Compositional Reasoning in Model Checking *

Sergey Berezin[1] Sérgio Campos[2] Edmund M. Clarke[1]

[1] Carnegie Mellon University — USA
[2] Universidade Federal de Minas Gerais — Brasil

Abstract. The main problem in model checking that prevents it from being used for verification of large systems is the *state explosion problem*. This problem often arises from combining parallel processes together. Many techniques have been proposed to overcome this difficulty and, thus, increase the size of the systems that model checkers can handle. We describe several *compositional model checking* techniques used in practice and show a few examples demonstrating their performance.

1 Introduction

Symbolic model checking is a very successful method for verifying complex finite-state reactive systems [7]. It models a computer system as a state-transition graph. Efficient algorithms are used to traverse this graph and determine whether various properties are satisfied by the model. By using BDDs [5] it is possible to verify extremely large systems having as many as 10^{120} states. Several systems of industrial complexity have been verified using this technique. These systems include parts of the Futurebus+ standard [12, 19], the PCI local bus [10, 20], a robotics systems [8] and an aircraft controller [9].

In spite of such success, symbolic model checking has its limitations. In some cases the BDD representation can be exponential in the size of system description. This behavior is called the *state explosion problem*. The primary cause of this problem is *parallel composition* of interacting processes. The problem occurs because the number of states in the global model is exponential in the number of component processes. Explicit state verifiers suffer from the state explosion problem more severely than symbolic verifiers. However, the problem afflicts symbolic verification systems as well, preventing them from being applied to larger and more complex examples.

The state explosion can be alleviated using special techniques such as *compositional reasoning*. This method verifies each component of the system in isolation and allows global properties to be inferred about the entire system. Efficient

* This research is sponsored by the the Semiconductor Research Corporation (SRC) under Contract No. 97-DJ-294, the National Science Foundation (NSF) under Grant No. CCR-9505472, and the Defense Advanced Research Projects Agency (DARPA) under Contract No. DABT63-96-C-0071. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of SRC, NSF, DARPA, or the United States Government.

W.-P. de Roever, H. Langmaack, and A. Pnueli (Eds.): COMPOS'97, LNCS 1536, pp. 81-102, 1998.
Springer-Verlag Berlin Heidelberg 1998

algorithms for compositional verification can extend the applicability of formal verification methods to much larger and more interesting examples. In this paper we describe several approaches to compositional reasoning. Some are automatic and are almost completely transparent to the user. Others require more user intervention but can achieve better results. Each is well suited for some applications while not so efficient for others.

For example, *partitioned transition relations* [6] and *lazy parallel composition* [11, 27] are automatic and, therefore, preferred in cases where user intervention is not desired (for example, when the user is not an expert). These techniques provide a way to compute the set of successors (or predecessors) of a state set without constructing the transition relation of the global system. Both use the transition relations of each component separately during traversal of the state graph. The individual results are combined later to give the set of states in the global graph that corresponds to the result of the operation being performed.

Another automatic technique is based on the use of *interface processes*. This technique attempts to minimize the global state transition graph by focusing on the communication among the component processes. The method considers the set of variables used in the interface between two components and minimizes the system by eliminating events that do not relate to the communication variables. In this way, properties that refer to the interface variables are preserved, but the model becomes smaller.

Assume-guarantee reasoning [17] is a manual technique that verifies each component separately. The behavior of each component depends on the behavior of the rest of the system, i.e., its environment. Because of this, the user must specify properties that the environment has to satisfy in order to guarantee the correctness of the component. These properties are *assumed*. If these assumptions are satisfied, the component will satisfy other properties, called *guarantees*. By combining the set of assume/guarantee properties in an appropriate way, it is possible to demonstrate the correctness of the entire system without constructing the global state graph.

All of these methods have been used to verify realistic systems. This shows that compositional reasoning is an effective method for increasing the applicability of model checking tools. Furthermore, it is a necessity for verification of many complex industrial systems.

The remainder of this paper is organized as follows: Section 2 introduces the formal model that we use for finite-state systems and the kinds of parallel composition we consider. Section 3 describes partitioned transition relations, and Section 4 discusses lazy parallel composition. Interface processes and assume-guarantee reasoning are described in Sections 5 and 6, respectively. Finally, the paper concludes in Section 7 with a summary and some directions for future research.