Protocol validation: a parallel technique to reduce the reachability tree

Abderrahim Benslimane

Laboratoire d’Informatique, UFR des Sciences et Technique
16 route de Gray, 25030 Besançon, France
E-mail: benslim@comte.univ-fcomte.fr

Abstract. Protocol validation detects the existence of logic errors in protocol design specifications. The commonly used technique to validate protocols is reachability analysis. However, it has a major drawback which is the combinatorial explosion of the state space. Several methods have been proposed to decrease the number of states. This paper presents a new method of parallel execution of actions, allowing reduction of reachable state space and detection of two types of logic errors: deadlocks and blocking reception states.

1 Introduction

Recently the growth of distributed computing and computer networks are increasing the complexity of communication protocols. The need for specifying and verifying formally the communication protocols is growing rapidly. Several models have been successfully used for this purpose. They are the "state transition" model, the "programming language" model, and the combination of the above two models. General surveys of these three approaches can be found in [2, 4, 15]. The main advantage of state transition models is that the validation of communication protocols can be easily automated. 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 language models are free from state space explosion problem, but they imply difficulty in automating verification process. They require special knowledge in the area of program proving.

In the state transition model, several approaches have been proposed. Examples most studied are Petri Nets and Finite state Machines. In the formalism of Petri nets [12, 13], each process is represented in the form of graph of round-shaped places, associated to actions, and bar-shaped transitions associated to events and conditions. Some properties are decidable in this model. In this case, Karp and Miller [10] have proposed a solution for decidability problem of a boundedness. In the models using communicating finite state machines (CFSM's, for short) [3, 4, 7, 18], each process is represented in the form of state transition machines 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 [2, 5, 11, 18] according to certain
progress rules. However, in practice the termination of the construction of the reachability tree of protocols specified in terms of finite state machines communicating over FIFO channels can be infinite.

The focus of this paper is the finite state machines model. Several techniques of reachability analysis using this model have been proposed. West [16] introduced perturbation technique. It is a reachability analysis (classical method) which explores all possible interactions of the processes by exhaustively generating all global states reachable from a given initial global state. It constructs a reachability tree. In the same way, Zafiropulo [17] introduced a validating method using a duologue matrix analysis for protocol consisting of only two processes. This technique is limited to protocols that must revert to an initial or quiescent state after a number of interaction steps. Some reduced reachability techniques have been introduced in the literature [1, 5, 8, 9, 14, 19, 21, 22]. For more detail, the reader can consult our report [20].

In this paper, we propose a new method of parallel execution of actions, allowing reduction of the reachable states space. This technique is based on suppression of redundant sequences. To create the successor of a global state, instead of performing at every step one action of one process, we propose to perform in parallel one action of every process. Only one step is necessary for verification. Logical errors, owed to conception, which can be detected are deadlocks and blocking reception states. A proof of correctness is given. Complexity and comparison of this method with others are studied. The number of states is divided by K!, compared with the classical method, where K is the number of processes.

The rest of the paper is organized as follows:
- in section 2, some definitions of the model used are given,
- the reachability analysis is presented in the section 3,
- the proposed method of validation allowing the creation of the reduced reachability tree is presented in section 4,
- in section 5, we give the algorithm to validate systems of CFSM's accompanied with two illustrating examples and calculation of number of nodes created,
- finally, concluding remarks are given in section 6.

2 Model

We consider a system $S$ of CFSM's consisting of a set of $K$ finite state processes that send messages to, and receive messages from, each other. The communication medium from one process to another is assumed to be error-free. A send event causes a message to be enqueued in a FIFO channel, while a receive event dequeues a message from a channel. We represent each process $P_i$, $i=1..K$, by a CFSM $A_i$.

A communicating machine is a directed labeled graph with two types of edges called sending and receiving edges.

**Definition 1**

Now, we define formally the model of CFSM $A_i$ as a four-tuple $<Q_i, A_i, q_{i0}, \delta_i>$ where
- $Q_i$ denotes the finite set of states of $A_i$.
- $A_i$ denotes the finite set of possible interactions of $A_i$. It is the set of actions written in the form `$a_{ij}$' that can be sent from $A_i$ to an other CFSM $A_j$ and those written in the form `$a_{ji}$' that can be received by $A_i$ from another CFSM $A_j$.
- $q_{i0}$ is the initial state of $A_i$. 