Symmetry for the Analysis of Dynamic Systems

Zarrin Langari and Richard Trefler*

David R. Cheriton School of Computer Science
University of Waterloo, ON, Canada
{zlangari,trefler}@cs.uwaterloo.ca

Abstract. Graph Transformation Systems (GTSs) provide visual and explicit semantics for dynamically evolving multi-process systems such as network programs and communication protocols. Existing symmetry reduction techniques that generate a reduced, bisimilar model for alleviating state explosion in model checking are not applicable to dynamic models such as those given by GTSs. We develop symmetry reduction techniques applicable to evolving GTS models and the programs that generate them. We also provide an on-the-fly algorithm for generating a symmetry-reduced quotient model directly from a set of graph transformation rules. The generated quotient model is GTS-bisimilar to the model under verification and may be exponentially smaller than that model. Thus, analysis of the system model can be performed by checking the smaller GTS-bisimilar model.

1 Introduction

Model checking is used to analyze finite state program models. Many of these models are composed of similar components. In practice, the number of components in these models may be dynamically changing within a given upper bound. For instance, for many communication protocols, the given bound arises naturally due to inherent limitations on system size. Examples of dynamic systems composed of similar components include communication protocols such as IP-telephony protocols where telephony features are dynamically assembled in a call over the Internet [15], network programs with a variable number of clients, and object-oriented systems such as dynamic heap allocation programs [12].

Due to the use of similar components, symmetry is often a feature of the above system models that can be exploited to reduce the state space of a model under verification. Unfortunately, existing symmetry-reduction methods [13,7,9] are not applicable to dynamic systems. In addition, they may offer only limited reduction to system models that are not fully symmetric. Full symmetry causes the system model to be invariant under arbitrary rearranging of the components, resulting in an exponential reduction by defining an equivalence relation on symmetric states of the system model. An example of a fully symmetric system model

* The authors’ research is supported in part by the NSERC of Canada. Zarrin Langari is currently in McMaster University, Hamilton, ON, Canada; and is supported in part by Mathematics of Information Technology and Complex Systems (MITACS).
Symmetry for the Analysis of Dynamic Systems

We propose a symmetry reduction method for analyzing visual models of dynamically evolving systems. Our symmetry reduction approach is applicable to non-fully symmetric system architectures such as hypercube, ring, and torus (used in metropolitan area networks that need high scalability) used for modelling next-generation communication and hardware protocols.

Motivation: Graphs provide visual and explicit operational semantics for presenting states and demonstrating structural symmetries of a system. GTs, which use this graph-based semantics, are straightforward formalisms that offer several key advantages over naive methods in modelling the dynamic evolution of multi-process systems. Recently, the GT formalism has been used to perform reasoning, including verification and error detection, on multi-component, reactive systems. Our motivation is to exploit the advantages that graph-based models provide for the modelling and analysis of dynamically evolving systems.

When systems are composed of several similar components, it is often convenient to identify the various components by their process indices. In a Kripke model of these systems, a state consists of the values of all global variables and the local states of each process. For example, consider a $3 \times 3$ toroidal mesh network of processes, as in Figure 1-a. A toroidal mesh is a grid network with wrap-around links, where each process can communicate to two other processes. A shared token is used to show the access of processes to some resource. In this example, the local state $T_{23}^+$ describes that the process in row 2, column 3 possesses a token (denoted by a plus sign) and is trying to access a shared resource (denoted by $T$), and the other processes are in their non-trying modes (denoted by $N$). Symmetries in these models are then represented as permutations of the process indices. Symmetry-reduction methods use the index permutation to build a symmetry-reduced quotient model that is equivalent, up to permutation, to the behaviour of the original model.

In Kripke models, the labelling of each state does not explicitly show the architecture of the system. On the contrary, in a GT model of the system, each global state is represented by a graph that explicitly provides the architecture in

---

**Fig. 1.** a) A non-fully symmetric $3 \times 3$ and b) a fully symmetric $2 \times 2$ toroidal mesh with four components is illustrated in Figure 1-b.