論 文

수정된 maximal progress 상태 탐사 방법에 의한 개선된 프로토콜 검증 알고리즘

正會員 李 哲 熙* 正會員 李 相 鎬** 正會員 高 源 國***

An Improved Protocol Validation Algorithm by Modified Maximal Progress Sequence

Chul Hee LEE*, Sang Ho LEE**, Won Guk KO*** *Regular Members*

要 約 본 논문에서는 두개의 유한 상태 통신 기계에 대한 통신 프로토콜을 검증할 수 있도록 수정된 maximal progress 상태 탐사 방법을 제안한다. 모든 도달 가능한 상태들을 생성하는 작업은 두개의 독립된 작업으로 구분하며, 각 작업에서는 어느 한 기계에 대한 도달 가능한 상태들을 수정된 maximal progress순서에 의해서 생성되도록 한다. 이러한 maximal progress 상태 탐사 방법은 기존의 maximal progress 상태 탐사 방법보다 더 적은 시간과 공간을 요구한다.

ABSTRACT A new approach to the reachability analysis of communication protocols is presented using a modified maximal progress state exploration for two communicating finite state machines. The task of generating all reachable states is divided into two independent subtasks. In each subtask, only the states which are reachable by forcing modified maximal progress sequence for one machine are generated. Modified maximal progress state exploration saves space and time over maximal progress state exploration.

I. 개 요

통신 프로토콜이란 통신이 정상적으로 진행되도록 하기 위하여 통신 개체 사이의 상호 작용을 규정화하는 규약이다[1].

많은 통신 프로토콜은 두 유한 상태 기계를 이용하여 모형화될 수 있다. 유한 상태 기계(finite state machine)는 두개의 단반향성, FIFO 찬넬(channel)을 이용하여 메세지를 교환함으로서 통신한다.

임의의 프로토콜에서 찬넬의 용량이 유한하다면 프로토콜의 검증은 모두 reachable state 를

*, ***崇實大學校 電子計算學科
Dept. of Computer Science, Soong Sil University
**忠北大學校 電算統計學科
Dept. of Computer Science and Statistics, Chung Buk National University
論文番號 : 88-23(接受1988. 5. 20)

생성시켜 이들을 조사함으로서 nonprogress 여부를 검증할 수 있다. 이와 같은 방법을 상태 탐사(state exploration)라 한다[3],[4]. 이 방법에서의 제일 큰 문제점은 프로토콜의 복잡도가 높아지면 상대적으로 생성해야할 reabhable state의 수가 너무 많아져 시간과 공간의 요구가 증대된다는 점이다. 이러한 문제를 해결하기 위하여 Rubin과 West는 두 기계의 상태의 변이가 같은 속도로 이루어진다는 가정하에 canonical sequence라는 개념을 제안하여 모든 reachable state를 생성하지 않고 프로토콜을 검증할 수 있는 방법을 고안하였다[5]. 이 방법은 시간과 공간의 측면에서 종래의 방법보다 개선을 이루었지만 같은 속도의 변이라는 가정으로 인하여 모든 nonprogress(deadlock, overflow, unspecified reception)를 탐지하지 못하는 단점을가진다. Gouda와 Y.T.Yu는 상태생성을 두개의 독립적인 subtask로 나누어 각각의 task에서 maximal progress를 형성하는 reachable state만을 생성하여 이들을 이용하여 nonprogress의 여부를 조사하는 방법을 제안하였다[6].

본 논문에서는 Gouda와 Y.T.Yu의 방법에서의 maximal progress sequence의 개념을 확장하여 보다 적은 크기의 reachable state 집합을 생성시켜 nonprogress 여부를 조사하는 방법을 제시한다. 이 방법은 Rubin과 West 방법의 제한성인 overflow를 탐지할 수 있으며 Gouda 와 Y.T.Yu의 방법보다는 더 적은 reachable state 집합을 운용함으로서 시간과 공간의 요구도를 줄일 수 있다.

Ⅱ. 통신기계

통신기계(communicating machine) M과 N은 각각 sending과 receiving edge라 부르는 두가지 형태의 edge를 가지는 directed labeled 그래프이다[3]. Sending edge는 메세지 집합G의 어떤 원소 g로서 레이블이 send(g), receiving edge는 receive(g)로 표현된다. M과 N의 각 노드(node)는 적어도 하나의 output edge를 가져야 하며 동일 노드의 output들은 레이블이 서로 달라야 한다. 어떤 노드의 모든 output들이 sending(혹은 receiving)인 경우 그 노드는 sending 노드(혹은 receiving 노드)라 부르며, sending과 receiving output이 혼재하는 노드는 mixed 노드라 부른다. M과 N은 각각의 노드중의 하나를 초기(initial)노드라 하고 M과 N의 각 노드는 그것의 초기 노드로부터 방향성 경로에 의하여 reachable이다. M과 N의 상태는 4-tuple〔v, w,x, y〕로 표현된다. 여기서 v는 M의 노드, w는 N의 노드이고 x와 y는 각각 집합 G의 메세지 열(string)로서 |x|≤K이고 |y|≤K이다. 여기서 |x|(혹은 |y|)는 메세지 열 x의 길이 (혹은 y의 길이)를 의미하며 K는 양의 정수로서 채널 용량이다.

초기 상태는 〔v, w, E, E〕로 표현되며 E는 메세지가 없음(empty string)을 뜻한다. 논리의 단순성을 위하여 채널의 용량은 유한하고 또 error- free로 가정한다. s=〔v, w, x, y〕가 M, N 의 임의 상태이고 e가 노드 v나 w의 output edge라 하자. 상태 $\hat{s}$이 e로써 s에 후행(follow over)한다는 것은 $s-e\rightarrow\hat{s}$으로 표현되며 이는 다음의 네가지 조건 중 어느 하나와 대등(equivalent)하다 :[7],[8].

1) 만일 e가 v에서 v′으로 가는 M의 레이블이 send(g)인 sending edge라면 |y|<K이고 s′=〔v′, w, x, y.g〕이다. 여기서 "."는 접합(concatenation) 연산자이다.

2) 만일 e가 w에서 w′으로 가는 N의 레이블이 send(g)인 sending edge라면 |x|<K이고 s′=〔v, w′, x.g, y〕이다.

3) 만일 e가 v에서 v′으로 가는 M의 레이블이 receive(g)인 receiving edge라면 x=g.x′이고 s′=〔v′, w, x′, y〕이다.

4) 만일 e가 w에서 w′으로 가는 N의 레이블이 receive(g)인 receiving edge라면 y=g.y′이고 s′=〔v, w′, x, y′〕이다.

s와 s′이 M과 N의 두 상태일때 s′이 s로부터 follow라고 하는 것은 $s\rightarrow s'$으로 표기하며 이는 M혹은 N에 방향성 edge가 있어 $s-e\rightarrow s'$이라는 것과 대등하다. 또 s′이 s로부터 reachable이라고 하는 것은 s=s′ 혹은 $s=s_1$, $s'=s_r$이고 i=1, ... r-1에 대하여 $s_i\rightarrow s_{i+1}$인 상태들

$s_1, s_2, \ldots, s_r$이 존재한다는 것과 대등하다. M과 N의 상태 s가 reachable이라고 하는것은 초기상태로부터 reachable이라는 것과 대등하다.

M과 N사이의 두 찬넬이 유한 용량을 가지므로 모든 reachable state 집합도 역시 유한하다. 따라서 통상적인 상태 탐사 기법을[9]을 적용하여 모든 reachable state를 생성시켜 이들로부터 어떤 상태가 nonprogress 상태인지를 조사할 수있다. nonprogress 상태의 유형에는 deadlock state, unspecified reception state와 overflow state가 있다[7]. 어떤 상태 s=〔v, w, x, y〕는 v와 w 노드가 모두 receiving 노드이고 x=y=E 일 때 deadlock state라고 한다. 또 어떤 상태 s=〔v, w, x, y〕가 unspecified reception state 라고 하는 것은 다음의 두 조건 중 어느 하나를 만족할 때이다.

i) $x=g_1 \cdot g_2 \ldots g_r$이고 v가 reveiving 노드이며 그 output edge의 label이 receive(g_1)인 것이 없다.

ii) $y=g_1 \cdot g_2 \ldots g_r$이고 w가 receiving 노드이며 그 output edge의 label 이 receive(g_1)인 것이 없다.

어떤 상태 s=〔v, w, x, y〕가 overflow state라고 하는 것은 다음의 두 조건 중 어느 하나를 만족할 때이다.

i) 노드 v가 sending 노드이며 |y|=K이다.

ii) 노드w가 sending 노드이며 |x|=K이다.

Ⅲ. Modified Maximal Progress Sequence

Y.T.Yu는 maximal progress sequence를 다음과 같이 정의하였다[7]. M에 대한 maximal pro progress sequence는 다음과 같은 M과 N에 대한 상대적 실행순서이다. : 첫째, M이 가능한한 최대로 실행한다. 즉 M의 입력 찬넬이 empty이며 receiving node에 도달할 때까지 진행한다. 그리고 나서 N이 M에 정확하게 한개의 메세지를 송신할 때까지 진행한다 ; 즉 M의 실행이 재개될 수 있다. 이와 같은 과정 이 되풀이 된다.

이와 같이 정의된 MPS(maximal progress sequence)를 이용하여 nonprogress state 의 존재성을 효율적으로 검증할 수 있는 알고리즘을 제안하였다.

본 논문에서는 Y.T.Yu와 Gouda가 제안한 MPS의 개념을 확장하여 보다 적은 갯수의 state로서 nonprogress 여부를 검증하는 알고리즘을 제안키로 한다.

Modified MPS를 다음과 같이 정의한다. M과 N이 유한 용량 찬넬을 이용하여 통신하는 기계라 하자. 그리고 다음과 같은 조건을 만족하는 Q를 modified MPS라 정의한다.

i) $Q=q_1 \cdot q_2 \ldots q_n, n \geq 1$
$q_i (i=1, 2, \ldots, n)$는 기계 M 혹은 N의 receiving 혹은 sending edge이고 empty 일 수 있음.

ii) k=1, 2, . . . , m에 대하여 sequence $q_1 \cdot q_2 \ldots q_k (k<n)$은 반드시 상태〔v, w, x, y〕에 도달되며 v는 receiving 노드이고 x=E이다.

iii) j=k+1, k+2, . . . , n에 대하여 sequence $q_{k+1} \cdot q_{k+2} \ldots q_n$은 상태〔v′, w′, x′, y′〕에 도달되며 v′는 receiving 노드이고 x′≠E 이다.

아래의 3개 정리는 어떤 nonprogress state가 reachable이면 그 nonprogress state는 modified MPS에 의해서도 reachable 임을 보인다.

정리 1 : deadlock state가 reachable이면 modified MPS에 의해서도 reachable이다.

정리 2 : unspecified reception state가 reachable이면 modified MPS에 의해서도 reachable이다.

정리 3 : overflow state가 reachable이면 modified MPS에 의해서도 reachable 이다.

Ⅳ. State Exploration Algorithm

M과 N이 유한 용량 찬넬을 이용하여 통신하는 두 기계라 할 때 nonprogress state에의 도달 여부를 결정하는 알고리즘은 다음과 같다. 이 알고리즘은 M과 N이 mixed 노드를 가지는 경우를 고려하여 modified MPS에 의한 모든 reachable state를 생성하여 조사하는 기능을 갖는다. 여기서 PRO는 이미 조사된 reachable state 를 표현하는 집합이고, INIT는 최초에는 초기상태를 가지나 알고리즘이 수행되면서 reachable state들을 보관하는 집합이고, NON은 nonprogress state를 저장하는 집합으로서 초기값은 공(null) 이다.

```
procedure detectnonprogress ;
  Var      PRO, INIT, NON/* set type
           varibles */
  procedure   maximal(X)/* X is a
              machine */
           while INIT is not empty do
           remove one state s from INIT
           if s is a nonprogress state then
           add s to NON and skip
           if s is in PRO then skip
           else add s to PRO ;
             case s=〔v, w, x, y〕of
             a : v is a sending node or
                 v is a receiving or mixed node and x≠E :
                 generate all states s′ such
                 that s-e-s′,
                 and e is in machine M ;
                 add all s′ to INIT
             b : v is a receiving node and x=E :
                 generate all states s′ such
                 that s-e→s′,
                 and e is in machine N ;
                 add all s′ to INIT
             c : v is a mixed node and x=E :
                 generate all states s′ such
                 that s-e→s′,
                 and e is in machine M or N;
                 for each generated s
                 do let s-e→s′ ;
                 if e is in M then add s′ to
                 INIT ;
                 elseif s′ is in INIT or PRO
                 then skip
                 else write s′ as 〔vR, w′, x′,
                 y′〕
                 and add it to INIT ;
                 /* vR is a receiving mixed
                 node */
             d : v is a receiving mixed node
                 and x=E :
                 generate all states s′ such
                 that s-e→s′,
                 and e is a receiving edge in
                 M ;
                 add all s′ to INIT
             e : v is a receiving mixed node
                 and x=E and
                 w is a receiving node and y
                 =E :
                 skip
             f : v is a receiving mixed node
                 and x=E and
                 (w is not a receiving node or
                 y≠E) :
                 generate all states s′ such
                 that s-e→s′,
                 and e is in machine N ;
                 for each generated s′
                 do let s′=〔vR, w′, x′, y′〕
                  if 〔v′, w′, x′, y′〕 is in INIT or
                  PRO
                  then skip
                  else add s′ to INIT
  endcase ;
     endwhile
  end maximal ;
     set PRO, NON to empty ;
```

```
    set initial state to INIT ;
    call maximal [M] ;
    call maximal [N] ;
    print PRO
end detectnonprogress ;
```

위 알고리즘은 임의의 상태 s=[v, w, x, y]로부터 기계 M과 N이 동시가 아니라 각각 진전할 수 있다. 각 상태로부터 항상 하나의 기계만이 실행가능하므로 생성되는 상태의 수는 종래의 상태 탐사 방법보다 작으며 Y.T.Yu의 알고리즘보다도 더 작다. 그 이유는 Y.T.Yu의 알고리즘에서는 기계 M의 진전이 불가능할 때 기계 N은 반드시 하나의 메세지를 송신할때까지 실행하나 modified 방법에서는 기계 N도 역시 실행 불가능할때까지 진전되도록 하였기 때문이다. 또한 모든 nonprogress state의 조사가 가능하다.

V. 사례 연구 및 검토

그림 1. (a)와 (b)의 두 통신 기계 M, N을 생각하자. 여기서 M과 N의 노드 1은 mixed node이다. 또 M과 N은 두 찬넬을 이용하여 통신하며 두 찬넬의 용량은 각각 3이라고 하자.

그림 1. (c)는 기계 M에 대한 modified maximal progress sequence를 적용하여 얻은 reachable state를 보이며, N에 대한 것은 그림 1. (d)에 보인다.

그림 1. (c)와 (d)에 있는 모든 state가 progress state이므로 M과 N은 nonprogress state에 도달하지 않는다. 편의상 메세지 $g_1, g_2, \cdots$는 그림에서 1, 2로 표현하였다.

그림 2. (a)와 (b)에는 Y.T.Yu의 알고리즘에 의한 기계 M과 N에 대한 reachable state tree가 보여진다[7]. 두 방법을 비교하면 Y.T.Yu의알고리즘에서는 모두 40개의 상태가 생성된 반면에, 본 논문의 알고리즘에서는 모두 25개의 상태가 생성되어 약 37%의 기억 공간 및 시간의 절약을 이루었음을 알 수 있다.

VI. 결 론

통신 프로토콜은 유한 상태 기계로 표현하여 검증이 가능하다. 표현된 유한 상태 기계를 이용하여 도달 가능한 모든 상태를 조사하여 progress 여부를 검증하는 방법을 상태 탐사방법이라 한다. 이 방법의 단점은 상태 폭발 문제이다.

본 연구에서는 Y.T.Yu의 알고리즘에 maximal progress sequence의 개념을 확장함으로서 보다 적은 갯수의 상태만으로도 프로토콜의 검증이 가능함을 보임으로써 상태 탐사 방법이 갖는 상태 폭발 문제를 보다 효율적으로 개선시켰다. 그러나 본 연구에서는 2개의 통신 기계 (M, N) 만을 고려 하였으며 3개 이상의 통신기계를 갖는 통신 모형에서도 적용 가능한 프로토콜 검증 알고리즘의 확장이 앞으로의 연구 과제이다.

參 考 文 獻

(1) G.V. Ramamoorthy, Y. Yaw and W.T. Tsai, "A Petri Net Reduction Algorithm for Protocol Analysis," ACM SIGCOMM, pp. 157-166, 1986

(2) G. V. Bochman, "Finite State Description of Communication protocol," Computer Networks, Vol. 2, pp. 361 361-371, 1978

(3) D. Brand and P. Zafiropulo, "On Communicating Finite State Machine," JACM, Vol. 30, pp. 323-342, Apr., 1983

(4) M. G. Gouda and Y. T. Yu, "Maximal-Progress state Exploration," Proc. ACM SIGCOMM, pp. 68-75, 1983

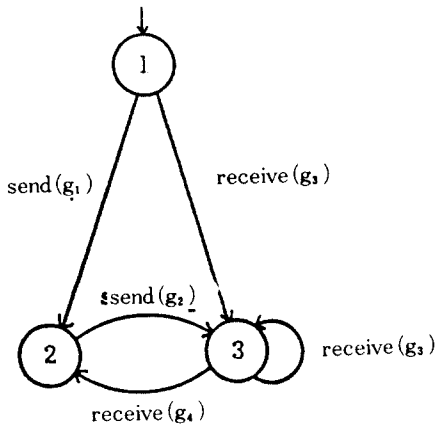
(5) J. Rubin and C. H. West, "An Improved Protocol Validation Techniques," Computer Networks, vol. 6, pp. 65-73, Apr., 1982

(6) M. G. Gouda and Y. T. Yu, "Protocol Validation by Maximal Progress State Exploration," IEEE Trans. Comm., Vol. COM-32, pp. 94-97, Jan., 1984

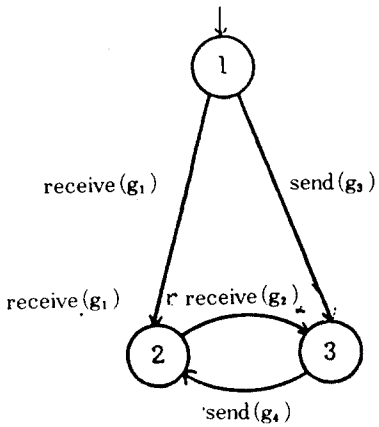
(7) Y. T. Yu, "Communicating Finite State Machine: Analysis and Synthesis," Ph. d. Dissertation, Dept. Computer Science, Univ. Texas at Austin, Jan., 1983

(8) Y. T. Yu and M. G. Gouda, "Deadlock Detection for a Class of Communicating Finite State Machines," IEEE Trans. Comm., Vol. COM-30, pp. 2514-2518, Dec., 1982

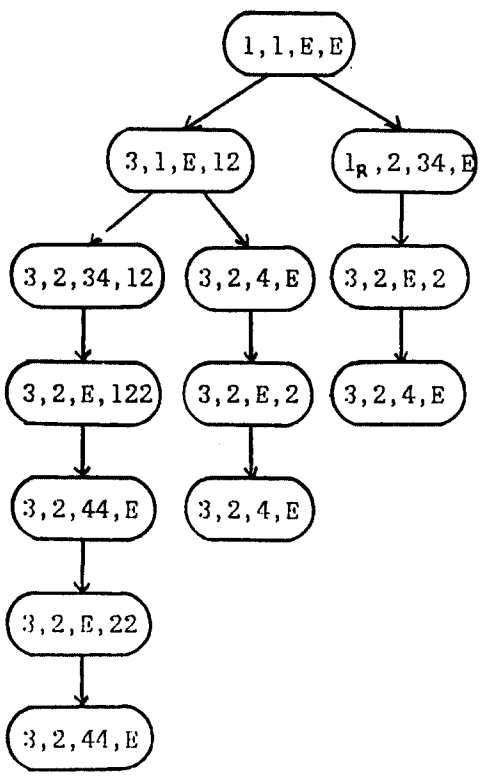
(9) C. H. West and P. Zafiropulo, "Automated Validation of a Communications Protocols," IBM J. Res and Develop., vol. 22, pp. 60-71, Jan., 1978 Vol.



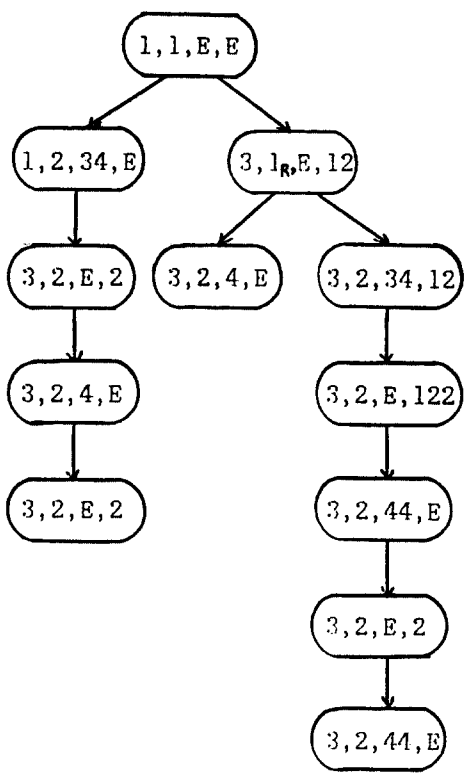
(a) M



(b) N

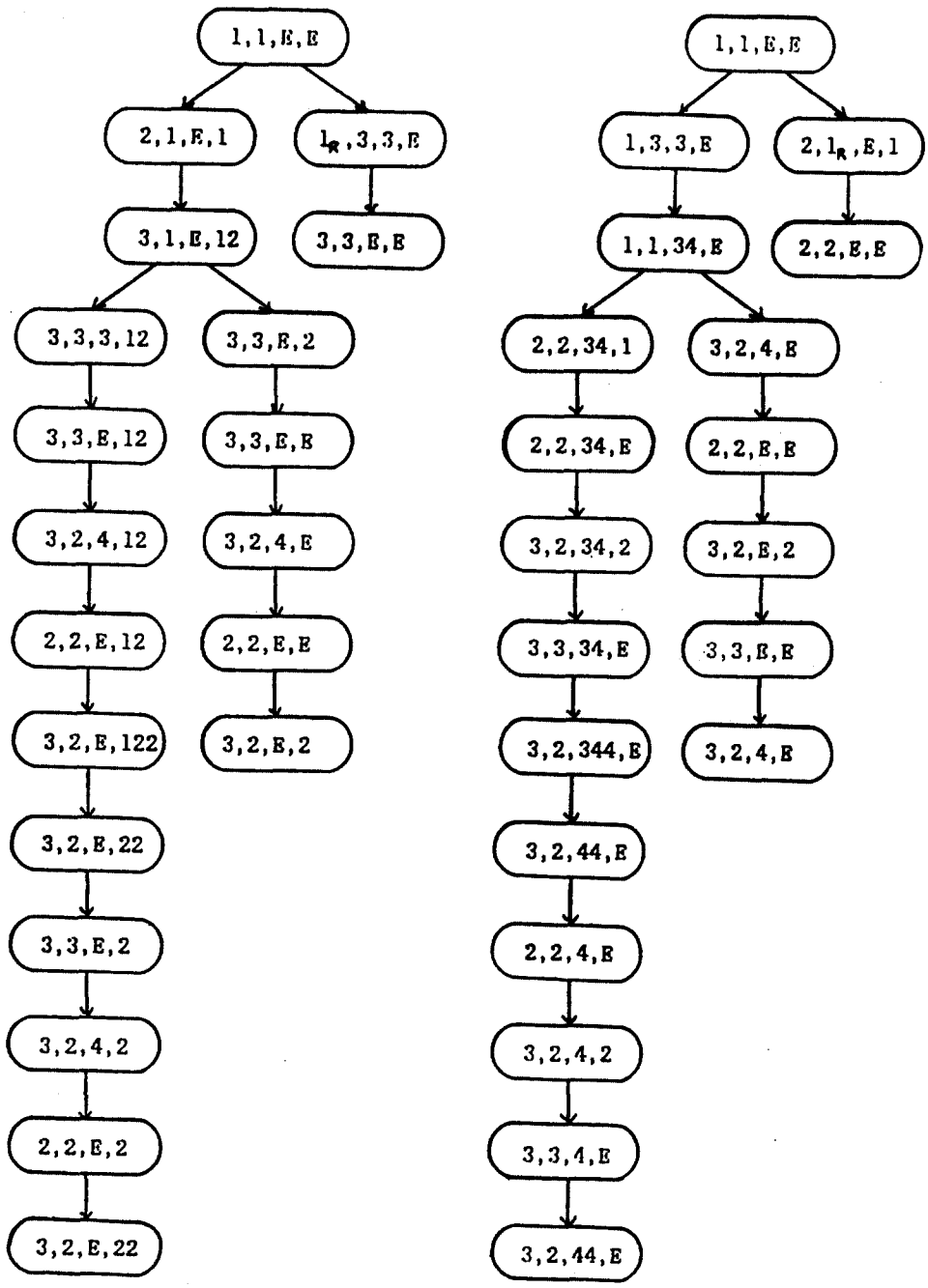


(c) state exploration for M



(d) state exploration for N

그림 1 기계M과 N의 예
Example for machine M and N.



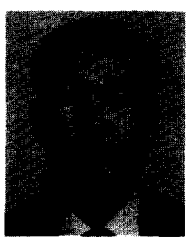
(a) state exploration for M

(b) state exploration for N

그림 2 Y. T. Yu의 reachable 상태도
Reachable state tree of Y. T. Yu.

李 哲 熙(Chul Hee LEE) 正會員
1934年 4 月18日生
1958年 6 月：陸軍士官學校(理學士)
1962年 8 月：美Purdue大學校大學院 電氣工學科(工學碩士)
1988年 2 月：中央大學校 大學院 電子計算學科(理學博士)
1962年 9 月～1973年 2 月：陸軍士官學校 電子工學科 教授
1973年 3 月～現在：崇實大學校 電子計算學科 教授

李 相 鎬(Sang Ho LEE) 正會員
1953年 3 月15日生
1976年：崇實大學校電子計算學科 卒業
1981年：崇實大學校 大學院 電子計算學科 卒業
1985年～現在：崇實大學校 大學院 電子計算學科博士 課程 中
1976年～1979年：韓國電力電子計算所勤務
1981年～現在：忠北大學校 電算統計學科 助教授



高 源 國(Won Kuk KO) 正會員
1952年 1 月25日生
1975年：陸軍士官學校 卒業
1979年：成均館大學校經營行政 大學院 電子資料處理學科 卒業
1986年～現在：崇實大學校 大學院 電子計算學科 博士 課程 中
1978年～1985年：陸軍中央電算處理所勤務