An algebraic approach to validate communication protocols

A. Benslimane

Laboratoire d'Informatique, UFR des Sciences et Technique
16 route de Gray, 25030 Besançon, France

Abstract- *Realizing the need for a precise modelisation of computer communications protocols, many different approaches are being further investigated. In this paper, we present an algebraic approch to formally specify protocols composed of many modules. The protocols studied are described in the model of finite state machine and then transformed into a set of process equations. The main concept, which is underlying here consists in carrying out transformations on these equations in order to prove some properties. The reachability analysis can then be defined in the form of algebraic transformation rules applied to the global system states. The properties to verify during the communication between processes are the detection of some logical errors such deadlocks, blocking or unspecified receptions.*

1. Introduction

The trend toward distributed systems and open communications networks has greatly increased the complexity of data communication protocols. It is thus essential to find mathematically precise and automated methods of specification and validation of protocols to overcome this complexity. For this, several models have been successfully used to study the communication protocols. They are the " state transition" model, the "programming langage" model, and the combination of the above two models. General surveys of these three approaches can be found in [13]. The main advantage of state transition models is that the validation of communication protocols can be easily automatized. However, the price paid is the so called "state space explosion" problem, in which, as the complexity (number of states and transitions) of the protocol increases, the number of systems states grows exponentially. On the other hand, programming langage models are free from state space explosion problem, but they create difficulty in automating verification process system. They require special knowledge in the the area of program proving. In thestate transition model, several approaches have been proposed. Examples most studied are Petri Nets and Finite state Machines. In the formalism of Petri nets [1, 2], each process is represented in the form of a Petri net and events are also denoted by places and bars of Petri net. Some properties are decidable in this model. In this case, Karp and Miller [3] have proposed a solution for decidability problem of a boundedness. In the models using finite state machines [4, 5, 7], each process is represented in the form of state transition machine and events (transmission or reception) cause a state transition. In this formalism, only certain protocols can be described (for example the management of the sequence numbers in the transport protocol cannot be described). In spite of the difference in modeling formats, the validation technique in both models is the same and consists in a reachability analysis where all the global states reachable from the initial one are generated exhaustively in order to detect logical errors like deadlock states, blocking or unspecified reception states, overflow and state ambiguities. Usually, the state generation is performed by the construction of a reachability tree [4, 6, 7, 8] according to certain progress rules. However, in practice the terminaison of the construction of reachability tree of protocols specified in terms of communicating finite state machines can be infinite.

In this paper, we relate our discussion to the model of finite state machines in which we gives briefly some analysis and reduced techniques.

Zafiropulo [9] introduced a validating method using a duologue matrix analysis for protocol consisting of only two processes. This technique is currently limited to protocols that must revert to an initial or quiscent state after a number of interaction steps.

An extended state transition model and verification algorithm was proposed by Itho and Ichikawa [10]. Their model is an extension of Zafiropulo's two process protocol model. Their theory based on the concept of reduced implementation sequences reduces the complexity of the reachability analysis. Zaho and Bochmann in [11] have presented a new approach to validate communication protocols consisting of only two processes. Their method uses a representation of protocol specification in the form of process equations and defines on algebraic

0-8186-2755-7/92 $03.00 © 1992 IEEE

verification approaches have been proposed to deal with the state space explosion problem resulting from the reachability analysis. Here, we introduce a now form of the reachability analysis. A transformation from a global state to another consists in a parallel evolution of all processes with equal speeds. Then, the obtained number of global states becomes much shorter than that required for the method based on the classical reachability analysis. In our model, the global state transition is composed by a parallel possible transition of all processes, in contrast to the classical method in which the state transition concerned for individual processes.
The rest of the paper is organised as follows : in section II, we introduce the process equations to represent the state transitions of communicating processes and we give some definitions of design errors. Section III presents the reachability analysis defined on the form of algebraic transformation rules. Section IV discusses the proposed reduced reachability analysis. In section V, we present an algorithm that uses the algebraic transformation rules to validate a given protocol. To conclude this paper an example for comparison is also given.

2. Protocol model

Communication protocols can be modeled as communicating finite state machines consisting of processes and channels between the processes. For every channel, the order of messages received is the same as the order of those transmitted. This assumption means FIFO message transmission. The time required for these transitions is supposed zero.
Each process of system can be represented as an automata consisting of state transitions (possible interactions of message sending and receiving). The form of each automata is based on Zafiropulo's protocol model.

2.1. Process equations representing state transitions

Informally, a communicating machine is a directed labeled graph with two types of edges called sending and receiving edges. Each node in the graph must have at last one output edge, and outputs of the same node must have distinct labels.
We distinguish three types of node :
- sending node : if all its outputs are sending edges,
- receiving node : if all its outputs are receiving edges, and
- mixed node : if its outputs are sending and receiving edges.

Now, we define formally the model of process P_i as a 4-uple $<S_i,I_i,A_{i,0},\delta_i>$ where
- $E_i = \{A_{i,0},A_{i,1},...\}$ denotes the finite set of states of process P_i.
- I_i denotes the finite set of possible interactions of process P_i. It is the set of transitions written in the form "$-a_{ij}$' that can be sent from process P_i to process P_j and those written in the form '$+a_{ij}$' that can be received by process P_i from process P_j.
- $A_{i,0}$ is the initial state of process P_i.
- δ_i is a partial function mapping for each i,

$\delta_i : S_i \times I_i \longrightarrow S_i$ $\delta_i(A_{i,k},\pm a_{ij})$ is the state entered after a process P_i transmits (to process P_j) or receives (from process P_j) message ' a_{ij}' in the current state $A_{i,k}$.
The state transition of process P_i can then be represented by process equations. We assume that the process equations are :

$$A_{i,k} = \sum_{A_{i,l}\in SI_i(A_{i,k})} a^{kl}_{ij}\, A_{i,l} \quad \text{for } i=1..N$$

where $SI_i(A_{i,k})$ denotes the set of the direct successors of state $A_{i,k}$ for the process P_i. 'a^{kl}_{ij}' is the state transition passing the process P_i and communicates with another process P_j, from $A_{i,k}$ to $A_{i,l}$.
a^{kl}_{ij} is written in the form $-a_{ij}$ (resp. $+a_{ij}$) to denotes an emission (resp. reception) of message 'a' from process P_i to process P_j (resp. by process P_i from process P_j).
We use the following notations :

(1) $E_i=<A_{i,k},<s_{ij}>_{j=1..N;j\neq i}>$ to represent a partial behavior, for a process P_i. It is a N-uplet consisting of current state $A_{i,k}$ of process P_i and N-1 event sequences s_{ij} sent by P_i to a process P_j (j=1..N and $j\neq i$).

(2) $E=<E_i>_{i=1..N}$ to represent a global behavior, for a system consisting of N process. $<A_{i,0},(\lambda,\lambda,...,\lambda)>_{i=1..N}$ represents the initial global behavior, where each message queue is empty.
For example, in figure 1, the initial partial behaviors for process P_1, P_2 and P_3 are respectively $<A_{1,0}, (s_{12}=\lambda, s_{13}=\lambda)>$, $<A_{2,0}, (s_{21}=\lambda, s_{23}=\lambda)>$ and $<A_{3,0}, (s_{31}=\lambda, s_{32}=\lambda)>$. The composition of those partial behaviors gives the initial global behavior $<<A_{1,0}, (\lambda,\lambda)>, <A_{2,0}, (\lambda,\lambda)>, <A_{3,0}, (\lambda,\lambda)>>$, where λ denotes the empty sequence.
For a simple notation, we write the process equations as follows :

$$A_{i,k} = \sum_{li} a^{li}_{ij}\, A_{i,l_i} \text{ for } i=1..N ;$$

$A_{i,l_i}\in SI_i(A_{i,k})$; j=1..N ($j\neq i$) .

2.2. Types of design errors

We can handle three potential design errors, namely deadlock states, blocking reception states and unspecified reception states.

Let $E=<A_{i,k},<s_{ij}>_{j=1..N;j\neq i}>_{i=1..N}$ a global behavior, then

- E is a deadlock state if and only if $\forall$ i $(\sum_{l_i} a^{l_i}_{ij} A_{i,l_i})$, $\forall$ l_i $a^{l_i}_{ij} > 0$ and $\forall$ j $(j\neq i)$, $s_{ij}=\lambda$.
- E is a blocking reception state if and only if $\exists$ i $(\sum_{l_i} a^{l_i}_{ij} A_{i,l_i})$, $\forall$ l_i $a^{l_i}_{ij} > 0$ and $\nexists$ j $(j\neq i)$ / $s_{ji}=(- a^{l_i}_{ij}).X$.
- E is an unspecified reception state if and only if $\exists$ i, $\exists$ j $\neq$ i such that : $s_{ji}=(-a_{ij}).X$, $A_{i,k}= \sum_{l_i} a^{l_i}_{ij} A_{i,l_i}$ and $\forall$ l_i $a^{l_i}_{ij} \neq (+a_{ij})$.

Remarks

- In those error definitions, we are not interested in their semantics but we are concerned with their syntactic forms.
- In our analysis, a blocking reception state when it occurs for only one process does not block the global system evolution. The corresponding process remains indefinitely in the same state until at last one event becomes admissible if any. But the other processes can evoluate if it is possible.

3. The proposed method

Now, we define the transformation rule for reachability applied to a partial behavior as follows :

(TP) $<A_{i,k},<s_{ij}>_{j=1..N;j\neq i}> \longrightarrow < \sum_{l_i} a^{l_i}_{ij} A_{i,l_i} , <s_{ij}>_{j=1..N ;j\neq i} >$

The algebraic transformation rule applied to a partial behaviors :

(ATP) $<\sum_{l_i} a^{l_i}_{ij} A_{i,l_i} , <s_{ij}>_{j=1..N;j\neq i}> \longrightarrow \sum_{l_i} < A_{i,l_i} , <s_{ij}. a^{l_i}_{ij} >_{j=1..N;j\neq i} >$

Thus, a transformation for the partial behavior consists of the application of the reachability transformation rule (TP) followed by the algebraic transformation (ATP).

Then, we define the basic algebraic transformation rules applied to a global behaviors :

(ATG1) $< \sum_{l_i} < A_{i,l_i} , <s_{ij}. a^{l_i}_{ij} >_{j=1..N} > >_{i=1..N} \longrightarrow \sum_{l_1} ... \sum_{l_i} ... \sum_{l_N} < A_{i,l_i} , <s_{ij}. a^{l_i}_{ij} >_{j=1..N;j\neq i} >_{i=1..N}$

(ATG2) $\alpha + \varepsilon \longrightarrow \alpha$

(ATG3) $\varepsilon + \alpha \longrightarrow \alpha$

The symbol a denotes an arbitrary correct global behavior and e denotes an erronous global behavior.

To simplify the reduction rules, we write the following equivalence between global behaviors :

$< A_{i,l_i} , <s_{ij}>_{j=1..N;j\neq i} >_{i=1..N} \Leftrightarrow < A_{i,l_i} , <s_{ij},s_{ji}>_{j=1..N;j>i} >_{i=1..N}$

Here, a global behavior is a N-uplet. Each i^{th} uplet consists of current state for a process P_i and N-i paires of sequences $< s_{ij},s_{ji}>$ with j>i.

s_{ij} denotes the sequence of transitions representing sending messages from P_i to P_j.

s_{ji} denotes the sequence of transitions representing sending messages from P_j to P_i.

Also, we define the transformation rules for reduction, applied to a global behaviors as follows :

(RG1) $< A_{i,l_i} , <(-a_{ij}).s_{ij},s_{ji}.(+a_{ji})>_{j=1..N;j>i} >_{i=1..N} \longrightarrow < A_{i,l_i} , <s_{ij},s_{ji}>_{j=1..N} >_{i=1..N}$

(RG2) $< A_{i,l_i} ,< s_{ij} .(+a_{ij}), (- a_{ji}).s_{ji}>_{j=1..N} >_{i=1..N} \longrightarrow < A_{i,l_i} , <s_{ij},s_{ji}>_{j=1..N} >_{i=1..N}$

(RG3) $< A_{i,l_i} , <s_{ij},s_{ji}>_{j=1..N;j>i} >_{i=1..N} \longrightarrow \varepsilon$ iff $\forall i , \exists j /$ $(s_{ij}=X.(+a_{ij})$ and $s_{ji} \neq (-a_{ji}).Y)$.

Finally, to simplify the global behaviors, we introduce the following transformation rules for simplification applied to a global behaviors :

Let $A_{i,k} = \sum_{l_i} a^{l_i}_{ij} A_{i,l_i}$ $(j\in [1..N] ; j\neq i)$ and $A_{h,k} = \sum_{l_h} a^{l_h}_{hp} A_{h,l_h}$ $(p\in [1..N] ; p\neq h)$ $(h\neq i)$.

Then,

(SR1)

$< A_{i,l_i} , < s_{ij} .(+a_{ij}), s_{ji}>_{j=1..N} >_{i=1..N}$
$\longrightarrow < A_{i,k} , <s_{ij}, s_{ji}>_{j=1..N} >_{i=1..N}$
where $A_{i,l_i} = \delta_i(A_{i,k},'+a_{ij}')$

if $\forall l_i$, $(\forall h \in [1..N]\ (h \neq i) / a^{l_h}{}_{hi} = '-a_{hi}') : a^{l_i}{}_{ih} \neq '+a_{ih}'$.

(SR2)

$< A_{i,l_i} , < s_{ij} .(+a_{ij}), s_{ji}>_{j=1..N} >_{i=1..N}$
$\longrightarrow \varepsilon$ where $A_{i,l_i} = \delta_i(A_{i,k},'+a_{ij}')$

if $A_{i,k}$ is a sending edge or $\exists h \in [1..N]$ $(h \neq i) / a^{l_i}{}_{ih} = '+a_{ih}'$ and $a^{l_h}{}_{hi} = '-a_{hi}'$.

Definitions

- A transformation step for the global behavior consists of the transformation of each partial behavior, followed by all possible basic algebraic transformations, reduced transformations applied to global behaviors and by a possible transformation rule for simplification.

- A global behavior $E = <A_{i,k},<s_{ij}>_{j=1..N}>_{i=1..N}$ is directly reachable from a global behavior $E' = <A'_{i,k},<s'_{ij}>_{j=1..N}>_{i=1..N}$ if and only if E can be obtained by one and only one transformation step from E'.

- A global behavior E is reachable if and only if there exists a sequence of global behaviors $\alpha_1, ..., \alpha_p$ such that $\alpha_1 = <A_{i,0},(\lambda, \lambda,..., \lambda)>_{i=1..N}$ (initial global behavior), $\alpha_p = E$ and α_{t+1} is directly reachable from α_t for t =1,..., p-1.

From the above definitions, we obtain the following result.

Theorem. *An arbitrary global behavior E, is a deadlock state or a blocking reception state for all processes if and only if it becomes ε after one transformation step.*

Demonstration-

Let $E = <A_{i,k},<S_{ij}>_{j=1..N;j\neq i}>_{i=1..N}$ a global behavior which becomes ε after one transformation step. By applying the rules (TP) and (ATG1) we obtain the global behaviors as follows :

(1) $\sum_{l_1} \sum_{l_i} \ldots \sum_{l_N} < A_{i,l_i} , <s_{ij}. a^{l_i}{}_{ij}>_{j=1..N;j\neq i}>_{i=1..N}$

Thus, by applying the algebraic transformation rules (ATG2) and (ATG3) the expression (1) becomes ε if :

$\forall l_i < A_{i,l_i} , <s_{ij}. a^{l_i}{}_{ij}>_{j=1..N;j\neq i}>_{i=1..N}$
$\longrightarrow \varepsilon$. By the rule (RG3),

$< A_{i,l_i} , < s_{ij}. a^{l_i}{}_{ij}>_{j=1..N ;j\neq i}>_{i=1..N}$ becomes ε if $\forall i$, $\exists j$ / $a^{l_i}{}_{ij} > 0$. This implies that E becomes ε if $\forall l_i$, $\forall i$, $\exists j$ / $a^{l_i}{}_{ij} > 0$ wich is equivalent to $\forall i$, $\forall l_i$ $a^{l_i}{}_{ij} > 0$. Moreover, If we have $\forall i$ and $\forall j$ $s_{ij} = \lambda$, then from the deadlock definition we deduce that E is a deadlock state. If not, it is clear that E is a blocking reception state, for all the communicating processes.

4. The reduced reachability

4.1. On the construction

The major difficulty in the construction of the reachability tree lies in the so-called state explosion problem. This occurs when the number of states that may need to be analyzed becomes impracticably large. For some complex protocols, the construction of reachability tree is unbounded. This may represents all global states that can be reached by the system.

The principal technique of reachability analysis was the perturbation technique [4,9]. Its consists on the "perturbation" of a global state into all possible successor states reachable by executing a single transition in one of individual process.

Itoh and Ichikawa [10] introduce another technique based on the concept of "Reduced Implementation Sequences". Using this technique, potential movements of modeled systems are compactly described and the number of global states was reduced.

Our technique allows a minimum number of global states. Based under the assumption that the time required for a transitions is zero, the transmission of a message and its reception are executed at the same time in one global transition, if the corresponding channel is empty. This means that any message ready to be received by a process should be accepted by a receiving transition.

4.2. On the termination

The problem is solvable for the class of protocols with all bounded channels . It's also solvable for the class of protocols with two processes and only one channel is unbounded. In contrast the problem is unsolvable for the protocols with a number of processes more than two, when no channel is bounded.

For those classes of protocols where a complete solution of construction of the reachability tree is not available, one can provide approximate solutions. For example

West [12] uses a channel bound as a parameter of validation.
In all general techniques of reachability analysis each global state reached during the exploration is analyzed for errors after verifying it has already or not observed in the validation.
To avoid a part of state explosion problem, we gives the following proposition .

Proposition.
Let $E'=<A'_{i,k},<S'_{ij}>_{j=1..N;j\neq i}>_{i=1..N}$ *be a global behavior reached by the exploration. Then we can stope the analysis of E' and the creation of it's successor global behaviors if there exists a global behavior*
$E=<A_{i,k},<S_{ij}>_{j=1..N;j\neq i}>_{i=1..N}$ *which is an ancestor of E' such that :*

$$\forall i,j\in[1..N]\ (i\neq j)\ (\ A'_{i,k}=A_{i,k}\ and\ S'_{ij}=(S_{ij})^n\ ;\ n\geq 1\ ).$$

In this case the system presents ineffectif cycles and the corresponding channels are unbounded.

5. Protocol validation algorithm

Based on the theorem, any global behavior that becomes ε after one reduced transformation step is a blocking state. It is a deadlock state if the global behavior that has produced ε in the form $<A_{i,k},(\lambda,\lambda,...,\lambda)>_{i=1..N}$, else it is a blocking reception state for all the processes.
The following algorithm explores all global behaviors reachable from the initial global behavior $<A_{i,0},(\lambda,\lambda,...,\lambda)>_{i=1..N}$ through transformation steps. Each reachable global behavior is analyzed for errors such as deadlock state and blocking reception state. During the validation, we can verify that it has already observed or not. We bound the size of the channels by a constant MAX_CHANNEL, introduced as a parameter of validation, and then the algorithm terminates when all global behaviors with bounded queu have been explored.

Algorithm

Inputs *: N communicating finite state machines and their initial states.*

Outputs: *- The set RS of reachable global behaviors of system ,*
- the set DBS of deadlock states and blocking reception states of all processes, and
- the set OS of global behaviors that the length of one of its sending sequence messages exceds the constant MAX_CHANNEL.

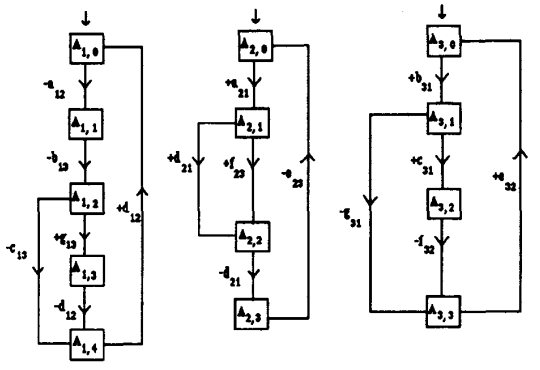
Initially RS has one element $E_0=<A_{i,0},(\lambda,\lambda,...,\lambda)>_{i=1..N}$ *corresponding to the initial global behavior.*

Initialisation :
$RS:=\{E_0\}$; $S:=E_0$; $DBS=OS=\emptyset$;
Step 1:
$C:=S$; $S:=\varepsilon$; *assume that* $C=\sum_i E_i$;
for *every* $E_i\neq\varepsilon$ **do**
let $SD=\sum_j E_j$ *the result of one transformation step of* E_i ;
if $SD=\varepsilon$ **then** $DBS=DBS\cup\{E_i\}$
(E_i *is a deadlock state or blocking reception state)*
else
for *every* $E_j=<A_{i,k},<S_{ij}>_{j=1..N}>_{i=1..N}$ **do if** $\exists i,\ j=1..N\ (i\neq j)\ /\ |S_{ij}|>MAX\_CHANNEL$
then $OS=OS\cup\{E_j\}$
else
if $E_j\notin RS$ (E_j *not already observed in the validation)* **and** E_j *dont verifie the above proposition*
then
begin
$S:=S+E_j$;
$RS:=RS\cup\{E_j\}$
end;
step 2 : **if** $S\neq\varepsilon$ **then return to** *step 1*
else *stop.*

6. An example

Figure 1. shows an example for a protocol consisting of three processes P_1, P_2 and P_3. The square denotes states and arrow denotes transitions with sending and receiving messages. The initial states of processes are $A_{1,0}$, $A_{2,0}$ and $A_{3,0}$. Events are basic units of communication between processes and represented by the symbols associated with the state transition arcs. The event symbol tackes the form ($-a_{ij}$) or ($+a_{ij}$), where ' a_{ij}' represents the type of message exchanged between the processes P_i and P_j. the sign '-' denotes a transmission, the sign '+' denotes a reception, the subscript 'i' denotes the subjectif process identity controling the event and the subscript 'j' denotes the objectif process identity. For example $-a_{ij}$ denotes an emission of message ' a_{ij}' from process P_i to process P_j. When a process is in any state,

the execution of event represents the traverse of the arc labelled by the corresponding event symbol and then it enters in a new state.



Process P_1 Process P_2 Process P_3

Figure 1. An example of three processes

In this example, the process equations of P_1, P_2 and P_3 are :

Process P_1

$A_{1,0}=(-a_{12})A_{1,1}$
$A_{1,1}=(-b_{13})A_{12}$
$A_{1,2}=(+g_{13})A_{1,3} + (-c_{13})A_{1,4}$
$A_{1,3}=(-d_{12})A_{1,4}$
$A_{1,4}=(+d_{12})A_{1,0}$

Process P_2

$A_{2,0}=(+a_{21})A_{2,1}$
$A_{2,1}=(+d_{21})A_{2,2} +(+f_{23})A_{2,3}$
$A_{2,2}=(-d_{21})A_{2,3}$
$A_{2,3}=(-e_{23})A_{2,0}$

Process P_3

$A_{3,0}=(+b_{31})A_{3,1}$
$A_{3,1}=(+c_{31})A_{3,2}+(-g_{31})A_{3,3}$
$A_{3,2}=(-f_{32})A_{3,3}$
$A_{3,3}=(+c_{32})A_{3,0}$.

6.1. Application of one transformation step to the initial global behavior :

From the initial global behavior

$$<<A_{1,0},<\lambda,\lambda>>,<A_{2,0},<\lambda,\lambda>>,<A_{3,0},<\lambda,\lambda>>>$$

with transformation rule for reachability (TP) we obtaind

$$<<(-a_{12})A_{1,1},<\lambda,\lambda>>,<(+a_{21})A_{2,1},<\lambda,\lambda>>,<(+b_{31})A_{3,1},<\lambda,\lambda>>>$$

by applying the algebraic transformation rule we obtaind

$$<<A_{1,1},<(-a_{12}),\lambda>>,<A_{2,1},<(+a_{21}),\lambda>>,<A_{3,1},<(+b_{31}),\lambda>>>$$

Now, by using the transformation rule for reduction (RG1) we have

$$<<A_{1,1},<\lambda,\lambda>>,<A_{2,1},<\lambda,\lambda>>,<A_{3,1},<(+b_{31}),\lambda>>>$$

Finally, the transformation rule for simplification (SR1) gives the following global behavior

$$<<A_{1,1},<\lambda,\lambda>>,<A_{2,1},<\lambda,\lambda>>,<A_{3,0},<\lambda,\lambda>>>.$$

The transformation rule (SR1) for simplification shows that , for the process P_3, the message b_{31} cannot be consummate and the process remains at the same state $A_{3,0}$.

6.2. Application of the algoritm to the protocol of the above example :

Finally, by using the algorithm proposed above, we obtain the following set of not repeated reachable global behaviors :

$\{<<A_{1,0},<\lambda,\lambda>>,<A_{2,0},<\lambda,\lambda>>,<A_{3,0},<\lambda,\lambda>>>$; $<<A_{1,1},<\lambda,\lambda>>,<A_{2,1},<\lambda,\lambda>>,<A_{3,0},<\lambda,\lambda>>>$; $<<A_{1,2},<\lambda,\lambda>>,<A_{2,1},<\lambda,\lambda>>,<A_{3,1},<\lambda,\lambda>>>$; $<<A_{1,3},<\lambda,\lambda>>,<A_{2,1},<\lambda,\lambda>>,<A_{3,3},<\lambda,\lambda>>>$; $<<A_{1,4},<\lambda,\lambda>>,<A_{2,1},<\lambda,\lambda>>,<A_{3,2},<\lambda,\lambda>>>$; $<<A_{1,4},<\lambda,(-c_{13})>>$; $<A_{2,1},<\lambda,\lambda>>,<A_{3,3},<(-g_{31}),\lambda>>>$; $<<A_{1,4},<\lambda,\lambda>>,<A_{2,2},<\lambda,\lambda>>,<A_{3,3},<\lambda,\lambda>>>$; $<<A_{1,0},<\lambda,\lambda>>,<A_{2,3},<\lambda,\lambda>>,<A_{3,3},<\lambda,\lambda>>>$; $<<A_{1,1},<(-a_{12}),\lambda>>,<A_{2,0},<\lambda,\lambda>>,<A_{3,0},<\lambda,\lambda>>>\}$

The global behavior

$<<A_{1,4},<\lambda,(-c_{13})>>,<A_{2,1},<\lambda,\lambda>>,<A_{3,3},<(-g_{31}),\lambda>>>$ is a blocking reception state. This it becomes ε after one transformation step and by transformations rules.

Our method gives 9 unrepeated global states and 2 repeated ones. However, the itho and Ichikawa's method gives 15 unrepeated global states and 2 repeated ones. It

is clear that our method requires a number of global states shorter than the reachability analysis of Itoh and Ichikawa.

Notation used in the Itho and Ichikawa's reachability tree :

- EMPTY : all channels in the corresponding global state are empty,
- (..., $\pm a_{ij}$, ...) : global transition,
- que^{i}_{j} : message queue from P_i to P_j,
- λ_i : waiting event for a process P_j(see [10]).

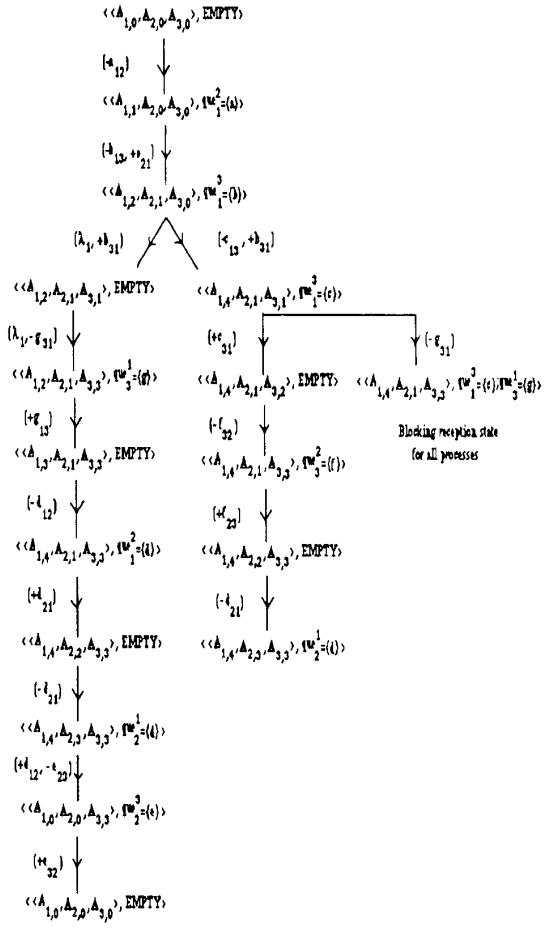


Figure 2. The reduced reachability tree obtained by Itoh and Ichikawa

7. Conclusion

Generalized approach to a multiprocess system of Zaho and Bochmann is introduced. The communicating processes over FIFO channels, defined here, are represented by process equations with states and transitions. The principle goal to validate the communication between processes is the identification of potential errors like deadlock states, blocking reception states and unspecified receptions. Thus, the reachability analysis is defined on the form of algebraic transformation rules applied to the global system behaviors to obtain a set of behaviors instead of reachability tree. An algorithm using those transformation rules is proposed to verify properties of the communication. Then, the number of reachable global behaviors of the system is reduced since the global transitions composed with parallel possible transitions of the processes. The basic idea of this technique consists in carrying out transformations on these equations in order to prove properties as blocked protocols.

REFERENCES

[1] P. M. Merlin, " Specification and validation of protocols", IEEE Trans. Commun., COM-27, 11, pp 1671-1680, Nov. 1979.

[2] J. L. Peterson, " In Petri net theory and the modeling of systems", Prentice Hall, 1981.

[3] R. M. Karp, R. E. Miller, " Parallel program schemata", A mathematical model for parallel computation, conf. Rec. 8th Ann., IEEE Symp. on Switching and Automata theory, Oct. 1976, IEEE, New York, pp 55-61.

[4] P. Zafiropulo et al., " Towards analyzing and synthesizing protocols", IEEE transactions on communication, vol COM 28, pp 651-661, April 1980.

[5] D. Brand, P. Zafiropulo, " On communicating finite state machines", Journal ACM 30, pp 323-342, 1983.

[6] P. M. Merlin, " a methodology for the design and Implementation of communication protocols", IEEE Trans. Commun., COM-24(6), pp 614-621, June 1976.

[7] G. Bochmann, "Finite state descriptions of communication protocols", Computer Networks, pp 361-372, October 1978.

[8] M. Gouda, Y. Yu, " Protocol validation by maximal progressive exploration", IEEE trans. Commun., vol. COM-32 N° 1, Jan. 1984.

[9] P. Zafiropulo, " Protocol validation by duologue-matrix", IEEE Trans. Commun., COM-26, 8, pp. 1187-1194, Aug. 1978.

[10] M. Itoh and H. Ichicawa , " Protocol verification algorithm using reduced reachability analysis", Trans. IECE Japan, vol. E66, N° 2, pp 88-93, Feb. 1983.

[11] J. Zaho, G. Bochmann, " Reduced reachability analysis of communication protocols : A new approach", Publication N°750, Dept. d'IRO, Université de Montréal, 1986.

[12] C. H. West, " General technique for communications protocol validation", IBM J. Res. Devel. N° 22, 4, pp 393-404, July 1978.

[13] " Communication Protocol Modeling." Edited by Carl A. Sunshine, Artech House.