基于克雷格插值的反例理解方法

黄宏涛* 黄少滨† 张涛‡ 陈志远§

哈尔滨工程大学计算机科学与技术学院，黑龙江 哈尔滨 150001

摘要：随着计算机软硬件系统日益复杂，如何保证其正确性和可靠性成为日益紧迫的问题。在为此提出的诸多理论和方法中，模型检测以其简洁明了和自动化程度高而引人注目，模型检测的研究大致涵盖以下内容：模态/时序逻辑、模型检测算法及其时空效率（特别是空间效率）的改进以及支撑工具的研制。这几个方面之间有着密切的内在联系，不同模态/时序逻辑的模型检测算法的复杂性不一样，优化算法往往是针对某些特定类型的逻辑公式，本文将就这几个方面分别加以阐述，最后介绍该领域的新进展。

关键词：很关键；很关键；非常关键

Understanding counterexamples using Craig Interpolation

Hongtao Huang ShaobinHuang Tao Zhang Zhiyuan Chen

College of Computer Science and Technology
Harbin Engineering University, Heilongjiang Harbin 150001

Abstract: Model checking is an automatic technique for verifying finite-state reactive systems, such as sequential circuit designs and communication protocols. Specifications are expressed in temporal logic, and the reactive system is modeled as a state-transition graph. An efficient search procedure is used to determine whether or not the state-transition graph satisfies the specifications. We describe the basic model checking algorithm and show how it can be used with binary decision diagrams to verify properties of large state-transition graphs. We illustrate the power of model checking to find subtle errors by verifying part of the Contingency Guidance Requirements for the Space Shuttle.

Keywords: Key; Key; the Key

1 引言

模型检测技术 [1, 2, 3] 的主优势之一是它能够在证明模型不满足给定规范的同时自动给出反例（一条违反给定规范的迹）。然而，反例仅仅反映了模型缺陷的症状，验证者仍然需要花费大量的时间和精力理解反例，以确定造成模型缺陷的原因。在软硬件系统规模快速膨胀的背景下，各种模型检测器能够处理的状态空间规模也有了长足提升。在实际应用中，模型检测器给出的反例也越来越长。从反例中提取迹失效的原因已经被证明为 NP 完全问题 [4, 5, 6]。因此，寻找高效的算法进行反例自动理解将成为模型检测技术将要面临的一个难题。

2 相关工作

近年来，如何从反例中发现模型缺陷的源头已经开始引起研究人员的注意。一些工作已经涉及到如何寻找模型失效的根本原因，并且提出了自动提取失效原因以简化程序调试工作的方法。

*通讯作者：horntau@gmail.com
†huangshaobin@hrbeu.edu.cn
‡zhangtaohrbeu@163.com
§chenzhiyuan@hrbeu.edu.cn

3 参考文献

参考文献

[1] Clarke E, Emerson E. Design and synthesis of synchronization skeletons using branching time temporal logic[J]. Logics of Programs, 1982: 52-71.

[2] Queille J, Sifakis J. Specification and verification of concurrent systems in CESAR[C]//International Symposium on Programming. 1982: 337-351.

[3] G.O. Clarke, E.M.Emerson. Model Checking[M]. Cambridge: MIT Press, 1999.

[4] Ben-David S. Applications of Description Logic and Causality in Model Checking[D]. University of Waterloo, 2009.

[5] Beer I, Ben-David S, Chockler H, et al. Explaining counterexamples using causality[C]//Computer Aided Verification. 2009: 94-108.

[6] Jalbert N, Sen K. A trace simplification technique for effective debugging of concurrent programs[C]//Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering. 2010: 57-66.