Chapter 2

Formal Verification Methods

ABSTRACT

This chapter provides a brief introduction to the domain of formal methods (Boca, Bowen, & Siddiqi, 2009) and the most commonly used verification methods (i.e., theorem proving [Harrison, 2009] and model checking [Baier & Katoen, 2008]). Due to their inherent precision, formal verification methods are increasingly being used in modeling and verifying safety and financial-critical systems these days.
2.1 INTRODUCTION

The history of formal methods dates back to Knuth and Dijkstra as both of them advocated the topic. Formal verification methods started to be investigated as CAD tools in the 1970s for software verification. However, the interest was marred by the fact that software bugs can be easily fixed by releasing a software patch and thus the added reliability of software is not worth the rigorous exercise of formal verification. There was some research activity related to the formal verification of security systems funded by the US National Security Agency in the 1980s but the real catalyst for the active research interest in formal verification was their usage in verifying digital hardware systems in late 1980s. This is mainly because hardware descriptions are often more regular and hierarchical than software ones, hardware primitives are less obscure than the ones used in software and the cost of an uncaught design bug in hardware is much more profound than software since the hardware silicon chip once fabricated cannot be fixed by releasing a patch but instead has to be re-designed and re-fabricated, which costs considerable amount of time and money. The Intel FDIV bug in 1994 (Pentium FDIV Bug, 2015) further enhanced the interest in formal hardware verification and the industry started to adopt formal hardware verification tools in their design flows in late 1990s. With the success of formal verification in hardware and due to some interesting developments in the underlying technologies, it started to be used again for software, transportation and security system analysis domains. Moreover, formal verification has recently been explored for accurately analysis of continuous physical systems, such as control systems, robotics and analog circuits. The future of formal methods seems to be quite promising and besides academia, industry giants, like Intel and Microsoft, are also actively participating in the research related to both foundational and practical aspects.

The added benefits of formal verification methods (Hall, 2007) come mainly at the cost of extreme rigor. Generally speaking, the expressiveness of a formal verification method is in direct proportion with the amount of required user intervention. Thus, formal verification of complex systems is more challenging and time consuming. Therefore, the general trend is to use a lightweight approach, i.e., use traditional verification methods, like simulation or testing, where accuracy of the analysis is not a big concern while using formal verification methods for the critical sections of the systems. On similar lines, hybrid formal verification methods are also being developed which allow us to partition the overall system model based on its complexity levels and thus facilitate using automatic formal verification methods for the rather simpler sections of the system while using the interactive methods with the complex sections.
Strategic Data-Based Wisdom: Applying Aristotle’s Theory of Action to the DIKW Model
Cristiano Cippe and Maria Cinque (2015). Strategic Data-Based Wisdom in the Big Data Era (pp. 172-196).
www.igi-global.com/chapter/strategic-data-based-wisdom/125052?camid=4v1a