# An Improvement of The Protocol Synthesis Algorithm

5T-2

Zhang Yaoxue Kaoru Takahashi Norio Shiratori Shoichi Noguchi

Research Institute of Electrical Communication, Tohoku University

## 1. Introduction

The production of logical error-free protocols is essential to reliable communications. Protocol synthesis is a technique for designing protocols without logical errors such as unspecified reception, deadlocks, nonexecutable transition and buffer overflow[2]. Many communication protocols can be modeled as two communicating finite-state machines which exchange messages over two one directional, unbounded, FIFO channels. We have developed an interactive protocol synthesis method to synthesis such two machines(two processes) in[2]. In this paper, we discuss an improved synthesis method to reduce the complexity of the computation of [2]'s approach.

## 2. Interactive protocol synthesis method[2]

The approach proposed in [2] is an interactive synthesis algorithm mainly consisting of three production rules and a deadlock avoidance algorithm. The production rules are used to produce a protocol by first constructing its Global State Transition Graph (GSTG). And the deadlock avoidance algorithm is used to help a designer to specify process states. Here, a global state is a composite state of the two process states $q_1$ and $q_2$ and two channel states $c_{12}$ and $c_{21}$. The subscripts used here represent the processes, i.e., $q_i$ is one of the states of process i and $c_{ij}$ is a state of the channel from Process i to Process j. A channel state is a message sequence in the corresponding channel.

The approach automatically prevents a designer from logical errors unspecified reception, nonexecutable transition and deadlocks. In addition, it automatically notifies the designer about the presence of buffer overflow. However, if a protocol to be synthesized is complex, then a large number of global states are needed. This is a problem which limits the application of [2]'s approach.

## 3. Improved synthesis method

We discuss a method to reduce the number of global states needed in [2]'s approach. In the perturbation analysis for protocol validation, an approach for reducing global states has been discussed[1]. We here extend the above approach to apply it to the protocol synthesis method[2]. That is, we extend the production rules given in [2] by constructing transitions of the two processes pairwise, where each pair contains one transition from each of the two processes. The deadlock avoidance algorithm given in [2] has not been modified because it does not effect the space complexity of [2]'s approach. The improved production rules are given as follows. Here, suppose that a progressing state selected from tip global states of GSTG is $S^k=(q_1{}^k,q_2{}^k,c_{12}{}^k,c_{21}{}^k)$ and its immediate successor to be constructed is $S^n=(q_1{}^n,q_2{}^n,c_{12}{}^n,c_{21}{}^n)$, respectively. And suppose that a designer can only specify sending messages at such process states that are only included in the tip global states or progressing state of GSTG.

[Improved production rules]

Rule 1:

If the designer has specified or is to specify sending a message x at $q_i{}^k$, but not at $q_j{}^k$ $(i=1 \text{ or } 2, j=1 \text{ or } 2, j\neq i)$.

Then A) Append transition arc $A=(a_1,a_2)$ to $S^k$. Here, $a_i=-x$, $a_j=+y$. Where, + and - represent message transmission and reception, respectively. And y is the first message of $c_{ij}{}^k$. If $c_{ij}{}^k$ is a empty sequence, then $y=x$.

B) B-1) Set $c_{ji}{}^n=c_{ji}{}^k$.

B-2) Make new channel states $c_{ij}{}^n$ by satisfying the relation: $y\cdot c_{ij}{}^n=c_{ij}{}^k\cdot x$ ("·":concatenation symbol).

B-3) If $q_i{}^k$, $q_i{}^n$ and the arc labeled -x linking them, or $q_j{}^k$, $q_j{}^n$ and the arc labeled +y linking them have not been produced before,then ask the designer to specify $q_i{}^n$ or $q_j{}^n$; else copy $q_i{}^n$ or $q_j{}^n$.

B-4) Append $S^n=(q_1{}^n,q_2{}^n,c_{12}{}^n,c_{21}{}^n)$ to arc A.

C) Produce $q_i{}^k$, $q_i{}^n$ and the arc labeled -x linking them in Process i; and $q_j{}^k$, $q_j{}^n$ and the arc labeled +y linking them in Process j, respectively.

Rule 2:

If the designer has specified or is to specify sending a message x at $q_i{}^k$ and a message y at $q_j{}^k$ simultaneously.

Then A) Append transition arc $A=(a_1,a_2)$ to $S^k$. Here, $a_i=-x$ and $a_j=-y$.

B) B-1) Make new channel states $c_{ij}{}^n$ and $c_{ji}{}^n$ by satisfying the relations: $c_{ij}{}^n=c_{ij}{}^k\cdot x$ and $c_{ji}{}^n=c_{ji}{}^k\cdot y$.

B-2) If $q_i{}^k$, $q_i{}^n$ and the arc labeled -x linking them, or $q_j{}^k$, $q_j{}^n$ and the arc labeled -y linking them has not been produced before, then ask the designer to specify $q_i{}^n$ or $q_j{}^n$;

else copy $q_i^n$ or $q_j^n$.

B-4) Append $S^n = (q_1^n, q_2^n, c_{12}^n, c_{21}^n)$ to arc A.

C) produce $q_i^k$, $q_i^n$ and the arc labeled -x linking them in Process i; and $q_j^k, q_j^n$ and the arc labeled -y linking them in Process j, respectively.

Rule 3:

If $c_{ij}^k$ is not a empty sequence.

Then A) Append transition arc $A = (a_1, a_2)$ to $S^k$. Here, $a_i = +x$ and $a_j = +y$ (x is the first message of $c_{ji}^k$ and y is the first message of $c_{ij}^k$).

B) B-1) Make new channel states $c_{ij}^n$ and $c_{ji}^n$ by satisfying the relations: $y \cdot c_{ij}^n = c_{ij}^k$ and $x \cdot c_{ji}^n = c_{ji}^k$.

B-2) If $q_i^k$, $q_i^n$ and the arc labeled +x linking them, or $q_j^k$, $q_j^n$ and the arc labeled +y linking them has not been produced before,
then ask the designer to specify $q_i^n$ or $q_j^n$;
else copy $q_i^n$ or $q_j^n$.

B-4) Append $S^n = (q_1^n, q_2^n, c_{12}^n, c_{21}^n)$ to arc A.

C) produce $q_i^k$, $q_i^n$ and the arc labeled +x linking them in Process i; and $q_j^k, q_j^n$ and the arc labeled +y linking them in Process j, respectively. □

By using above production rules instead of the production rules of the synthesis algorithm given in [2], we obtain the improved synthesis algorithm.

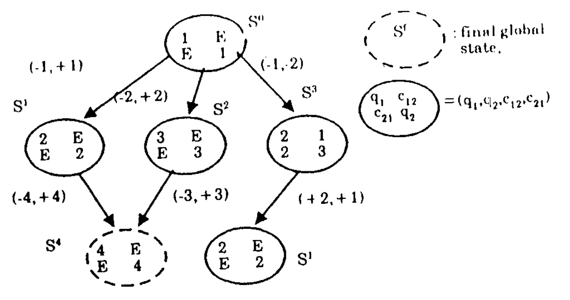


Fig.2 IGSTG of call set up protocol of X.25 constructed by the improved approach

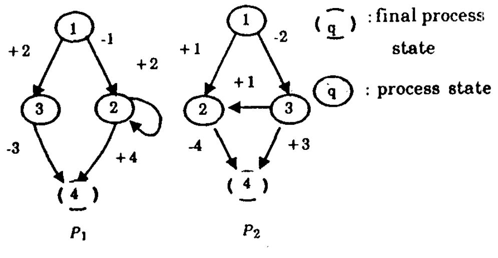


Fig.3 Call set up protocol of X.25 synthesized by two approaches

Theorem:

*Let X be the number of global states needed by using [2]'s approach for synthesizing a protocol. Let Y be the number of global states needed by using the improved method given above for synthesizing the same protocol. We have*

$$Y \leq X/2 + 1. \quad \square$$

## 4. Example

Fig.1 shows GSTG of call set up protocol of X.25, constructed by [2]'s synthesis algorithm. The corresponding protocol decomposed from GSTG is shown in Fig.3. Fig.2 shows IGSTG which is constructed by the proposed improved method. Obviously, the decomposition of IGSTG leads to Fig. 3 and IGSTG is a subgraph of GSTG. Moreover, there are 15 nodes in GSTG shown in Fig.1 and only 6 nodes in IGSTG shown in Fig.2. Hence, the number of global states reduced by the improved method in this example is 9.

## 5. Conclusions

An improved synthesis method for [2]'s approach has been discussed. The number of global states produced by the improved method is much fewer than the one produced by the method[2]. Consequently, efficiency for protocol synthesis goes up. However, since the reachable global states have not been produced completely, this method is not able to notisfy the designer about the presence of buffer overflow.

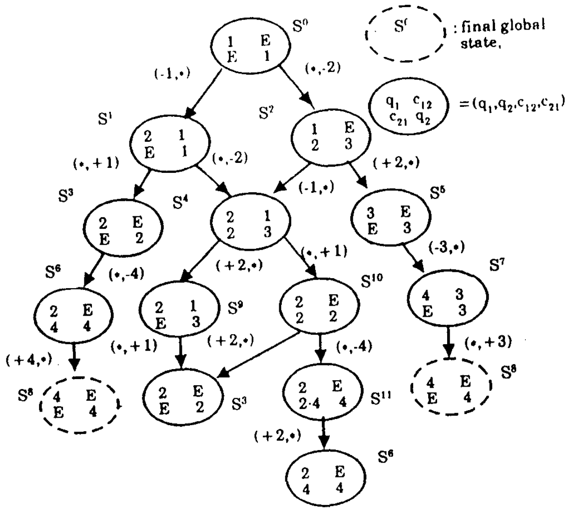


Fig.1 GSTG of call set up protocol of X.25 constructed by [2]'s approach

*References*


1. J.Rubin and C.H.West:"An improved protocol validation technique", Computer Networks, vol. 6,pp, 66-73, Apr.1982.
2. Zhang et al,"A protocol synthesis algorithm using global state transition graph",Paper of Technical Group, SE86-32, IECE of Japan, 1986.