A Technique to Alleviate the State Space Explosion for Eventual Model Checking, Its Support Tool and Case Studies

2110429 Moe Nandi Aung

Nowadays, software and hardware play a crucial role in many areas of our daily lives, such as education, finance, healthcare, communication, and more. In these applications, it is important to prevent failures because they are not allowed or acceptable. Model checking, as a formal verification technique, holds great promise due to its ability to automate the verification process once concise formal models are constructed. It has found extensive application in various industries, particularly in the realm of hardware verification. However, the state space explosion problem which refers to the exponential growth in the number of states that must be considered during the verification process, rendering it computationally demanding or even infeasible, still remains in it.

We address the state space explosion in model checking, which stands as one of the most significant hurdles in this field. To mitigate this problem, we propose a divide & conquer approach to eventual model checking (DCA2EMC). As the name implies, our approach primarily focuses on handling eventual properties, which are expressed in linear Temporal Logic (LTL) as $\Diamond\varphi$, where φ represents a state proposition. Eventual properties informally say that something will eventually happen. These can be used to express many important software requirements of a system. For example, halting or termination is one important software requirement and this can be formalized by utilizing the eventual properties.

Our divide & conquer approach involves dividing the original eventual model checking problem into multiple smaller model checking problems, each of which is tackled. We have proved a theorem that demonstrates the equivalence of the multiple smaller model checking problems to the original eventual model checking problem. We have also constructed an algorithm based on the theorem in order to build a support tool capable of performing verification of eventual properties in model checking. Additionally, we have developed a tool that supports the proposed approach. Our support tool is developed in Maude, a high-level language and high-performance system that supports both equational and rewriting logic computation. We have conducted some case studies and experiments. Our approach provides a promising solution for tackling the state space explosion in model checking for eventual properties.

Keywords: a divide and conquer approach, model checking, eventual properties, the state space explosion, Maude.