, part II is proved.

Since the problem is in NP and is NP-hard, the above decision problem is NP-Complete. This result and Lemma 6 imply that for an interaction matrix C and vector Trans, the problem of selecting a sub-matrix C' with maximum send/receive density is NP-complete. The problem of finding a sub-matrix C' of C with maximum normalized send/receive density is more difficult than the problem of finding a sub-matrix with maximum send/receive density. This is because,

$$NSRD(C') = \frac{SRD(C')}{\Sigma\{Trans[i] \mid \ Mi \ is \ represented \ in C'\}}$$

The denominator in the above formula is a variable and hence it makes the problem of finding sub-matrix C' with maximum NSRD more difficult than that of finding a sub-matrix C' with maximum SRD. This observation and Lemma 7 imply that for an interaction matrix C, the problem of selecting a sub-matrix C' with maximum normalized send/receive density is NP-complete.
Q.E.D.

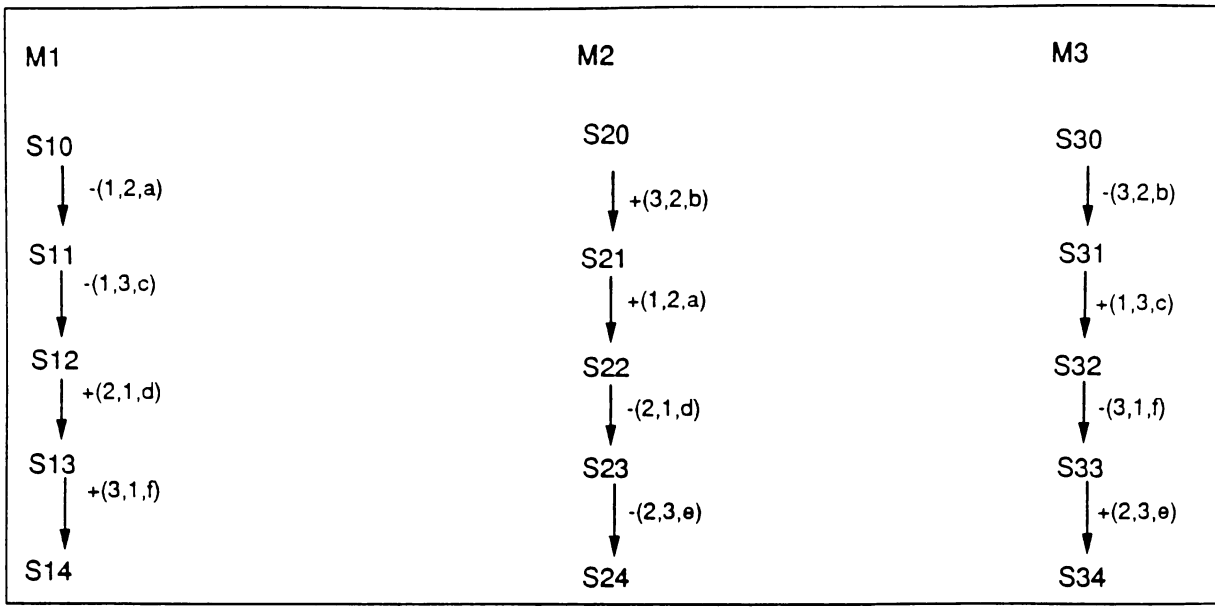

Figure 1: CFSMs M1, M2 and M3

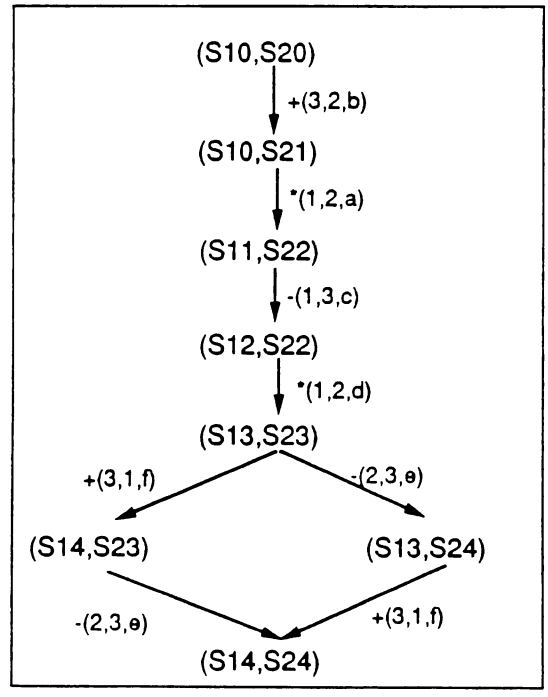

Figure 2: RG((M1,M2),{(1,2),(2,1)})

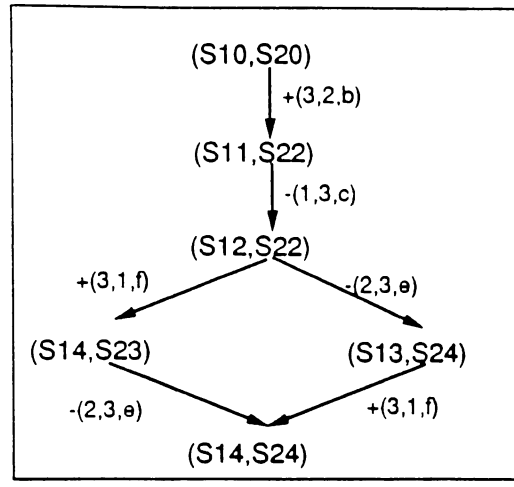

Figure 3: MRG((M1,M2),{(1,2),(2,1)})

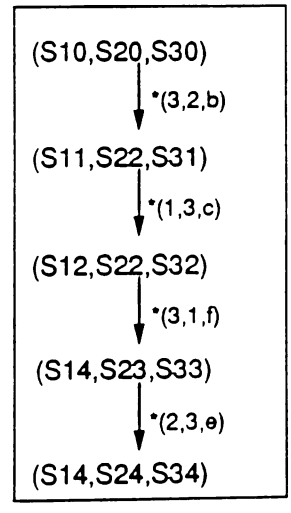

Figure 4: RG((MRG((M1,M2),{(1,2),(2,1)}),M3),{(1,3),(2,3), (3,1),(3,2)})

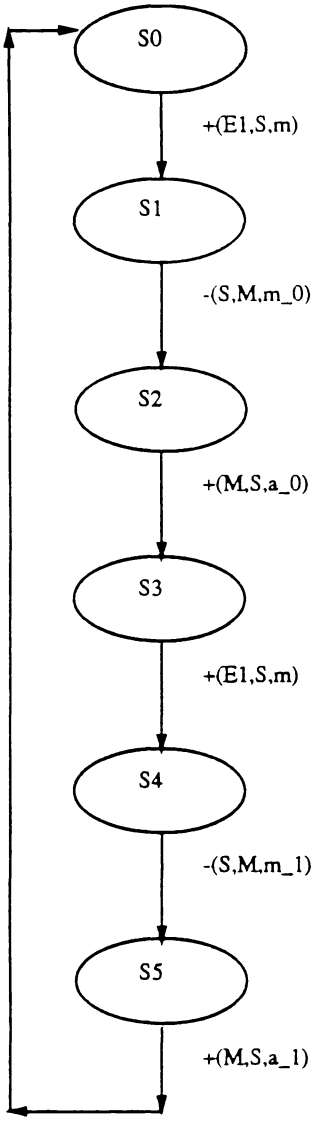

Figure 5: **Sender**

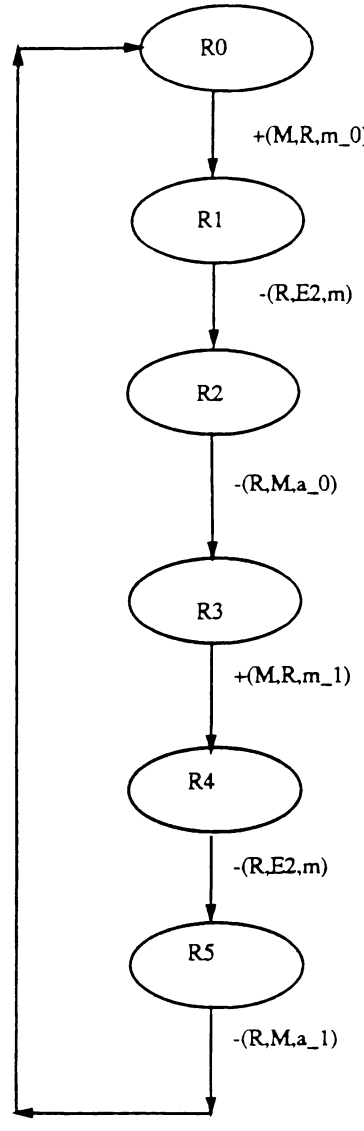

Figure 6: **Receiver**

E1: Entity 1
E2: Entity 2
M: Medium

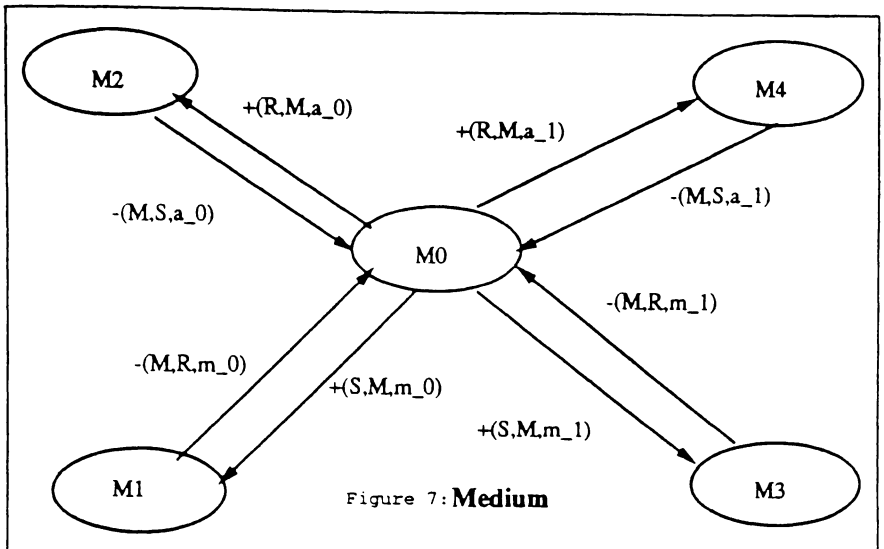

Figure 7: **Medium**

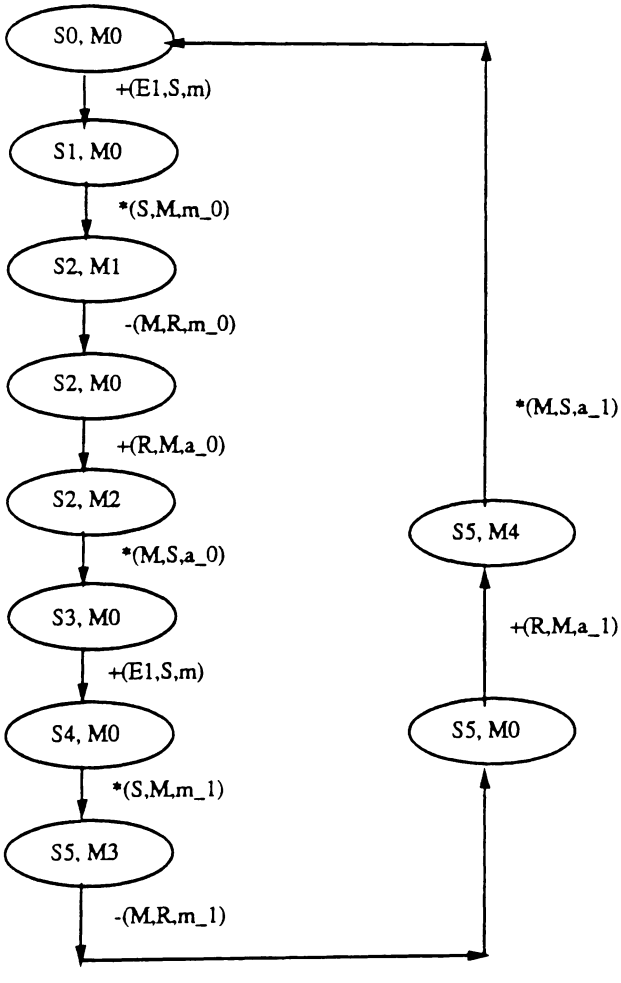

Figure 8: **RG((S,M),{(S,M),(M,S)})**

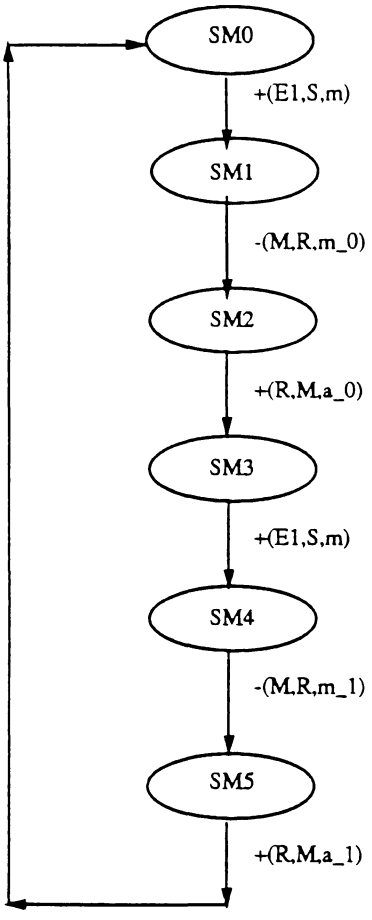

Figure 9:
MRG((S,M),{(S,M),(M,S)})

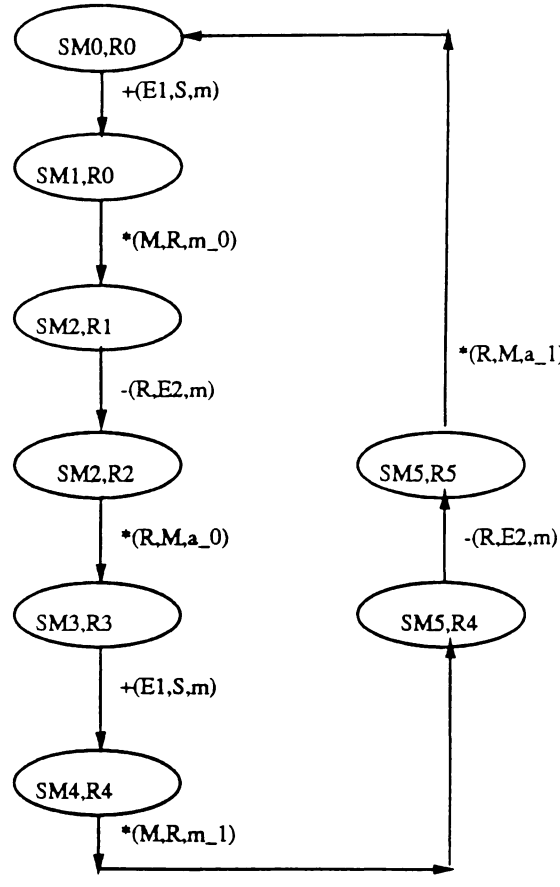

Figure 10:
RG((MRG((S,M),{(S,M),(M,S)}),R), {(S,R),(M,R),(R,S),(R,M)})

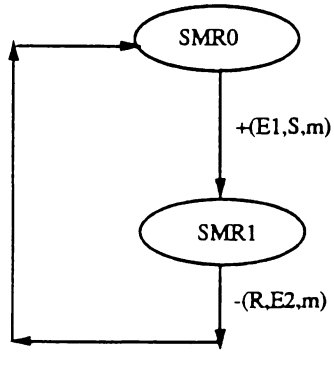

Figure 11:**MIRG((MRG((S,M),{(S,M),(M,S)}),R),{(S,R),(M,R),(R,S),(R,M)})**