論 文

확장된 선형 탐색에의한 프로토콜 검증

正會員 이 흥 규* 正會員 윤 현 수* 正會員 김 병 만* 正會員 공 재 철** 正會員 황 시 영***

A Protocol Validation by Extended Circular Exploration

Heung Kyu Lee,* Hyun Soo Yun,* Byeong Man Kim,* Jae Chul Gong,**
Si Young Whang*** *Regular Members*

ABSTRACT

In this paper, we propose an improved method of state exploration called the *extended circular exploration* (*ECE*) for the efficiency of state exploration and the reduction of explosively growing state. The ECE does not need to explore all the reachable global states, and it can be applied to N ($N \geq 2$)-party protocol with alternative routes, i.e., it is applicable to the protocol in which adaptive routing mechanism can be performed. The ECE eliminates a restriction of topology of the conventional *circular exploration* (*CE*) by exploring only those global states which are reachable, provided that the participant processes of any group of transitions proceed at the same speed, and that they can be formed as a cycle. Thus the state space explored is not exhaustive.

The algorithm presented can detect deadlock error and unspecified reception error. It requires storage space and/or execution time much less than those of the *conventional perturbation*(*CP*). It might be used for a solution of the N-process collision and the interference mechanism.

要 約

본 논문에서는 프로토콜 검증을위한 상태 탐색시에 발생하는 상태의 무한정한 증가를 방지하고 상태 탐색의 효율성을 높이기위하여 확장된 선형 탐색이라고 불리우는 상태탐색 방법을 제안하였다. 본 논문에서 제안된 확장된선형 탐색방법은 모든 도달가능한 상태를 탐색할 필요가 없으며 한 프로세스에서 다른 프로세스의 통신노선이 하나이상이 존재하고 상호통신중인 프로세스의 상태변이가 같은 속도로 진행되며 메세지의 통신형태가 하나의 사이클을 구성할수 있다면 2개 이상의 프로세스사이에서의 통신 프로토콜을 검증할수가 있다. 본 논문에서 제시된 방식에의하여 종전에 제시되었던 선형탐색 방식에의한 프로세스간 통신 형태에 대한 제약을 제거할수있다.

또한 본 논문에서 제시된 알고리즘은 되도록 에러및 규정되지않은 리셉숀 에러들을 검출할수가 있으며 종전의 perturbation방식보다 훨씬 적은 기억용량에서도 수행가능하고 또한 그 수행속도도 매우 빠르다. 따라서 본 논문에서 제시된 방식은 N개의 프로세스 사이의 충돌및 상호 교란에대한 하나의 해답을 제시하고있다.

I. Introduction

A communication protocol is defined as a set of rules to govern the cooperation between the communicating processes through message exchange which ensures an orderly delivery of information between them [1]. It can be modeled by *communicating finite state machine* (CFSM, the defi-

*韓國科學技術院
KAIST
**三星電子 情報通信部門 特殊研究所
***三星綜合技術院 情報시스템研究所
論文番號 : 92-56(接受1991. 10. 17)

nition of this CFSM is given in Section 2.) that exchange messages on many unidirectional and unbounded FIFO channels[2,3,4]. The protocol model is validated by showing that its communication satisfies some desirable properties such as freedom from deadlocks, unspecified receptions, and overflows*[5].

The most straightforward and well known technique to validate a given network of CFSM is called *state exploration*[5]**. It is based on an exhaustive exploration of all possible interactions of communicating protocol entities. Properties of the protocol can then be verified based on the global states and the global state reachability graph(or validation tree). The *reachability graph* of a given network is a directed graph whose vertices correspond to the reachable states of the network, and whose arcs denote its state transitions. In this method, deadlocks, unspecified receptions, and overflows are easily detected. The reachability graph is well suited for checking the above properties because these properties are a direct consequence of the reachability graph. However, *state explosion*—the the number of global states to be explored grows rapidly with the complexity of the protocol—makes it impossible to generate and check all reachable global states.

Many researchers have investigated the problem of state explosion, and the results were fair progress(FP)[6,7], maximal progress(MP)[8], and circular exploration(CE)[9]. The FP was the first improvement of CP but it is applicable only to 2-party protocol. The MP was the first achievement in which multiprocessor is used to handle the expanding global states through dividing the task into independent subtasks, but it is also applicable only to 2-party protocol. The CE is used for N-party protocol but it allows some restricted topologies in which there can be no alternative route between any two processes.

However, it is impractical to restrict the topology* in the real protocol. There are many protocols performed on various topologies with alternative routes in which adaptive routing mechanism can be performed in practice : note that the alternative routes between any two nodes enhance the fault tolerance of the network. Signaling System Number 7[10] is one of them. In this protocol(See Fig.1), it is inevitable to provide the alternative routes in case of link failures, using change-over and change-back procedures, for examples [11]. Thus we propose a new method which can be applied to Nparty protocol with alternative routes, and the circular exploration is extended to handle various topologies.(Definition of circular exploration is given in Section 3.)

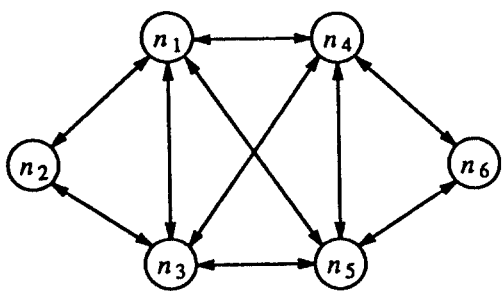

Fig. 1. Example of network topology with alternative routes[10].

Besides handling various communication topologies, the presented algorithm for protocol validation is derived to improve the efficiency of state exploration, to solve(or relieve) the problem of state explosion, and to develop a validation technique for $N(N \geq 2)$-party protocol. In this

*Definitions of these properties are given in Section 3.

**Sometime it is called *reachability analysis, or perturbation*. we will call this technique to the CP which means conventional perturbation.

*In this notation, the topology mean a possible communication structure among many processes. Thus it is not a physical structure but a logical structure among processes for communication.

paper, we present a new method of state exploration called the *extended circular exploration* (*ECE*), which does not need to explore all of the reachable global states and can be appplied to *N*-party protocol with alternative routes (see Section 3.1). With this ECE, we can eliminate a restriction of topology of the CE. The ECE explores only those global states which are reachable, provided that the particinant processes of any group of transitions proceed at the same speed, and that they can be formed as a cycle. The state space thus explored is not exhaustive.

The algorithm presented can detect deadlock errors and unspecified reception errors. It requires storage space and execution time much less than those of the CP but, a little more than those of the CE. It can be considered as a solution of the N-process collision and the interference mechanism[6]. In Section 2, the definition of N CFSMs is described. In Section 3, we describe the definition needed to extend the notion of CE, define detectable errors by this method, and present the validation algorithm and its corresponding theorem. Section 4 shows the efficiency of the presented algorithm compared to the previous method and we conclude in Section 5.

Ⅱ. N Communication Finite State Machines (CFSM)

Before formally stating the ECE, it is necessary to introduce the definitions to be used here, which are similar to the ones in[12,13]. (They are repeated here for completeness)

Definition 1. The *network topology* of a communication system is a directed labeled graph whose nodes are processes of the system, i.e., P_i denotes i-th node N_i, and arcs denote links (or channels) that allow the interaction between processes, i.e., C_{ij} denotes the channel from P_i to P_j, where $1 \le i \le N$, $1 \le j \le N$, and $i \neq j$*.

Definition 2. A communicating finite state machine (**CFSM**) can be represented by a directed labeled graph where ;

i)nodes in a CFSM are referred as states,

ii)each of the arcs is labeled t_{xy}, and is referred to as a sending arc if t is a negative integer, or is referred to as a receiving are otherwise, where the absolute value $|t|$ of t is a positive integer called a message, and x and y are distinct positive integers from 1 to N, and

iii)one of the states in a CFSM is identified as the initial state and all the states in the CFSM are reachable by directed paths from the initial state.

An are in a CFSM represents a transition in a process. A sending arc $-t_{xy}$ represents the transition in the process P_x whose execution results in the message t being inserted to the channel C_{xy} from the process P_x to the process P_y. A receiving arc t_{xy} represents the transition in the process P_x whose execution can receive the first message t in the channel C_{xy}.

The communication between processes is assumed to take place via a communication medium, which we model as a simplex channel, linking a pair of processes, so that C_{xy} represents the channel that transport information from P_x to P_y. We assume that channels have the following properties[5].

(a)The channel C_{xy} will accept messages from P_x and will deliver them to P_y in the same order as they are received, i.e., FIFO property.

(b)The time taken to transport massages between processes is variable and unspecified.

(c)Each channel C_{xy} has a predefined channel

*Throughout this paper, we consider only the N-party protocol composed of N-phocesses, $P_1,\ldots,P_N$, where $N \ge 2$.

capacity Cap_{xy}.

At any instant of time, a condition of the communication system can be completely represented by a global state defined as follows[14].

Definition 3. A global state of $F_1,\ldots,F_n$ is a pair (S, M) where

i) S is a 1-by-N matrix $[s_i]$, where s_i is a state in F_i for $1\leq i\leq N$, and

ii) M is a N-by-N matrix $[m_{ij}]$, where m_{ij} is a string of messages such that $L(m_{ij})\leq Cap_{ij}$, where $L(m_{ij})$ represents the length of the string m_{ij}.

A global state $([s_i], [m_{ij}])$ is called initial global state iff* s_i is the initial state of F_i and m_{ij} is an empty string(i.e., C_{ij} is empty), for $1\leq i\leq N$ and $1\leq j\leq N$. Informally, for $1\leq i\leq N$ and $1\leq j\leq N$, a global state $([s_i], [m_{ij}])$ implies that the execution of F_i has reached the state s_i, while the channel C_{ij} has a message sequence m_{ij}.

The rules for the execution of a transition can formally be represented as follows.

Definition 4. Let $g=([s_i], [m_{ij}])$ be a global state and e be an arc from state s_x labeled t_{xy}. Then a global state g' is said to *follow g over e*, denoted by $g\xrightarrow{e}g'$, iff the following two conditions are satisfied :

(i) if e is a sending arc from s_x to s'_x in F_x, then $L(m_{xy})<Cap_{xy}$ and $g'=([a_i], [b_{ij}])$, where $[a_i]=[s_i]$ except that $a_x=s'_x$, and $[b_{ij}]=[m_{ij}]$ except that $b_{xy}=m_{xy}\cdot|t|$, where $\cdot$ is the concatenation operator ; and

(ii) if e is a receiving arc from s_x to s'_x in F_x, then $g'=([a_i],[b_{ij}])$, where $[a_i]=[s_i]$ except that $a_x=s'_x$, and $[b_{ij}]=[m_{ij}]$ except that $m_{xy}=t\cdot b_{xy}$.

From the initial global state, any other global state can be reached only by executing transitions successively. Thus any change in the communication system can be represented as a sequence of reachable global states defined as follows[15].

Definition 5. A global state g is called *reachable* iff $g=g_0$ or there exists a sequence of global states $g_1, g_2,\cdots,g_k$ such that $g=g_k$ and, for $1\leq i\leq k$, $g_{i-1}\xrightarrow{e_i}g_i$ for some arc e_i, where g_0 is the initial global state[15].

Ⅲ. Improved Validation Algorithm 1.Extended Circular Exploration

As discussed in Section 2, the CP explores all the reachable global states. In this Section, we propose a new method of state exploration which does not explore all the reachable global states to remedy state explosion of the CP. That is to say, we explore only those global states which are reachable, provided that the participant processes of any group of transitions proceed at the same speed, and that they can be formed as a cycle. The state space thus explored is not exhaustive.

Definition 6. Let I be a finite set of pairs of positive integers. Then I is called *circular* if there exists an ordered set $\{(i_k,j_k)\,|\,(i_k,j_k)\in I\}$ with m elements such that ;

(i) $i_1=j_m$,

(ii) $i_{k+1}=j_k$ for $1\leq k\leq m-1$, and

(iii) all i_k's are distinct, and thus all j_k's are also distinct, where (i_k,j_k) is the k-th element of the ordered set and m is the number of elements in I.

For example, $\{(1,2),(2,3),(3,1)\}$ is circular.

Definition 7. A finite set C of arcs is called a *circular group* (CG) iff the set {(x,y) | there exists an arc labeled t_{xy} in C} is circular.

In this paper, the CG is classified into six types as follows ;

(i) direct CG in which two processes com-

*iff is a abbreviation of' if and only if'.

municate with each other directly :

(1)DSCG(Direct Sending Circular Group) in which the two processes send a message to each other simultaneously, i.e., two-process collision of the FP,

(2)DRCG(Direct Receiving Circular Group) in which the two processes receive a message from each other simultaneously,

(3)DMCG(Direct Mixed Circular Group) in which one process P_x sends a message to P_y and P_y receives a message from P_x or vice versa,

(ii)indirect CG in which N processes within a cycle are communication to one another indirectly ;

(4)ISCG(Indirect Sending Circular Group) in which all the N processes send a message to each other simultaneously, i.e., N-process collision as presented in the FP,

(5)IRCG(Indirect Receiving Circular Group) in which all the N processes receive a message from one another simultaneously,

(6)IMCG(Indirect Mixed Circular Group) in which some processes send messages and the others of the N processes receive messages simultaneously.

For examples, $\{-4_{23}, -7_{32}\}$ is a DSCG, $\{+7_{23}, +4_{32}\}$ is a DRCG, $\{-5_{13}, +5_{31}\}$ is a DMCG, $\{-1_{12}, -2_{23}, -3_{31}\}$ is an ISCG, $\{+3_{13}, +1_{21}, +2_{32}\}$ is an IRCG, and $\{+3_{13}, +1_{21}, -7_{32}\}$ is an IMCG, respectively.

Fig.2 illustrates the relation between CGs and topology (arcs). We can see that a) each are (or channel) generates a DMCG (Note that the *empty medium abstraction* uses only DMCGs[2]), b) each directed cycle of 2-arcs does a DSCG, a DRCG, and a DMCG, c) each directed cycle of i (≥ 3)-arcs does a DMCG, an ISCG, and an IRCG, d) each directed cycle with some bidirectional arcs does all the CGs, e) any cycles in which some arcs have different direction generate only IMCGs. This topology is similar to that of the reordering mechanism [6], and it cannot be executed by the ECE at all, i.e., any IMCG can be executed only after at least one DSCG(or ISCG) is processed since a receiving arc in the IMCG cannot be executed until a corresponding message as the first one in its channel does not exit.

Note that Hwang's topology, in total, covers only five types of CGs due to its topological restriction, i.e., the IMCG cannot exist at all with no alternative route. Therefore, the lemma 1 of Hwang's[9] cannot be applied to this topology(See Fig.2). The following is the coverage of

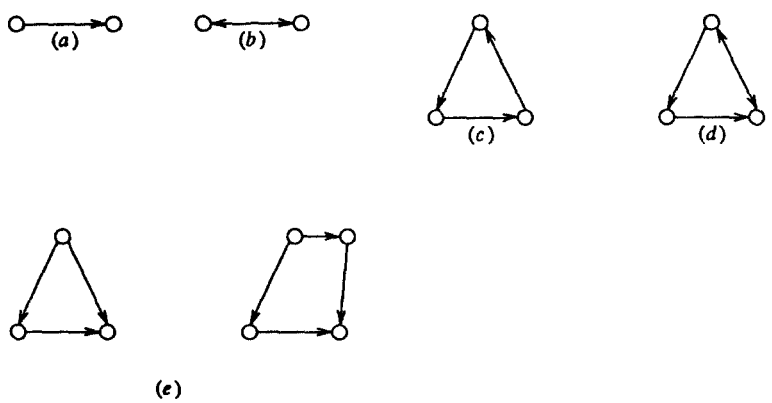

Fig. 2. The relation between topology and CGs.
(a) (b) (c) (d) (e)

topologies in each method of protocol validation in the view of these CGs :

1. the CP uses a single arc for state exploration,
2. the FP uses three types of CGs(i.e., DSCG, DRCG, and DMCG),
3. the CE uses two more types of CGs(i.e., ISCG and IRCG) to those of the FP, and finally,
4. the ECE covers one more type of CG(i.e., IMCG) to those of the CE.

The following definition formally represents that the execution of transitions in a CG can lead to a new global state.

Definition 8. Let $g=([s_i],[m_{ij}])$ and g' be two global states, and let C be a CG consisting of arcs from s_1 or s_2 or ··· or s_N. Then g' is said to *circularly follow g over C*, denoted by $g \overset{C}{\Rightarrow} g'$, iff there exists a sequence of arcs $e_1,\ldots,e_k$, where $e_i \in C$ for $1 \le i \le k$ and k is the number of elements in C, such that $g \overset{e_1}{\rightarrow} g_1 \overset{e_2}{\rightarrow} \cdots \overset{e_{k-1}}{\rightarrow} g_{k-1} \overset{e_k}{\rightarrow} g'$, for some global states $g_1,\ldots,g_k$.

Similar to the concept of *reachability* previously defined, the execution of a sequence of transition groups leads to a circularly global state defined as follows.

Definition 9. A global state g is called *circularly reachable* iff $g=g_0$ or there exists a sequence of global states $g_1,\ldots,g_k$ such that $g=g_k$ and, for $1 \le i \le k$, $g_{i-1} \overset{C_i}{\Rightarrow} g_i$ for some CG C_i, where g_0 is the initial global state.

The validation technique proposed in this paper explores all the circularly reachable global states through all the CGs rather than all the reachable global states like the CP. We call this exploration the *extended circular exploration*(*ECE*).

2. Errors Detectable

Let us define the fundamental errors detected by the proposed protocol validation algorithm.

Definition 10. A circularly reachable global state is called *no further progress* (*NFP*) if from which any circular groups can not be formed nor be executed at all.

Definition 11. For any circularly reachable global state $g=([s_i],[m_{ij}])$ with NFP, we define the following two types of errors :

(a)*Type-I error* : All the arcs from state s_i are receiving arcs for $1 \le i \le N$, and all m_{ij}'s are empty strings.

(b)*Type-II error* : There exists a nonempty string m_{xy} or a sending arc from s_x labeled t_{xy}, but for all arcs e from s_y, each of e is a receiving arc which satisfies one of the following three conditions :

(i)t' is not equal to the first message of the nonempty string m_{xy},

(ii)there does not exist a sending arc from s_y labeled t''_{xy} such that m_{xy} is an empty string and $t'=-t''$, and

(iii)m_{xy} is an empty string, r=x, and $t' \neq -t$, where t'_{yr} is the label of e.

Definition 12. For any circularly reachable global state $g=([s_i],[m_{ij}])$, we define the TYPE-III error as follows :

Type-III error : There exists a sending arc from s_x labeled t_{xy}, but $L(m_{xy}) \ge Cap_{xy}$.

3. Protocol Validation Algorithm

Algorithm 1 below presents the protocol validation procedure using the ECE. For simplicity, we define a *follower* of a global state g as being a global state which circularly follows g over C, for some CG C.

Algorithm 1. Algorithm for protocol validation by the ECE.

[step-1]Let G be a set of circularly reachable global states, initially containing only the initial global state.

[step-2]Find an element g of G whose followers have not been determined. If no such el-

ement exists, i.e., g is a state with NFP, then check whether or not g contains some types of errors (type-I,II) and record them, if any. Otherwise, terminate the algorithm as the validation is completed.

[step-3]Check whether or not g contains type-III error (record it, if any).

[step-4]Calculate the set G_g of global states consisting of all the followers of g through all the six types of CGs, if possible.

[step-5]Add all the elements of G_g which are not already in G to the set G.

[step-6]Repeat from step-2.

Algorithm 1 generates a reachability tree consisting of circularly reachable global states as nodes and circular groups as arcs. The reachability tree generated by Algorithm 1 is called the *extended group perturbation tree*(**EGPT**).

Theorem 1. Algorithm 1 always terminates.

Proof. Since the set of states and the set of arcs in a CFSM is finite, and the capacities of all channels are also finite, the set of circularly reachable states must be finite. Therefore the only case that would not terminate is the sequence involving a loop. But this sequence cannot be repeated, since each of the created circularly reachable global state is checked against all of the previously generated circularly global states. Thus the algorithm will terminate. □

We will show that Algorithm 1 can detect deadlocks and unspecified receptions by the two theorems below.

Lemma 1. A circularly reachable global state is reachable.

Proof. Let g and g_0 be a circularly reachable global state and initial global state, respectively. Then, by Definitions 7 and 9, there exists a sequence of global states $g_1, g_2, \ldots, g_k$ such that $g=g_k$ and, for $1 \le i \le k$, $g_{i-1} \xrightarrow{e_i} g_i$ for some arc e_i. Thus, by Definition 4, g is reachable. □

Theorem 2 There is a deadlock in a protocol iff Algorithm 1 detects a type-I error from the protocol.

Proof. "If part" : Since the definition of deadlock and that of type-I error are equivalent, the "if part" can be proved by showing that a reachable deadlock state is circularly reachable, i.e., Algorithm 1 detects it as a type-I error. Let $d=([s_i],[m_{ij}])$ be a reachable deadlock state. Then, by Definitions 5 and 9, the m_{ij}'s are empty strings and there exists a sequence of arcs $Q=e_1e_2\cdots e_m$ such that $g_0 \xrightarrow{e_1} g_1 \xrightarrow{e_2} \cdots \xrightarrow{e_m} g_m$ for some global states $g_1, g_2, \ldots, g_{m-1}$, where g_0 is the initial global state and $g_m=d$.

Let, for $1 \le i \le N$, $Q_i(0)$ be the sequence of arcs generated from Q by eliminating all the arcs in Q which are not in P_i. Then without loss of generality, we can assume that there are r nonempty sequences $Q_1(0), Q_2(0), \ldots, Q_r(0)$. Let, for $1 \le i \le r$, $e_i(0)$ be the first arc of $Q_i(0)$ and $(x_i(0), y_i(0))$ be the subscript pair of the label of $e_i(0)$(obviously, $x_i(0)=i$). This implies that the r processes participate in the deadlock sequence. Then, *i*) we can get a CG C_1 with k_1 arcs from r arcs in $e_i(0)$. This can be proved by Corollary 1 of APPENDIX.

Because any arcs t_{xy} in $e_i(0)$ mean that the process P_x executes an event from/to the process P_y, it can be viewed as a vertex with outdegree=1, i.e., $d^+(v)=1$. So, this situation can be viewed as a graph $G=(V, E)$ with r vertices such that $d^+(v)=1$, $v \in V$. And thus, the fact that a CG C_1 can be obtained can be proved by Corollary 1 of APPENDIX.

Now, there remain $r-k_1$ arcs in $e_i(0)$. Let $Q_i(1)$ be the sequence from $Q_i(0)$ with $e_i(0)$ in C_1 eliminated and $e_i(1)$ be the first arc of $Q_i(1)$. If there exists an empty sequence $Q_a(1)$, then the total number of arcs in $e_i(1)$ is decremented by b, where b is the number of a's in $Q_a(1)$ above. And

now, we can get another CG C_2 with k_2 arcs from $(r-b)$ arcs in $e_i(1)$. Similarly, we can get some CGs C_i with k_i arcs from $e_i(l)$ until all the $Q_i(1)$'s are empty, for $0 \le l \le n$, where n is the maximum number of CGs obtained from Q.

Finally, *ii*)the number of arcs p in Q is exactly equivalent to the total sum of k_i in C_i, i.e., $p=\sum_{l=1}^{l=n} k_l = 2m'$, where m' is the number of sending (or receiving) arcs in Q, since Q has the same number of sending and receiving arcs (Remind that Q is the sequence of arcs that leads to the deadlock state).

From any pair of arcs above, (t'_{xy}, t''_{yx}) where $-t'=t''$, we can see that it means that process P_x send and P_y receive an event or vice versa. Since an arc of each process can be viewed as an edge from x to y, (or vice versa), we can say that the indegree of any vertex v and outdegree of it are same, i.e., $d^+(v)=d^-(v)$, for $1 \le v \le r$, in this graph $G=(V, E)$ with r vertices and p edges.

Thus this situation can be viewed as Theorem B of APPENDIX. This implies that by eliminating all the CGs C_i with k_i arcs, $1 \le i \le n$, there remains no edges in G at all in the end. That is to say, there do not exist any arcs in Q which do not participate in some CGs C_i. And thus we can get some CGs C_i from Q such that $g_0 \overset{C_1}{\Rightarrow} g_1' \overset{C_2}{\Rightarrow} \cdots \overset{C_{i+1}}{\Rightarrow} g_{i+1}' \overset{C_{i+2}}{\Rightarrow} \cdots \overset{C_n}{\Rightarrow} g_n' = d$, where g_i's are circularly reachable global states. From *i*) and *ii*) we can conclude that the reachable deadlock state d is circularly reachable.

"Only if part" : Since the definition of deadlock and that of type-I error are equivalent and, by Lemma 1, every circularly reachable global state is reachable, there exists a deadlock if Algorithm 1 detects a type-I error. □

Theorem 3 There exists an unspecified reception in a protocol iff Algorithm 1 detects a type-II error from the protocol.

Proof. "If part" : Let $g=([s_i],[m_j])$ be a reachable global state with a unspecified reception. Then there exists a string m_{xy} whose first message will not be received. Let $C(m_{xy})$ of g be r and assume that there do not exist the corresponding r receiving arcs in the process P_y. Then, there exists a circularly reachable global state $g_c=([s(c)_i],[m(c)_{ij}])$ which is generated finally with m_{xy} empty in EGPT, and satisfies the followings :

(1)there exists a sequence of arcs $Q=e_1e_2 \cdots e_n$ with $g_c \xrightarrow{e_1} g_1 \cdots \xrightarrow{e_n} g_n=g$ such that no CGs can be formed from Q at all, and there remain r messages in m_{xy} of g in the end.

(2)But, by appending the corresponding r receiving arcs in P_y, we can get r CGs C_{vj} such that $g_0 \overset{C_1}{\Rightarrow} g_1 \overset{C_2}{\Rightarrow} \cdots \overset{C_c}{\Rightarrow} g_c \overset{C_{v1}}{\Rightarrow} g_{v1} \overset{C_{v2}}{\Rightarrow} \cdots \overset{C_{vj}}{\Rightarrow} g_{vj} \cdots \overset{C_{vr}}{\Rightarrow} g_{vr}=g_l$, where g_i's are circularly reachable global states, for $0 \le i \le l$ and $1 \le j \le r$.

And thus g_l is circularly reachable by Theorem 2 since all the channels are empty at the state g_l (Note that we appended r receiving arcs with a view to solving the unspecified reception state). Therefore g_c before any g_{vj}, for $1 \le j \le r$, is circularly reachable and at this g_c or g itself, type-II error must be detected by Definition 11.

This implies that Algorithm 1 detects a type-II error at the state g_c or g itself from which no CGs can be formed at all.

"Only if part" : The "only if part" is true because, by Lemma 1, every circularly reachable global state is reachable and, obviously, there is an unspecified reception in a circularly reachable global state if there is a type-II error in the circularly reachable global state. □

Not only type-I and type-II errors but also the type-III error contributes to the validation of a protocol. There is an overflow in a protocol if Algorithm 1 detects type-III error from the protocol. But, the Algorithm 1 does not always detect any type-III errors.

Ⅳ. Evaluations

Complexity of the protocol validation technique based on state exploration heavily depends on the size of the system interaction domain [5], i.e., the number of states to be explored. In this chapter, we will demonstrate that the protocol validation technique of the ECE is much more efficient than the CP in time and/or storage space by comparing the numbers of global states generated by these three techniques, and that the CE can not be applied to the generalized topology. We will compare the efficiency of the ECE with the number of global states which was generated during the validation for those three techniques. The number of global states generated by the ECE is not greater than that of the CP. In other words, the number of circularly reachable global states is not greater than the number of reachable global states. This is because, by Lemma 1, every circularly reachable global state is reachable.

1. Example

Let us present an example to show that the ECE is much more efficient than the CP and that the CE cannot be applied to the generalized topology* In this example, an dashed arrow means a transition through an IMCG unable to be executed in the CE. Fig.3 shows an example, in which there is an alternative route from P_1 to P_3, for 3-party protocol with the interference mechanism[6], To validate this protocol the ECE generates 9 global states as shown in Fig. 3(c), but the CP makes 156 states. Therefore the ECE can reduce the execution time by a factor of 17.3. We can also see that the CE cannot detect a type-I error from Fig.3(c) since it cannot reach the state C10 : Note that the CE does not define any IMCG. In the CE, the only mixed CG which is able to be generated is the DMCG, while the ECE is being progressed from the state B7 to the state C10 through an IMCG. Similarly, the ECE also generates 9 distinct global states compared with 81 states with the CP, and the storage space is reduced by a factor of 9 with the ECE.

Table 1 shows the number of states generated and results obtined by those three techniques from the example.

2. The Least Upper Bounds

The previous example shows that the CE cannot be applied to the generalized topology, while both CP and ECE support those topologies. Thus, in this Section, both CP and ECE are used in calculating the upper bounds of the number of global states generated during the validation. Let us compare two least upper bounds, B_{CP} and B_{ECE}, of the number of the global states that can be generated by the CP and ECE for $n(n \geq 2)$-party protocol based on a topology provided that every state of a process always has arcs corresponding to the topology.

Table. 1 Results of example.

	Results	# of states
CP	2 unspecified receptions, 1 deadlock	156
CE	2 type-IIs	7
ECE	2 type-IIs, 1 type-I	9

The CP generates all of the reachable global states. Therefore, $B_{CP}=N_s*N_a$, where $N_s=\prod_{i=1}^{i=n} f_i$, where f_i is the number of states in F_i* for $1 \leq i \leq n$, and N_a is the number of actions possible at every combination of states g_j for $1 \leq j \leq N_s$. N_a is given as following :

$$N_a=\sum_{i=1}^{i=n}\left(\sum_{l=1}^{l=n} K_{sil}+\sum_{l=1}^{l=n} K_{ril}\right),$$

*Note that the FP, CE, and ECE show the same results in case of 2-party protocol, and that the CE and ECE also produce same results in case of N-party protocol without any alternative routes.

*it denotes a finite state machine i.

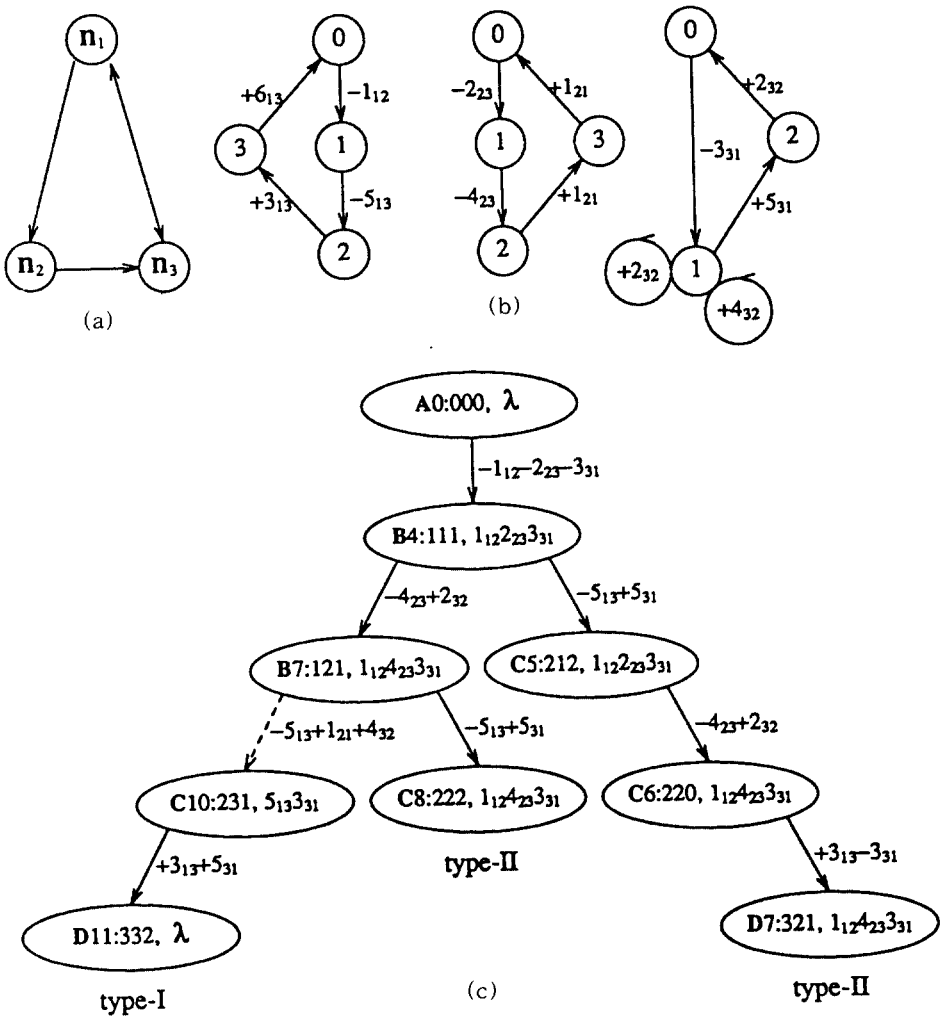

Fig. 3. An example. (a) topology. (b)CFSMs with 3-party protocol which causes the interference mechanism [6]. (c)EGPT by ECE.

where K_{sil} is the number of message types able to be sent from F_i to F_l at a combination of states g_j and K_{rli} is the number of message types able to be received from F_i by F_l at a combination of states g_j : Note that for every i, l, $K_{sil}=K_{rli}$ amd $K_{sil}=0$ if $i=l$, where $1\leq i\leq n$, $1\leq l\leq n$.

The ECE, on the other hand, explores all of the circularly reachable global states rather than all of the reachable global states. Thus, $B_{ECE}=N_{GS}*N_{CG}$, where N_{GS} is the number of circularly reachable global states and N_{CG} is the number of circular groups.

N_{CG} may be calculated from the topology since the number of each types of CGs can be calculated, for all the six types of CGs. That is to say, there are two CGs for each dicycle of i arcs ($i=2$, for DSCG and DRCG or $i\geq 3$, for ISCG and IRCG), a DMCG for every channel, and some IMCGs within a dicycle of $i(\geq 3)$ arcs. Thus, $N_{CG}=2*N_{DC}+N_{DM}+N_{IM}$, where N_{DC} is the number

of dicycles, N_{DM} is that of DMCGs, and N_{IM} is that of IMCGs, i.e., $N_{IM}=N_I-N_{IS}-N_{IR}$, where N_I is the number of indirect CGs, i.e., total sum of the number of ISCGs(N_{IS}), IRCGs(N_{IR}), and IMCGs.

Note that we can form an ISCG and an IRCG from a dicycle of $i(\geq 3)$ arcs. And we may form some IMCGs if the dicyle has at least one bidirectional arc (cf.Fig.2(d)), i.e., additional j arcs of reverse direction. So, $N_I=N_{IS}+N_{IR}+N_{IM}=\sum_{i=3}^{i=r}(1+1+\sum_{j=1}^{j=i-1} {}_iC_j)==\sum_{i=3}^{i=r}\sum_{j=0}^{j=i} {}_iC_j$, i is the number of arcs(processes) in a dicycle, r is the maximum number of i, $r\leq n$, where j is the number of arcs taken which are those of reverse direction to the rest of $i-j$ arcs within a dicycle with i arcs. For example, let the j arcs be the receiving arcs. Then an ISCG is generated when $j=0$, an IRCG when $j=i$, and the rest may be IMCGs.

And $N_{CS}=B_{CP}/N_P$ since a transition through a CG with m arcs implies that there are $m!$ paths from a circularly global state to another one by the CP, and each path generates m global states in CP(cf. Fig.4), where N_P is the number of global states from a circularly reachable global state to another one in CP before generation of CGs for all dicycles.

This N_P may be calculated from the topology of a protocol since some CGs may be generated from dicycles. Considering all of the types of CGs we get the following equation of N_P,

$$N_P=NS_D+NS_I$$

where NS_D is some parts of N_P due to the direct CGs, i.e., $NS_D=(2!*m_2*2+N_{DM})*2$ and NS_I is other parts of N_P due to the indirect CGs, i.e., $NS_I=\sum_{i=3}^{i=r} i!*m_i*\sum_{j=0}^{j=i} {}_iC_j$, where i is the number of arcs (processes) in a dicycle, i.e., the number of CGs which can be generated from the dicycle, r is the maximum number of i, m_i is the number of dicycles with i arcs (processes), and $i!$ is the number of permutations within a dicycle(Note that the DMCG case is not considered since it provides only one way when the corresponding channel is empty).

Thus we can say that $B_{ECE}\ll B_{CP}$, since $N_P\gg N_{CG}$ (Remind that $N_{CG}=2*N_{IR}(=\sum_{i=2}^{i=r} m_i)+N_{DM}+N_{IM}(=\sum_{i=3}^{i=r}$

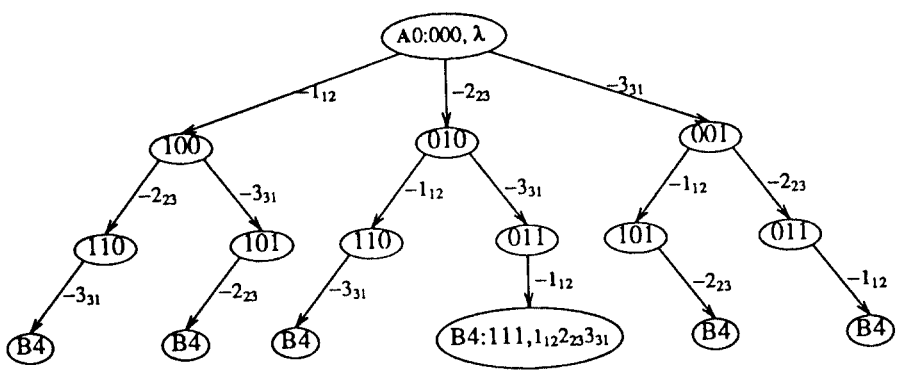

Fig. 4. Some part of the reachability graph of Fig.3(b) by the CP.*

*Note that it shows only the states generated during exploration from A0 to B4. From this example, we can see that how much redundant the CP is.

$\sum_{j=1}^{j=i-1} {}_iC_j$) and $N_P=(2!*m_2*2+N_{DM})*2+\sum_{i=3}^{i=r} i!*m_i*\sum_{j=0}^{j=i} {}_iC_j$, where $B_{ECE}=N_{GS}*N_{CG}=(B_{CP}/N_P)*N_{CG}$. As an example, for the communication network with the topology as shown in Fig. 3(a), $N_s=4*4*3=48$, $N_a(2+1)+(1+1)+(1+2)=8$, $N_{CG}=2*2+4+2=10$, and $N_P=(2!*1*2+4)*2+3!*1*4*3=88$ under the assumption of $K_{si}=1, 1\le i,j\le n$, for simplicity. Thus $B_{CP}=48*8=384$ and $B_{ECE}=(384/88)*10=43.6$.

V. Conclusions

For the purpose of reducing the number of global states generated during protocol validation by the CP, we proposed a variation of state exploration called the extended circular exploration (**ECE**) that explores only those global states which are reachable, provided that the participant processes of any group of transitions proceed at the same speed, and that they can be formed as a cycle. The state space thus explored is not exhaustive. We also presented an algorithm based on the ECE. The ECE is applicable to the $N(N\ge2)$-party protocol with alternative routes which is represented by N CFSMs and in which adaptive routing mechanism is used due to the alternative routes. Considering the applicabilities and the used CGs, the ECE may be viewed as a superset of the FP [6] and the CE[9]. The correctness of this argument is proved by the fact that the ECE can solve the two of three characteristics of $N(\ge3)$-process protocol, i.e., N-process collision and the interference mechanism [6], whereas the CE can not solve them due to the restriction of topology(Note that the CE is a superset of the FP). Furthermore it can detect deadlocks and unspecified receptions. And this algorithm can save time and/or storage in comparison with the CP. However, even with this ECE, the reordering mechanism [6] can not still be solved. Further works are needed for solving this problem. And additional further works are development of decidable algorithm for the boundedness of the communication network and for computing the smallest possible capacities for corresponding channels between any two processes.

Appendix. Two Significant Theorems

Theorem A For any digraph $G=(V, E)$ with $d^+(v)=d^-(v)$, $v\in V$, let $V'=\{v\in V|d^+(v)=d^-(v)=0\}$ be the set of isolated vertices. Then $G'=(V-V', E)$ with n vertices such that $d^+(v)=d^-(v)\ge1$, has at least one directed cycle C.

Proof. Let us prove it by contradiction. Assume that even a cycle does not exist. Since $d^+(v)\ge1$, every vertex $v_i\in V-V'$ has at least one outgoing edge and vertex $v_j\in V-V'$ which is adjacent from v_i. For any vertex v_i, an arbitrary vertex of v_j's must be always determined. Thus, the total number of v_i's and v_j's become greater than n in the end, otherwise we can generate a directed cycle C. Fig.5 illustrates this situation. At last, the number of vertices becomes infinite. This is a contradiction. Thus G' has at least one directed cycle C. □

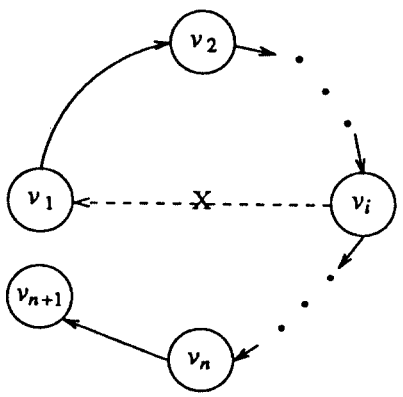

Fig. 5.

Corollary 1. Any graph $G=(V, E)$ with n vertices such that $d^+(v)=1$, $v\in V$, has at least one directed cycle C.

Theorem B For any multi-graph $G=(V, E)$ with

$d_G^+=d_G^-(v)$, $v\in V$, and p edges, we can get some cycles C_i's with m_i arcs, for $1\le i\le n$, which we eliminate from G one by one, there remains no edges in G at all in the end, i.e., there remains only a null graph at last. That is to say, $p=\sum_{i=1}^{n} m_i$, where n is the maximum number of cycles C_i.

Proof. Let us prove it by induction on the number of edges e in G.

(i)Basis : if $e=0$ then there exists a null graph. So it's O.K.

(ii)Assume that the theorem holds during $e<k$.

(iii)When $e=k$: since G has a directed cycle C with m edges, we can get subgraph $G'=(V, E-C)$ such that $d_{G'}^+(v)=d_{G'}^-(v)$. And the number of edges e' in G' becomes less than k (Note that $e'=k-m$ and $m>0$). Thus the theorem also holds when $e=k$.

From (i),(ii), and (iii) we can conclude that the theorem always holds.

References

1. A. Danthine and J. Bremer, "Modeling and Verification of End-to-End Transport Protocols," *Computer Networks*, Vol.2, 1978, pp.381-395.
2. Gregor v. Bochmann, "Finite State Description of Communication Protocols," *Computer Networks*, Vol.2, 1978, pp.362-372.
3. Carl A. Sunshine, "Formal Techniques for Protocol Specification and Verification," *IEEE Computer Magazine*, Vol.12, No.9, 1979, pp.20-27.
4. Pitro Zafiropulo, Colin H. West, Harry Rudin, D.D. Cowan, and Daniel Brand, "Towards Analyzing and Synthesizing Protocols," *IEEE Trans. Comm.*, Vol.COM-28, No.4, April 1980, pp. 651-661.
5. Colin. H. West, "General Technique for Communications Protocol Validation," *IBM J. Research and Development*, Vol.22, No.4, July 1978, pp.393-404.
6. J.Rubin and Colin H.West, "An Improved Protocol Validation Technique," *Computer Networks*, Vol.6, No.2, 1982, pp.65-73.
7. Mohamed G. Gouda and Ji-Yun Han, "Protocol Validation by Fair Progress State Exploration," *Computer Networks and ISDN Systems*, Vol.9, No.5, 1985, pp.353-361.
8. Mohamed M. Gouda and Y. T. Yu, "Protocol Validation by Maximal Progress State Exploration," *IEEE Trans. Comm.*, Vol.COM-32, No.1, January 1984, pp.94-97.
9. Si-Yeong Hwang and Jung Wan Cho, "An Improved Validation Technique for a Class of Communication Models," *Information Sciences*, Vol.39, 1986 pp.299-310.
10. CCITT Study Group XI, "Specifications of Signaling System No.7," *CCITT Red Book*, Vol. 6, Fascicle VI.7, Geneva, Switzerland, 1985.
11. Abdi R. Modarressi and Ronald A. Skoog, "Signaling System No. 7 : A Tutorial," *IEEE Comm. Magazine*, Vol.28, No.7, July 1990, pp. 19-35.
12. Philip M. Merlin, "Specification and Validation of Protocols," *IEEE Trans. Comm.*, Vol. COM-27, No.11, November 1979, pp.1671-1680.
13. Si-Yeong Hwang, "An Improved Protocol Validation Technique by Generating the Group Perturbation Tree," *Ph.D. dissertation*, KAIST, Seoul, KOREA, 1985.
14. Daniel Brand and Pitro Zafiropulo, "On Communicating Finite-State Machines," *J.ACM*, Vol.30, No.2, April 1983, pp.3323-342.
15. Yao-Tin Yu and Mohamed M. Gouda, "Deadlock Detection for a Class of Communicating Finite State Machines," *IEEE Trans. Comm.*, Vol.COM-30, No.12, December 1982, pp. 2514-2518.

이 흥 규(Heung Kyu Lee) 정회원
1955년 10월 3일생
1978년 2월 : 서울대학교 전자공학과 졸업(공학사)
1979년 2월 : 대원전기연구소 연구원
1981년 2월 : 한국과학원 전산학과 졸업(이학석사)
1984년 8월 : 한국과학기술원 전산학과 졸업(공학박사)
1985년 2월 : 한국과학기술원 대우교수
1986년 8월 : 미국 미시간 대학 Manufacturing Center연구원
1986년 9월~현재 : 한국과학기술원 조교수
1987년 8월 : 영국 Imperial College in London 교환교수
※주관심분야 : 고장허용 구실시간및 고장허용 시스템등임.

공 재 철(Jae Chul Gong) 정회원
1963년 2월 26일생
1982년~1986년 : 서울대학교 자연과학대학 계산통계학과 졸업 (B. S.)
1989년~1991년 : 한국과학기술원 전산학과 졸업 (M. S.)
1986년~1990년 : 삼성전자 통신연구소 근무
1991년 3월 : 삼성전자 정보통신부문 특수연구소 주임 연구원
※주관심분야 : Protocol engineering(esp. protocol validation), Multimedia system, System simulation, Network management(esp. tactical network management)

윤 현 수(Hyun Soo Yun) 정회원
1979년 : 서울대학교 공과대학 전자공학과 졸업
1981년 : 한국과학기술원 전산학과 석사학위 취득
1981년~1984년 : 삼성전자 연구원
1988년 : 오하이오 주립대학 전산학 박사학위 취득
1988년~1989년 : AT & T Bell Labs. 연구원
1989년~현재 : 한국과학기술원 전산학과 조교수
※주관심분야 : Parallel Computer Architecture, Interconnection Network, Protocol Engineering, Neural Network

황 시 영(Si Young Whang)
1953년 9월 27일
1976년 2월 : 서울대학교 자연과학대학 계산통계학과 졸업
1978년 2월 : 한국과학기술원 전산학과 졸업(M. S.)
1986년 2월 : 한국과학기술원 전산학과 졸업(Ph. D.)
전공분야
- Network architecture
- Protocol validation
- Distributed processing

1978년 1월~1980년 2월 : 삼성전자 컴퓨터 사업부 개발실 근무
1980년 3월~1985년 2월 : KAIST 컴퓨터 구조 연구실
1985년 3월~1987년 2월 : 삼성전자 연구소 5 연구실 근무
1987년 3월~1990년 6월 : 삼성종합기술원 정보시스템 연구 1실장
1990년 7월 : 삼성종합기술원 정보시스템 연구소 I-PROJECT실장

김 병 만(Byeong Man Kim) 정회원
1987년 : 서울대학교 공과대학 컴퓨터공학과 졸업
1989년 : 한국과학기술원 전산학과 석사학위 취득
1989년~현재 : 한국과학기술원 박사과정 재학중
1992년~현재 : 금오공과대학 전자계산기 공학과 교수
※주관심분야 : Theorem Proving, Protocol Engineering, Logic Programming, Object-Oriented Programming