形式化方法类课程设置及教学内容探索

华保健 *　　樊淇梁　　潘志中
中国科学技术大学软件学院
bjhua@ustc.edu.cn*　　sa613162@mail.ustc.edu.cn　　sg513127@mail.ustc.edu.cn

摘要—分析目前计算机和软件工程专业形式化方法类课程的现状，并根据软件工程发展形势的最新要求，以及课程体系与 **ACM/IEEE CS2013** 计算机科学课程体系规范中的知识主体的对应关系，指出现行课程体系的可改进之处，阐述重新构建形式化方法类课程体系的可行途径。

关键词—形式化方法；编程语言；软件可靠性
中图法分类号—G642

I. 引言

随着互联网、物联网、人工智能、区块链的高速发展，计算机系统及软件已深植各行业，但是随之而来的软件缺陷和软件脆弱性问题也日趋严重。以区块链行业为例，仅在 2020 年，软件安全问题造成的经济损失就超过了 121 亿美元 [1]。形式化方法作为计算机科学的一个重要研究方向，由于其在保证计算机软硬件可靠性和安全性方面发挥的重要作用，越来越受到学术界和工业界的重视，目前在该研究方向上已经产生了许多重要研究成果，并已产生了 15 位图灵奖获得者 [2]。随着形式化方法在计算机软硬件可靠性与安全性方面日益发挥重要作用，业界亟需更多的掌握形式化方法理论和相关技术的专业人才。这也给相关的课程设置和人才培养体系，提出了更高的挑战；

针对这一挑战，目前国内高校已经开设了形式化方法类课程，如北京大学的“形式化方法”、清华大学的“形式语言与自动机”、中国科技大学的“形式化方法”等；这些课程的开设，对于完善课程体系、提高课程建设质量、提高人才培养的质量，都起到了积极作用。但是，现有的相关课程还存在以下主要问题：

Q1. 课程理论性过强，学生对于课程内容中的知识点吸收与消化相对困难。

Q2. 课程内容设置中的理论和实践的结合度有待提高，学生不了解理论的使用场景，导致学习兴趣下降。

通讯作者：华保健（bjhua@ustc.edu.cn）

Q3. 课程实践作业设计不够科学，学习难度较高。

针对以上问题，我们结合当前业界计算机软硬件可靠性和安全性保障新趋势，基于 ACM/IEEE CS2013 计算机科学课程体系规范中的知识主体 [3]，结合对形式化方法课程开设的总体经验，重点讨论介绍在计算机和软件工程专业研究生层次，开设形式化方法类课程的教学内容选取与探索。

II. 软件工程的最新趋势和需求

随着计算机软件的广泛应用，软件质量问题越加突显 [4]，人们对软件可靠性提出了更高的要求。形式化方法作为一种保障计算软硬件可靠性与安全性的重要技术，国内外学术界和工业界越来越重视，研究形式化方法技术在软件质量和软件可靠性中的应用也越来越广泛。

在学术界，Oheimb [5] 介绍了形式化方法技术在工业产品、特别是在飞机电子分发软件（electronic distribution system，EDS）应用上的验证；Davidson 等 [6] 使用符号执行的技术，查找嵌入式软件中的漏洞；Matsuo [7] 把形式化方法技术运用于区块链技术，对区块链进行安全评估与验证；Yang 等 [8] 使用符号传播技术，来增强对深层神经网络的鲁棒性验证；钱等 [9] 提出了一种基于形式化技术的功能验证方法，来保证核电安全级仪控系统稳定性、可靠性和安全性。

在工业界，许多企业已成立专门的实验室，专注于形式化方法理论在实际中的应用，并已初具成果。微软开发了定理证明器 Z3 [10] 、形式化验证工具 SLAM [11] 等，并把这些形式化验证工具应用于许多产品的研发；Facebook 开发了 Infer [12] 工具，已全面应用于 Messenger [13]、Instagram [14] 等知名手机应用，对代码安全性进行检测；国内的华为公司，在 2019 年的华为开发者大会上宣布，已把形式化方法应用于鸿蒙 OS，

表 I: CS 2013 课程体系规范中与形式化方法课程相关的知识主体

课程	知识领域	核心 1 级	核心 2 级	选修
形式化方法	DS/集合、关系与函数	涉及		
	DS/基础逻辑	涉及		
	DS/证明方法	涉及	涉及	
	PL/形式语义			涉及
	SE/形式化方法			涉及
	AL/高级计算复杂度			涉及
	AL/基础自动机的可计算性及复杂度		涉及	

显著的提升了“微内核技术用于可信执行环境”内核的安全。

随着国内外学术界和工业界对于形式化方法理论研究的逐步深入，以及形式化方法在实际工程实践中的推广应用，形式化方法学科的重要性日益凸显。但是，形式化方法人才培养相对滞后，相关专业设置以及课程体系不能满足最新的教学要求，迫切需要建设并完善一门课程体系先进、内容丰富、实践针对性强的形式化方法课程。

III. 形式化方法课程体系

A. *ACM/IEEE CS2013* 知识体系

ACM/IEEE CS2013 知识主体由 18 个知识领域组成，其中与形式化方法类课程有关的知识领域有四个，分别是：离散结构 DS、程序设计语言 PL、软件工程 SE、算法与复杂度 AL。课程知识点分为“核心 1 级”、“核心 2 级”和“选修”共三类，形式化方法课程在课程设计时涉及了以上知识点。

表 I列出了形式化方法课程与 DS、PL、SE 和 AL 4 个知识领域、知识主体之间的关系。从表中可见，形式化方法课程与相关的其它课程知识体系，都有着密切联系：包括核心 1 级知识点中的集合、关系与函数、基础逻辑、证明方法等；核心 2 级知识点中的证明方法、基础自动机的可计算性及复杂度等；以及选修知识点中的形式语义、形式化方法和高级计算复杂度等。

ACM/IEEE CS2013 中有 4 个知识主体涉及了形式化方法及相关知识，充分说明了形式化方法课程在计算机及相关专业开设的重要性和必要性。

B. 课程设置建议

形式化方法作为计算机科学的传统研究方向，具有较长的历史。1949 年，Turing [15] 发表的论文提到了程序正确性问题。1962 年，McCarthy [16] 在 IFIP 上的演讲论述了形式语言和程序设计理论的重要性。1968 年，在德国召开的 NATO 软件工程会议，提出建立软件开发和生产的数学基础。之后，形式化方法一词开始被广泛使用，形式化方法的定义也趋于明确，即一种基于严格数学基础，对计算机软硬件进行描述、开发和验证的技术。发展至今，形式化方法已与人工智能、量子计算、生物计算等领域融合，形成一门高度抽象、深度融合的交叉学科，这对于形式化方法课程的教学也提出了更高的要求。

我院于 2010 年起开设形式化方法课程，期间对课程内容与实践进行了多次迭代和更新，目前在课程内容、授课方式和实践设计上相对比较合理。课程已完整实施多轮次，在这过程中已获得一些成效和经验，以期更好地解决前述 Q1~Q3 问题。

对于问题 Q1，为了解决学生对于形式化方法理论知识理解与掌握相对困难的问题，建议从授课内容、授课形式上着手进行解决。对于授课内容，建议把经典理论与现代最新研究进展相结合，与时俱进，更好体现课程内容先进性与科学性。对于授课形式，建议采用理论与实际案例相结合的方式，通过实际案例讲深讲透知识点，让学生充分理解并吸收所教授的内容。

对于问题 Q2，为了让学生认识形式化方法知识在实际中的应用以及激发学生的学习兴趣，建议从实践作业和授课形式入手。第一，建议在实践作业中设计实际研究或工程中遇到的经典问题，引导学生采用形式化方法的理论知识去解决问题，使学生充分了解形式化方法在软件可靠性中的重要作用。同时，建议在实践作业中加入使用形式化方法理论构建的现代化工具介绍与使用，如辅助定理证明工具、SAT/SMT 工具等，使学生认识到形式化方法现代工具的有效性。第二，在授课的过程中，可加入对最新研究前沿和研究进展的介绍，尤其是前沿工具开发和系统应用的介绍，这可以帮助学生

表 II: 形式化方法课程大纲

章节名称	章节内容
基础知识	集合、关系与映射、上下文无关文法、基于结构的归纳法、文法的实现
命题逻辑	语法、自然演绎系统、构造逻辑、语义系统、可靠性与完备性、可判断性
布尔可满足性	合取范式、解析与传播、DPLL 算法
谓词逻辑	语法、自然演绎系统、构造逻辑、语义系统、可靠性与完备性、可判断性
等式与未解释函数理论	可满足性模理论、等式理论、并查集与等价类、未解释函数
线性算术	语法、Fourier-Motzkin 消元法、单纯形法、分支定界法
数据结构理论	比特向量、数组、指针、字符串
理论组合	Nelson-Oppen、理论凸性、DPLL(T) 算法
符号执行	机器抽象模型、操作语义、简单命令式语言、路径条件、混合执行等
程序验证	霍尔三元、最弱前条件、验证条件等
程序合成	基于语法的合成、公理化合成等

进一步了解理论的使用场景。

对于问题 Q3，为了减轻学生学习的难度，建议采取两方面措施。第一，对于实践作业内容，可采取循序渐进、由易到难的题型设计；对于涉及程序设计的问题，建议采用事先设计代码框架，让学生阅读部分代码来完成空缺代码的填充，并且提供充分的测试例，学生可以根据测试例快速判断编写代码的正确性。第二，对于课程实践作业中涉及到软件或工具，建议从安装过程就开始详细介绍，最好提供详细的安装步骤和使用手册，尽可能地降低学生对于工具的上手的难度；设计引导性题型，引导学生分步熟悉软件或工具的具体功能。

目前的不同院校的教学实践中，形式化方法一般有两种设置模式：作为本科高年级的选修课或作为研究生阶段软件与理论或信息安全相关专业的必修课。尽管在本科阶段开设关于形式化方法的先导性介绍课程有助于学生尽早理解和掌握该领域的核心知识；但是，基于笔者的教学实践，建议在本科阶段以教授基础知识为主，把理论性强的课程内容以及实践环节放在研究生阶段的课程中讲授。形式化方法是一门理论性较强的学科，部分概念过于抽象，在本科阶段教授这部分知识对学生的学习挑战较大。而在研究生阶段开设该课程，学生有较完善的知识储备，对于课程内容的理解和接受会更容易。

IV. 形式化方法课程教学探索

形式化方法课程自 2010 年来我们已授课多轮次，并在 2019 年根据形式化方法的最新发展，以及为了使学生能够更好的认识形式化方法课程的重要性，我们从课程内容、授课方式和实践作业做了全面的更新，课程的主页详见 http://staff.ustc.edu.cn/~bjhua/courses/theory/2021/schedule.html。

A. 教学内容

1) 理论内容: 形式化方法课程总共设置为 60 个学时，课程大纲涵盖数理逻辑基础、理论及应用、证明理论、函数式编程、SAT/SMT 以及程序建模 6 个领域，重点讨论命题逻辑、谓词逻辑、可满足性模理论等。

根据课程大纲，课程内容具体可划分为 11 个章节，分别为：基础知识、命题逻辑、布尔可满足性、谓词逻辑、等式与未解释函数、线性算术理论、数据结构理论、理论组合、符号执行、程序验证、程序合成。具体章节内容安排如表 II 所示。

2) 课程实践: 课程实践共设计了 9 个，部分章节的实践作业进行了合并。课程实践的设计遵循了两个主要原则：一是紧密贴合课程的理论内容；二是紧跟形式化方法理论在实际中的应用。课程实践中共涉及一门语言以及两个软件工具，分别为：Python 编程语言、Coq 辅助定理证明器 [17] 和 Z3 自动定理证明器 [10]。

Python 程序语言。Python 是整个课程实践中的主要程序设计语言，课程实践选择 Python 的原因基于以

下考虑：第一，Python 是一门热门语言，稳居最受欢迎的编程语言的前 10 位，应用广泛；第二，Python 程序上手容易，对于初学者相对友好；第三，很多 SAT/SMT 工具含有 Python 接口，使用比较方便。

Coq 辅助定理证明器。课程使用了 Coq 辅助定理证明器，Coq 是由法国 Inria 开发的一款成熟的定理辅助证明工具；学生通过使用 Coq，完成命题逻辑和谓词逻辑的交互证明，可以加深对命题证明过程的理解。

Z3 定理证明器。课程使用了 Z3 定理证明器，Z3 是由微软研究院开发的一款开源证明器，目前在形式化方法研究中被广泛使用 [18]。Z3 拥有良好的 Python 接口绑定，使用方便。实践作业需要学生基于 Z3 的 Python 绑定，去解决具体的问题。

一般情况下，在每一个课程实践中，我们会设计两种类型的题目：一种是常规题型，要求选课的学生在规定的时间内独立完成；另一个种是挑战题型，主要是针对部分学有余力的同学完成有难度的题目，这部分题目不是必须的。

3) 课程考核: 形式化方法考核有三部分组成：课程实践、期中考试和期末考试，占比分别为：40%，30%，30%。本课程更注重学生平时的课程实践，所以在考核占比中，实践作业的占比要高于期中和期末成绩。对于期中期末考试，我们在考试范围的覆盖上会不同，即期末考试不会涉及期中考试之前的知识，这样的考试设计也对学生学习提出了更高要求。

B. 经验教训

至今为止，更新后的课程已进行了完整的三轮次的授课，根据对授课过程的总结、对课后实践完成情况的分析、以及对课程调查问卷的反馈，发现课程目前仍存在以下问题：

1) 部分学生没有接触过形式化相关知识的学习，对课程内容的理解与掌握仍存在困难。
2) 部分学生对于形式化方法重要性仍缺乏一个正确的理解。
3) 部分学生无法适应大量课程实践，无法按时提交课程作业。

根据以上问题，我们分别在授课内容、课程实践和课程安排上作出了调整。在授课内容方面，适当扩充对知识点历史背景和案例的介绍，以期让学生能更好的理解知识点的全貌和发展脉络。在课程实践方面，增加了对形式化方法理论的实际应用场景的讨论，加强学生对形式化方法重要性的理解。同时，考虑到形式化方法课程的抽象性和难度，在课程的安排上，我们采取理论授课和回顾课相结合的方式，在回顾课上，由课程助教负责带领学生完成对课程内容的复习和巩固，并且有针对性的补充一些相关知识点及解答学生的疑问。

此外，为了及时回答学生的疑问以及了解学生的学习进度，我们采用 Piazza [19] 开放平台对课程进行管理，方便老师和学生进行互动交流和研讨；采用 QQ 群组进行课程相关消息发布；采用学校提供的作业平台提交实践作业和评定成绩。

V. 结语

一门完善、高质量的形式化方法课程，对于培养掌握和理解形式化方法理论和技术的专业人才，具有积极的促进作用。尽管我们已做出了一些有意义的尝试，并取得了一些成效，但形式化课程的完善与质量提升、以及形式化方法的专业人才培养是一个长期的过程，仍有一段长的路要走，需要持续不断的摸索、实践以及完善。

参考文献

[1] https://ncstatic.clewm.net/rsrc/2021/0129/15/e93a0825fa2b6d35b2ee59b70fab6bbb.pdf

[2] 王戟, 詹乃军, 冯新宇, 刘志明. 形式化方法概貌 [J]. 软件学报,2019,30(01):33-61.

[3] Joint Task Force on Computing Curricula (ACM and IEEE). Computer Science Curricula 2013: Curriculum Guidelines forUndergraduate Degree Programs in Computer Science. New York: ACM Press, 2013. 27-38.

[4] 刘波, 刘志明, 裘宗燕, 秦晓. 加强计算机本科专业程序正确性知识教育与能力培养 [J]. 计算机教育,2018(02):135-139.

[5] D. van Oheimb, "Formal Security Analysis in Industry, at the Example of Electronic Distribution of Aircraft Software (EDS)," Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (isola 2006), 2006, pp. 5-5, doi: 10.1109/ISoLA.2006.55.

[6] Davidson D, Moench B, Ristenpart T, et al. FIE on firmware: Finding vulnerabilities in embedded systems using symbolic execution[C]//22nd USENIX Security Symposium (USENIX Security 13). 2013: 463-478.

[7] S. Matsuo, "How formal analysis and verification add security to blockchain-based systems," 2017 Formal Methods in Computer Aided Design (FMCAD), 2017, pp. 1-4, doi: 10.23919/FMCAD.2017.8102228.

[8] Yang P, Li J, Liu J, et al. Enhancing robustness verification for deep neural networks via symbolic propagation[J]. Formal Aspects of Computing, 2021: 1-29.

[9] 钱一名, 刘志凯, 梁成华, 王冬. 核电安全级仪控系统形式化功能验证 [J]. 核电子学与探测技术,2019,39(05):621-625.

[10] Z3.https://github.com/Z3Prover/z3

[11] https://www.microsoft.com/en-us/research/project/slam/

[12] https://fbinfer.com/

[13] https://www.messenger.com/

[14] https://instagram.com/

[15] Turing A. Checking a large routine. Report of a Conf. on High Speed Automatic Calculating Machines, Cambridge University Math.Lab., 1949. 67-69.

[16] McCarthy J. Towards a mathematical science of computation. In: Proc. of the IFIP Congress. 1962. 21-28.

[17] Coq.https://github.com/coq/coq

[18] CCF Formal Methods Technical Committee. Advances and trends on formal methods. In: The Progress Report of ComputerScience and Technology in China from 2017 to 2018. Beijing: China Machine Press, 2018. 1-68 (in Chinese with English abstract).

[19] https://piazza.com/

[20] 张昱, 许胤龙. 编程语言与原理类课程设置及教学内容探索 [J]. 计算机教育,2019(02):19-22.

[21] 刘强, 陈越, 骆斌, 等. “软件工程”课程教学实施方案 [J]. 中国大学教学, 2011, 2: 41-44.