29

Verification of Liveness Property for Communicating FSM's with Conditional Transitions depending on State Visiting Numbers

Teruo Higashino, Akio Nakata, Tatsuo Itoh and Kenichi Taniguchi
Dept. of Information and Computer Sciences, Osaka University,
Toyonaka, Osaka 560, Japan. Tel: +81-6-850-6607. Fax: +81-6-850-6609.
Email : {higashino,nakata,yun,taniguchi}@ics.es.osaka-u.ac.jp

Abstract

Many communication protocols are modeled as finite state machines (FSM's). In general, the size of states becomes large in order to treat parameter values such as sequence numbers, and the state explosion may occur. In this paper, we will propose an *FSM/C model* where the executability of each transition may depend on the number of times that its starting state has been visited, and propose a technique for verifying a liveness property for the model where we avoid the state explosion. For verifying the liveness property, we use an integer linear programming technique.

Keywords

protocol, communicating FSM, verification, liveness, state explosion, integer linear programming

1 INTRODUCTION

Many communication protocols are modeled as finite state machines (FSM's). FSM models are useful for the verification of communication protocols, and many verification techniques such as reachability analyses have been proposed. In FSM models, in general, the size of states becomes large in order to treat parameter values such as sequence numbers, and the state explosion may occur. In order to solve the problem, many state reduction techniques such as projection have been proposed [5, 6, and so on].

In this paper, first, we will propose an *FSM/C model* where the executability of each transition may depend on the number of times that its starting state has been visited. In this model, one counter C_{s_i} is used for each state s_i to hold the number of times that the FSM/C has visited state s_i. In our model, we may give two types of constraints for each transition : *upper bound constraints* and *lower bound constraints*. If the upper bound constraint k is given for a transition $s_i \rightarrow s_j$, then the FSM/C can execute the transition only when the value of the counter C_{s_i} is less than k (or when the remainder for C_{s_i} divided by a positive integer m is less than k), that is, only when the FSM/C has visited state s_i at most $k-1$ times (or when $mod(C_{s_i}, m) < k$ holds). If the lower bound constraint k is given for a transition $s_i \rightarrow s_j$, then the FSM/C can execute the transition only when the value of the counter C_{s_i} is greater than or equal to k (or when the remainder for C_{s_i} divided by a positive integer m is greater than or equal to k). Using this model, we can model some of communication protocols that treat such parameter values, and the number of states becomes much less than that in a general FSM model.

Next, we will propose a verification technique of a *liveness property* in the FSM/C model. If two communicating FSM/C's return to the pair of the initial states eventually regardless of transitions and their communication channels become empty, then we say that the communicating FSM/C's have the liveness property. We use an integer linear programming technique for the verification. We introduce new variables to hold the number of executions of transitions, and construct constraints consisting of integer linear inequalities using the values of the new variables and counters. The constraints denote the connecting relations between transitions and states. By deciding the satisfiability of the constraints and calculating the maximum (or minimum) values of some object functions for the constraints, we prove the liveness property of the given communicating FSM/C's. The liveness property discussed in this paper is close to the stabilization property in [3]. We treat only the pair of the initial states as the safe state and propose a technique to verify that given communicating FSM/C's return to the safe state from any unsafe state eventually. The technique proposed in this paper is close to the techniques in [1, 2]. While

G. V. Bochmann et al. (eds.), *Formal Description Techniques VIII*

the papers [1, 2] only treat FSM models, our paper treats the FSM/C model which is an EFSM model. In the traditional reachability analysis, the state explosion may occur if the values of the upper/lower bound constraints are large. However, the verification time spent in our technique does not depend on the values of the upper/lower bound constraints. Therefore, it is useful to avoid the state explosion.

The paper is structured as follows. In Section 2, we will define our FSM/C model. The constraints for a single FSM/C is described in Section 3. A verification technique to prove the liveness property for two communicating FSM/C's is given in Section 4.

2 FSM/C MODEL

[Definition 2.1 (FSM/C model)]

An FSM/C is a 5-tuple $< S, A, C, \delta, s_0 >$ where

S : a set of states $\{s_0, \ldots, s_n\}$

A : a set of actions $\{a_1, \ldots, a_m\}$
Each action is either a sending/receiving action, or a local action. If an action a_h is a sending (receiving) action, it is denoted by a_h^- (a_h^+).

C : a set of counters $\{C_{s_0}, \ldots, C_{s_n}\}$
Each counter C_{s_i} holds the number of times the FSM has visited state s_i. The initial value of each counter C_{s_i} is zero.

δ : $\subseteq S \times A \rightarrow S$: a transition relation
We describe each transition as $s_i \xrightarrow{<cond, a_h>} s_j$ ($s_i, s_j \in S$, $a_h \in A$). The executability condition (*transition condition*) *cond* of the transition is either *true*, $(C_{s_i} < k)$, $(C_{s_i} \geq k)$, $(mod(C_{s_i}, m) < k)$ or $(mod(C_{s_i}, m) \geq k)$ (where k and m are positive integers). Here, a transition $s_i \xrightarrow{<true, a_h>} s_j$ denotes that at state s_i, the action a_h is always executable and that the FSM moves to state s_j after the action a_h is executed. A transition $s_i \xrightarrow{<(C_{s_i} < k), a_h>} s_j$ denotes that at state s_i, the action a_h is executable only when the value of the counter C_{s_i} is less than k (i.e., when the FSM has visited state s_i at most $k-1$ times). Conversely, a transition $s_i \xrightarrow{<(C_{s_i} \geq k), a_h>} s_j$ denotes that at state s_i, the action a_h is executable only when the value of the counter C_{s_i} is greater than or equal to k (i.e., when the FSM has visited state s_i at least k times). Moreover, a transition $s_i \xrightarrow{<(mod(C_{s_i}, m) < k), a_h>} s_j$ denotes that at state s_i, the action a_h is executable only when the remainder of the value of the counter C_{s_i} divided by m is less than k. And, a transition $s_i \xrightarrow{<(mod(C_{s_i}, m) \geq k), a_h>} s_j$ denotes that at state s_i, the action a_h is executable only when the remainder of the value of the C_{s_i} divided by m is greater than or equal to k.

s_0 : an initial state. □

In this paper, the transition conditions $(C_{s_i} < k)$ and $(mod(C_{s_i}, m) < k)$ are referred to as the *upper bound constraints*, while $(C_{s_i} \geq k)$ and $(mod(C_{s_i}, m) \geq k)$ are referred to as the *lower bound constraints*. We call a transition with an upper/lower bound constraint as a *conditional transition*.

[Example 2.1]

M_1 and M_2 in Fig. 1 and 2 are examples of FSM/C specifications, respectively. In Fig. 1, first, M_1 sends a connection request a^-, then receives the confirmation b^+, and moves into state s_2. Next, it executes a sending action c^- 8 times. After that, it sends d^- and then moves into state s_4. At state s_4, either after receiving e^+, or receiving f^+ 8 times followed by g^+ once from the other node, it moves its state into s_2. After repeating the above sending and receiving sequence 4 times, it sends the disconnection request h^-, receives the confirmation k^+ and finally returns to the initial state s_0. M_2 in Fig. 2 is the specification of a FSM/C communicating with M_1. It firstly receives the connection request a^+, then sends the confirmation b^-, and moves into state t_2. From the state t_2, it executes the sending and receiving actions c, d, e, f and g, corresponding to those of M_1. After that, when it receives the disconnection request h^+, then it sends the confirmation k^- of the request and returns to its initial state t_0. □

To simplify the discussion here, we make the following four restrictions on a specification M.

(A1) For every state s_i, if there exists a conditional transition starting with s_i, then state s_i has exactly two conditional transitions, one is in the form of $(C_{s_i} < k)$ or $(mod(C_{s_i}, m) < k)$, and the other is in the form of $(C_{s_i} \geq k)$ or $(mod(C_{s_i}, m) \geq k)$.

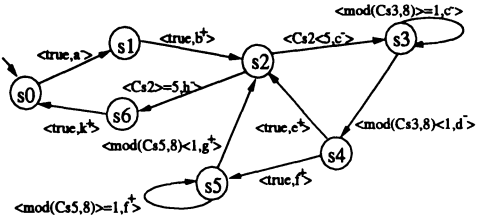

Figure 1 FSM/C specification M_1

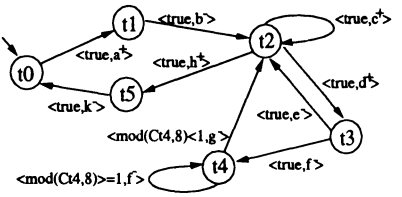

Figure 2 FSM/C specification M_2

(A2) The initial state has no self-loops and no conditional transitions.
(A3) At each state, at least one transition is executable.
(A4) Each transition is deterministic, that is, for any state s_i and any action a_h, there exist no two transitions such as $s_i \xrightarrow{<true,a_h>} s_j$ and $s_i \xrightarrow{<true,a_h>} s_k$.

Both M_1 and M_2 in Fig. 1 and 2 satisfy the above four restrictions.

3 CONSTRAINTS FOR SINGLE FSM/C

We consider a transition sequence α of a specification M. Assume that M is in state s_i. We call the state the *current state*. We introduce variables $X_{s_i s_j a_h}$ and F_{s_i}. The variable $X_{s_i s_j a_h}$ denotes the number of executions of the transition $s_i \xrightarrow{<cond,a_h>} s_j$ of M. We assume that the value of $X_{s_i s_j a_h}$ is zero if a transition $s_i \xrightarrow{<cond,a_h>} s_j$ does not exist. F_{s_i} denotes the variable whose value is one if the current state of M is s_i, or zero otherwise. In the rest of the paper, we assume that the value of every variable is a non-negative integer. The constraint expression is constructed as follows:

(I) Since the current state is unique, the following constraint holds (here, we assume that s_0 is the initial state and that the number of total states is $n+1$):

$$\sum_{i=0}^{n} F_{s_i} = 1$$

(II) Next, for each state, we consider the relation among the number of executions of the incoming/outgoing transitions and the value of the counter.

(II.1) For each state s_i except the initial state s_0, the following constraint holds (here, we assume $A = \{a_1, \ldots, a_m\}$).

$$\sum_{j=0}^{n}\sum_{h=1}^{m} X_{s_j s_i a_h} = C_{s_i} = \sum_{j=0}^{n}\sum_{h=1}^{m} X_{s_i s_j a_h} + F_{s_i} \quad (i \neq 0)$$

This expression says that the value of the counter C_{s_i} (which represents how many times M has visited state s_i) is equal to the sum of the numbers of executions of all incoming transitions. It also says that the sum of the numbers of executions of all outgoing transitions is equal to $C_{s_i} - 1$ if the current state is s_i, or C_{s_i} otherwise.

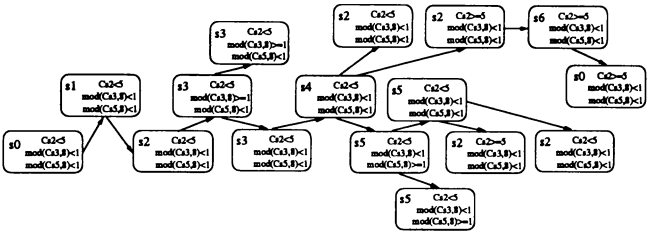

Figure 3 A reachability graph for FSM/C specification M_1

(II.2) For the initial state s_0, the following constraint holds.

$$\sum_{j=0}^{n}\sum_{h=1}^{m} X_{s_j s_0 a_h} = C_{s_0} = \sum_{j=0}^{n}\sum_{h=1}^{m} X_{s_0 s_j a_h} + F_{s_i} - 1$$

(III) We now consider the conditional transitions beginning with state s_i. Since the restriction (A1) of Section 2 holds, there exists a unique number k for the upper/lower bound constraints w.r.t. the state. Two cases are possible:

(III.1) For a given state s_i, if there exist a transition with an upper bound constraint $s_i \xrightarrow{<(C_{s_i}<k),a_p>} s_v$ and a transition with a lower bound constraint $s_i \xrightarrow{<(C_{s_i}\geq k),a_q>} s_u$, then the following two constraints hold.

$$C_{s_i} < k \Rightarrow X_{s_i s_u a_q} = 0$$
$$C_{s_i} \geq k \Rightarrow X_{s_i s_v a_p} = k - 1$$

(III.2) For a given state s_i, if there exist a transition with an upper bound constraint $s_i \xrightarrow{<(mod(C_{s_i},m)<k),a_p>} s_v$ and a transition with a lower bound constraint $s_i \xrightarrow{<(mod(C_{s_i},m)\geq k),a_q>} s_u$, then the following four constraints hold where two new variables Cd_{s_i} and Cr_{s_i} are introduced.

$$C_{s_i} = m * Cd_{s_i} + Cr_{s_i}$$
$$m - 1 \geq Cr_{s_i} \geq 0$$
$$Cr_{s_i} < k \Rightarrow X_{s_i s_u a_q} = (m - k) * Cd_{s_i}$$
$$Cr_{s_i} \geq k \Rightarrow X_{s_i s_v a_p} = k * Cd_{s_i} + (k - 1)$$

Note that for any value of C_{s_i}, the values of Cd_{s_i} and Cr_{s_i} are determined uniquely.

(IV) Using all transition conditions used in a given FSM/C specification, we construct a kind of reduced reachability graph and check which transition conditions hold for each state. For example, the FSM/C specification M_1 in Fig. 1 uses three pairs of transition conditions $< (C_{s_2} < 5), (C_{s_2} \geq 5) >$, $< (mod(C_{s_3}, 8) < 1), (mod(C_{s_3}, 8) \geq 1) >$ and $< (mod(C_{s_5}, 8) < 1), (mod(C_{s_5}, 8) \geq 1) >$. Fig. 3 is the reachability graph for M_1. In the graph, each node has a state name and a set of transition conditions. They represent which transition conditions hold at the state shown in a node. The reachability graph is constructed from the root node step by step. If a new node is constructed in a step and the node has been constructed in the previous steps, then we do not extend the node. If the node has not been constructed in the previous steps, we extend the node in the next step. Let $n + 1$ be the number of states and let l be the number of transition conditions. We can eventually construct the reachability graph because the size (number of nodes) of the graph is at most $(n + 1) * 2^l$. Note that the size of this reachability graph is much less than that of a general reachability graph.

(IV.1) From this graph, first, we omit the sets of transition conditions which are not reachable from the root node. If a set $\{cond_1, cond_2, cond_3\}$ of transition conditions is not reachable, then we give the constraint $not(cond_1 \wedge cond_2 \wedge cond_3)$.

(IV.2) Next, for each reachable set of transition conditions, we check which states satisfy the set of transition conditions. If only states s_i and s_j satisfy the set $\{cond_1, cond_2, cond_3\}$ of transition conditions, then we give the constraint $cond_1 \wedge cond_2 \wedge cond_3 \Rightarrow F_{s_i} + F_{s_j} = 1$.

(IV.3) Here, we give constraints for the initial values of the counters. In the reachability graph in Fig. 3, the set of all upper bound constraints $(C_{s_2} < 5)$, $(mod(C_{s_3}, 8) < 1)$ and $(mod(C_{s_5}, 8) < 1)$ holds for the

(*FS) $F_{s_0} + F_{s_1} + F_{s_2} + F_{s_3} + F_{s_4} + F_{s_5} + F_{s_6} = 1$
(*S0-1) $X_{s_6 s_0 k} = C_{s_0} = X_{s_0 s_1 a} + F_{s_0} - 1$
(*S1-1) $X_{s_0 s_1 a} = C_{s_1} = X_{s_1 s_2 b} + F_{s_1}$
(*S2-1) $X_{s_1 s_2 b} + X_{s_4 s_2 e} + X_{s_5 s_2 g} = C_{s_2} = X_{s_2 s_3 c} + X_{s_2 s_6 h} + F_{s_2}$
(*S2-2) $(C_{s_2} < 5) \Rightarrow X_{s_2 s_6 h} = 0$
(*S2-3) $(C_{s_2} \geq 5) \Rightarrow X_{s_2 s_3 c} = 4$
(*S3-1) $X_{s_2 s_3 c} + X_{s_3 s_3 c} = C_{s_3} = X_{s_3 s_3 c} + X_{s_3 s_4 d} + F_{s_3}$
(*S3-2) $C_{s_3} = 8 * Cd_{s_3} + Cr_{s_3}$
(*S3-3) $7 \geq Cr_{s_3} \geq 0$
(*S3-4) $(Cr_{s_3} < 1) \Rightarrow X_{s_3 s_3 c} = 7 * Cd_{s_3}$
(*S3-5) $(Cr_{s_3} \geq 1) \Rightarrow X_{s_3 s_4 d} = Cd_{s_3}$
(*S4-1) $X_{s_3 s_4 d} = C_{s_4} = X_{s_4 s_2 e} + X_{s_4 s_5 f} + F_{s_4}$
(*S5-1) $X_{s_4 s_5 f} + X_{s_5 s_5 f} = C_{s_5} = X_{s_5 s_5 f} + X_{s_5 s_2 g} + F_{s_5}$
(*S5-2) $C_{s_5} = 8 * Cd_{s_5} + Cr_{s_5}$
(*S5-3) $7 \geq Cr_{s_5} \geq 0$
(*S5-4) $(Cr_{s_5} < 1) \Rightarrow X_{s_5 s_5 f} = 7 * Cd_{s_5}$
(*S5-5) $(Cr_{s_5} \geq 1) \Rightarrow X_{s_5 s_2 g} = Cd_{s_5}$
(*S6-1) $X_{s_2 s_6 h} = C_{s_6} = X_{s_6 s_0 k} + F_{s_6}$
(*UN-1) $not((C_{s_2} < 5) \wedge (Cr_{s_3} \geq 1) \wedge (Cr_{s_5} \geq 1))$
(*UN-2) $not((C_{s_2} \geq 5) \wedge (Cr_{s_3} < 1) \wedge (Cr_{s_5} \geq 1))$
(*UN-3) $not((C_{s_2} \geq 5) \wedge (Cr_{s_3} \geq 1) \wedge (Cr_{s_5} < 1))$
(*UN-4) $not((C_{s_2} \geq 5) \wedge (Cr_{s_3} \geq 1) \wedge (Cr_{s_5} \geq 1))$
(*RE-1) $((C_{s_2} < 5) \wedge (Cr_{s_3} < 1) \wedge (Cr_{s_5} < 1)) \Rightarrow F_{s_0} + F_{s_1} + F_{s_2} + F_{s_3} + F_{s_4} + F_{s_5} = 1$
(*RE-2) $((C_{s_2} < 5) \wedge (Cr_{s_3} \geq 1) \wedge (Cr_{s_5} < 1)) \Rightarrow F_{s_3} = 1$
(*RE-3) $((C_{s_2} < 5) \wedge (Cr_{s_3} < 1) \wedge (Cr_{s_5} \geq 1)) \Rightarrow F_{s_5} = 1$
(*RE-4) $((C_{s_2} \geq 5) \wedge (Cr_{s_3} < 1) \wedge (Cr_{s_5} < 1)) \Rightarrow F_{s_0} + F_{s_2} + F_{s_6} = 1$
(*IN-1) $((C_{s_2} < 5) \wedge (Cr_{s_3} < 1) \wedge (Cr_{s_5} < 1)) \Rightarrow 5 * F_{s_0} + 5 * F_{s_1} + C_{s_2} \leq 5$

(*M1-L) $L(M_1) = X_{s_0 s_1 a} + X_{s_1 s_2 b} + X_{s_2 s_3 c} + X_{s_2 s_6 h} + X_{s_3 s_3 c} + X_{s_3 s_4 d}$
$+ X_{s_4 s_2 e} + X_{s_4 s_5 f} + X_{s_5 s_5 f} + X_{s_5 s_2 g} + X_{s_6 s_0 k}$

Figure 4 Constraint expression $CE(M_1)$ and path length $L(M_1)$

root node and some of its descendant nodes. However, from the reachability graph, we can recognize the values of the counters C_{s_2}, C_{s_3} and C_{s_5} are all zero at states s_0 and s_1 because states s_0 and s_1 are visited only when the values of these counters are zero. Therefore, we give constraints which represent that the values of the counters C_{s_2}, C_{s_3} and C_{s_5} are zero at states s_0 and s_1.

We refer to the logical conjunction of all the constraints above as a *constraint expression w.r.t. the transition relation of the specification* M, denoted by $CE(M)$. The constraint expression $CE(M)$ represents the constraints w.r.t. the numbers of execution of transitions and the numbers of visiting states, while M begins its execution with the initial state and arrives at any current state.

The length $L(M)$ of a path from the initial state to the current state can be expressed as follows:

$$L(M) = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{h=1}^{m} X_{s_i s_j a_h}$$

For M_1 in Fig. 1, the constraint expression $CE(M_1)$ and the length of the path $L(M_1)$ are given in Fig. 4. (*FS) is the constraint for the above (I), and (*S0-1),...,(*S6-1) are the constraints for the above (II) and (III). From the reachability graph in Fig. 3, we can construct the constraints for the above (IV). (*UN-1),...,(*UN-4) are the constraints for (IV.1), (*RE-1),...,(*RE-4) for (IV.2) and (*IN-1) for (IV.3).

[Proposition 3.1 (Transition sequence)]
For any transition sequence (path) α of M starting from the initial state, let σ be a value assignment corresponding to α for the variables $X_{s_i s_j a_h}$, F_{s_i}, C_{s_i}, Cd_{s_i} and Cr_{s_i}. Then, $\sigma(CE(M))$ holds. □
[Theorem 3.1]
For a given FSM/C specification M, let $Live(M)$ be a constraint expression defined as follows:

$$Live(M) \equiv CE(M) \wedge (C_{s_0} = 0 \vee (C_{s_0} = 1 \wedge F_{s_0} = 1))$$

If $Live(M)$ is satisfiable and the maximum value of $L(M)$ is finite, M eventually returns to its initial state.
(Proof)
From Proposition 3.1, any possible transition sequence which begins with the initial state s_0 and firstly returns to s_0 or never returns to s_0, satisfies $Live(M)$. Since the maximum value of $L(M)$ is finite, M

never executes transitions infinitely before firstly returning to s_0. Moreover, from the restriction (A3) of Section 2, M can always execute some transition before returning to s_0. Hence, if the maximum value of $L(M)$ is finite, M eventually returns to its initial state. □

4 VERIFICATION OF LIVENESS PROPERTY FOR COMMUNICATING FSM/C'S

In this section, we explain a verification technique of the liveness property for two communicating FSM/C's. We assume that two communicating FSM/C's are connected by two reliable unbounded communication channels which are modeled as FIFO queues with infinite capacities. Firstly, we give some channel constraints. Secondly, we explain a method to prove the liveness property.

Let's consider communicating FSM/C's consisting of M_s and M_t. We denote them as $< M_s, M_t >$. Suppose that $Chan_{st}$ denotes the communication channel from M_s to M_t, and that $Chan_{ts}$ denotes the communication channel from M_t to M_s. First, we consider the condition to return the pair of the initial states. Here, we call a state of M as a *receiving state* if M can execute only receiving actions at the state. Suppose that both the initial states of M_s and M_t are not receiving states. In this case, if M_t returns to its initial state and M_s does not, then M_t can execute a sending (or local) action at its initial state before M_s returns to its initial state. So, we cannot guarantee that M_t stays at its initial state until M_s returns to its initial state. In this case, we cannot say that M_s and M_t always return to the pair of their initial states. In order to guarantee that they always return to the pair of their initial states, either one of the initial states of M_s and M_t must be a receiving state. Therefore, in this paper, we only consider the case that either one of the initial states of M_s and M_t is a receiving state. For example, $< M_1, M_2 >$ in Fig. 1 and 2 satisfy this condition because the initial state of M_2 is a receiving state. Now, for convenience, we assume that the initial state of M_t is a receiving state for given communicating FSM/C's $< M_s, M_t >$.

4.1 Channel Constraints

Here, we give the channel constraints. A part of this technique is proposed in [1, 2]. Now, suppose that M_s has a sending action a_h^-, and that M_t has a receiving action a_h^+. For the communicating FSM/C's $< M_s, M_t >$, the sending action a_h^- is executable at any moment. However, the receiving action a_h^+ is not executable if there is no message a_h at the top of the communication channel. So, we can assume that the number of execution of the sending action a_h^- must be more than or equal to that of the receiving action a_h^+. Suppose that the numbers of states of M_s and M_t are $n_s + 1$ and $n_t + 1$, respectively. Then, for any sending/receiving action a_h, if a_h is a sending action of M_s, we assume that

$$N(a_h) = \sum_{i=0}^{n_s} \sum_{j=0}^{n_s} X_{s_i s_j a_h} - \sum_{i=0}^{n_t} \sum_{j=0}^{n_t} X_{t_i t_j a_h}$$

If it is a receiving action, then we treat $-N(a_h)$ as $N(a_h)$. $N(a_h)$ denotes the difference between the numbers of execution of the sending action a_h^- and receiving action a_h^+. Then, the following constraint holds.

$$N(a_h) \geq 0 \qquad \text{(*Num)}$$

Hereafter, we refer to the logical product of the constraints (*Num) over all of the sending/receiving actions a_h as a *channel constraint* of M_s and M_t, denoted by $CH(M_s, M_t)$. For example, let us consider FSM/C specifications M_1 and M_2 in Fig. 1 and 2. The channel constraint $CH(M_1, M_2)$ of M_1 and M_2 is shown in Fig. 5. (*Ch1-a), (*Ch1-c), (*Ch1-d) and (*Ch1-h) are constraints for the channel $Chan_{12}$. (*Ch2-b), (*Ch2-e), (*Ch2-f), (*Ch2-g) and (*Ch2-k) are constraints for the channel $Chan_{21}$.

The above constraints only constrain the *number* of times each transition has been executed. They do not constrain the *order* of the transitions. So, heuristically, we add constraints concerning with the *order* of the transitions as a part of the channel constraint $CH(M_s, M_t)$. Let $L(M)$ denote the language accepted by M where all states are considered as the final states and where we assume that M does not execute any transition if it returns to its initial state. And $L(M_s, Chan_{st})$ and $L(M_t, Chan_{st})$ denote the languages obtained from $L(M_s)$ and $L(M_t)$ by deleting all symbols not appearing as the I/O's for the channels $Chan_{st}$ and $Chan_{ts}$, respectively. For example, for $< M_1, M_2 >$ in Fig. 1 and 2,

$$\begin{aligned} L(M_1, Chan_{12}) &= \{a \cdot (c^8 \cdot d)^4 \cdot h\} \\ L(M_2, Chan_{21}) &= \{b \cdot (e + f^8 \cdot g)^* \cdot k\} \end{aligned}$$

Since $L(M_1, Chan_{12}) = \{a \cdot (c^8 \cdot d)^4 \cdot h\}$ holds, we can assume that the number of the corresponding

(*Ch1-a) $X_{s_0 s_1 a} - X_{t_0 t_1 a} \geq 0$
(*Ch1-c) $(X_{s_2 s_3 c} + X_{s_3 s_3 c}) - X_{t_2 t_2 c} \geq 0$
(*Ch1-d) $X_{s_3 s_4 d} - X_{t_2 t_3 d} \geq 0$
(*Ch1-h) $X_{s_2 s_6 h} - X_{t_2 t_5 h} \geq 0$
(*Ch2-b) $X_{t_1 t_2 b} - X_{s_1 s_2 b} \geq 0$
(*Ch2-e) $X_{t_3 t_2 e} - X_{s_4 s_2 e} \geq 0$
(*Ch2-f) $(X_{t_3 t_4 f} + X_{t_4 t_4 f}) - (X_{s_4 s_5 f} + X_{s_5 s_5 f}) \geq 0$
(*Ch2-g) $X_{t_4 t_2 g} - X_{s_5 s_2 g} \geq 0$
(*Ch2-k) $X_{t_5 t_0 k} - X_{s_6 s_0 k} \geq 0$
(*OR-cd) $X_{t_2 t_2 c} - 8 * X_{t_2 t_3 d} \geq 0$
(*OR-dh) $X_{t_2 t_3 d} - 4 * X_{t_2 t_5 h} \geq 0$
(*OR-fg) $(X_{s_4 s_5 f} + X_{s_5 s_5 f}) - 8 * X_{s_5 s_2 g} \geq 0$

Figure 5 Channel Constraint $CH(M_1, M_2)$

receiving action d^+ in M_2 must be less than or equal to the quotient for the number of c^+ divided by 8. From the language representing sending actions, the verifier can add any constraints concerning with the *order* of the transitions. If suitable constraints are added, the possibility for succeeding the verification becomes large. In Fig. 5, we add the constraints (*OR-cd) and (*OR-dh) as the constraints for the channel $Chan_{12}$, and the constraint (*OR-fg) for the channel $Chan_{21}$.

[Proposition 4.1 (Pair of transition sequences)]

For any transition sequence (path) α_s of M_s which begins with the initial state s_0 and firstly returns to s_0 or never returns to s_0, let σ_s be a value assignment corresponding to α_s to the variables $X_{s_i s_j a_h}$, F_{s_i}, C_{s_i}, Cd_{s_i} and Cr_{s_i}. And for any transition sequence α_t of M_t corresponding to α_s, let σ_t be a value assignment corresponding to α_t to the variables $X_{t_i t_j a_h}$, F_{t_i}, C_{t_i}, Cd_{t_i} and Cr_{t_i}. Then, $\sigma_t(\sigma_s(Live(M_s) \wedge CE(M_t) \wedge CH(M_s, M_t)))$ holds. □

4.2 Verification of Liveness Property

In this section, we explain a verification technique to prove the liveness property for $< M_s, M_t >$. If M_s and M_t eventually return to their initial states and the channels become empty again, we say $< M_s, M_t >$ have the liveness property. Here, for space limitation, we assume that $< M_s, M_t >$ have no deadlocks. The details of a method for proving $< M_s, M_t >$ have no deadlocks are given in [7].

In order that $< M_s, M_t >$ have the liveness property, at first, the communication channels must be empty. If a communication channel is empty, then, for any symbol a_h, the difference between the numbers of sending and receiving actions must be zero, that is, $N(a_h) = 0$ must hold. Therefore, the following relation holds.

$$N(M_s, M_t) = \sum_{h=1}^{m} N(a_h) = 0$$

[Theorem 4.1 (Liveness property)]

Suppose that given communicating FSM/C's $< M_s, M_t >$ have no deadlocks, and that the initial state of M_t is a receiving state. For the following two constraints,

$$Prog(M_s, M_t) \equiv Live(M_s) \wedge CE(M_t) \wedge CH(M_s, M_t)$$
$$Prog2(M_s, M_t) \equiv Prog(M_s, M_t) \wedge (C_{s_0} = 1) \wedge (F_{s_0} = 1)$$

if the following five conditions hold,

(1) $Prog(M_s, M_t)$ is satisfiable,
(2) The maximum values of both $L(M_s)$ and $L(M_t)$ for $Prog(M_s, M_t)$ are finite,
(3) The minimum value of F_{t_0} for $Prog2(M_s, M_t)$ is one,
(4) The maximum value of C_{t_0} for $Prog2(M_s, M_t)$ is one,
(5) The maximum value of $N(M_s, M_t)$ for $Prog2(M_s, M_t)$ is zero,

then the communicating FSM/C's $< M_s, M_t >$ have the liveness property.

(Proof)

Every path of M_s returning to the initial state s_0 has a finite length because the maximum value of $L(M_s)$ is finite. We also assume that there are no deadlocks. So, if the current state of a path is not the initial state s_0, then we can extend the path until the path reaches the initial state s_0. The extended

path also has a finite length. For any path of M_s, the corresponding path of M_t also has a finite length because the maximum value of $L(M_t)$ is finite. Since the minimum value of F_{t_0} is one and the maximum value of C_{t_0} is one for $Prog2(M_s, M_t)$, we can guarantee that M_t always returns to its initial state t_0 when M_s returns to s_0. And M_t reaches t_0 just once. Since the maximum value of $N(M_s, M_t)$ is zero, the communication channels become empty when M_s returns to s_0. Then, the theorem holds. □

For the communicating FSM/C's $< M_1, M_2 >$ in Fig. 1 and 2, the constraint $Prog(M_1, M_2)$ is satisfiable, and the maximum values of $L(M_1)$ and $L(M_2)$ are both 76. Also the minimum value of F_{t_0} is one, the maximum value of C_{t_0} is one, and the maximum value of $N(M_1, M_2)$ is zero. Therefore, the communicating FSM/C's $< M_1, M_2 >$ have the liveness property.

5 CONCLUSION

In this paper, we have proposed a technique to prove the liveness property for a restricted class of communicating FSM's with counters.

In general, the time necessary for the verification of the liveness property depends on the number l of the upper/lower bound constraints. Although we must solve integer linear programming problems for $O(2^l)$ times in the worst case, usually the number l is much less than the number $n+1$ of states. And the number of states in the FSM/C model is much less than that in a general FSM model because we can use counters in the FSM/C model. Therefore, the integer linear programming problems can be solved in reasonable times. For example, LINDO [4] can be used for solving the integer linear programming problems. The CPU time used for deciding the satisfiability for $Prog(M_1, M_2)$ and calculating the maximum value of $L(M_1)$ was about 5.0 seconds (an IBM PC/AT compatible machine GATEWAY2000 (CPU : INTEL DX4)). Our approach does not depend on the concrete values k of the upper/lower bound constraints. It can be applied even if the values k of the upper/lower bound constraints are very large. In fact, even if we use different upper/lower bound constraints ($C_{s_2} < 101$), ($mod(C_{s_3}, 128) < 1$) and ($mod(C_{s_5}, 128) < 1$) instead of ($C_{s_2} < 5$), ($mod(C_{s_3}, 8) < 1$) and ($mod(C_{s_5}, 8) < 1$) in Fig. 1 and 2, the CPU time used for the verification was almost the same. On the other hand, if the traditional reachability analysis is used, then the verification time does not depend on the number l. However, it depends on the concrete values k of the upper/lower bound constraints and the verification becomes infeasible when their values are large because the state explosion occurs. Therefore, our approach is potentially useful to avoid the state explosion.

We have a plan to develop a tool for proving the liveness property mechanically in order to show the usefulness of our approach. We believe that the verification approach described in this paper can be extended to N communicating FSM/C's. These are future works.

REFERENCES

[1] G.S. Avrunin, U.G. Buy, J.C. Corbett, L.K. Dillon and J.C. Wileden : "Automated Analysis of Concurrent Systems with Constrained Expression Toolset", IEEE Trans. on Soft. Eng., Vol. 17, No. 11, pp.1204-1222, 1991.

[2] J.C. Corbett : "Verifying General Safety and Liveness Properties With Integer Programming", Proc. Computer Aided Verification '92, pp.337-348, 1992.

[3] M.G. Gouda and N. J. Multari : "Stabilizing Communication Protocols", IEEE Trans. on Computers, Vol. 40, No. 4, pp.448-458, 1991.

[4] LINDO : "Linear interactive and discrete optimizer for linear, integer, and quadratic programming problems", LINDO Systems, Inc..

[5] S.S. Lam and A. U. Shanker : "Protocol Validation via Projection", IEEE Trans. on Soft. Eng., Vol. SE-10, No. 4, pp.325-361, 1984.

[6] J.R. Zhao and G. v. Bochmann : "Reduced Reachability Analysis of Communication Protocols : A New Approach", Proc. 6th IFIP Int. Conf. Protocol Specification, Testing and Verification (PSTV VI), pp.243-254, North-Holland, 1986.

[7] T. Higashino, A. Nakata, T. Itoh and K. Taniguchi : "Verification of Liveness Property for Communicating FSM's with Conditional Transitions depending on State Visiting Numbers", ICS Research Report, 95-ICS-6, Dept. Information and Computer Sciences, Osaka University, 1995.