交互式定理证明学术会议ITP 2015简介

张兴元[1] (email: xingyuanzhang@126.com)
Christian Urban[2] (email: christian.urban@kcl.ac.uk)
吴春寒[1] (email: wuchunhan@gmail.com)
王金双[1] (email: wangjinshuang@139.com)

[1] 解放军理工大学，中国南京
(PLA University of Science and Technology, China)
[2] 伦敦国王学院，英国伦敦
(King's College London, United Kingdom, United Kingdom)

2015年度的“交互式定理证明”学术会议（Interactive Theorem Proving，简称ITP 2015）将于今年8月24至27日在南京举行，会议网址为：

http://www.inf.kcl.ac.uk/staff/urbanc/itp-2015/

这是该系列会议首次在欧美之外的地区举行，会议的主办者希望通过这次会议推动交互式定理证明技术在中国的发展。在正式会期的前后将分别开设定理证明系统Isabelle（网址：https://www.cl.cam.ac.uk/research/hvg/Isabelle/）（暑期学校时间：8月21日至23日）和Coq（网址：https://coq.inria.fr/）（暑期学校时间：8月27日至29日）的暑期学校，邀请此领域的著名学者讲课，为有意应用此项技术进行研究的老师和同学提供启蒙服务。此次会议由伦敦大学国王学院的Christian Urban和解放军理工大学的张兴元共同担任会议主席（Co-Chair），我们竭诚期望计算机学会同行们的大力支持和积极参与。

1 交互式定理证明技术简介

“交互式定理证明”是指通过用户与计算机之间的交互来表述和证明符号逻辑中的命题，证明的正确性由计算机自动检查，具有很高的可信度，简单的证明还可以由计算机自动完成以节省人力。此项技术是符号逻辑与计算机系统相互作用的产物。可以说符号逻辑与计算机有着不解之缘：符号逻辑最初是为数学命题及其形式化的证明而引入的，“计算机”这一概念本身就是符号逻辑研究的自然结果，而计算机所提供的强大计算能力也是符号逻辑得以真正实现并发挥作用的物质基础。

随着计算机在社会生活中重要性的日益提高，符号逻辑（特别是交互式定理证明技术）有了更为重要的应用。如果说人们当初在提出符号逻辑的时候所关心的还只是数学命题的真假的话，今天的计算机科学家和工程师试图利用这一技术所解决的问题要难得多。以“用于火箭控制的程序是存在的”这一命题为例，对数学家而言，这也许只是一个不证自明的事实，充其量也不过是一个只需简要说明理由的简单命题。但对计算机工程师而言，他们必须给出该命题的一个“构造性证明”：包括一个具体的火箭控制程序及其正确性证明。如果说，数学证明中的微小错误无损命题本身的成立的话，那么，火箭控制程序及其正确性证明中任何微小的错误都有可能导致火箭坠毁、人员伤亡之类的重大事故。

确保计算机系统（特别是软件系统）的正确性有着极高的难度。每次考试都能得99分的学生无疑是优异的，即使如此优秀的学生，每做100道题，也会有一处错误，而且这个错误是他自己发现不了的。假如让如此优秀的学生来编写一个C程序，每100个函数就会有一个自己发现不了的错误。而随着计算机系统硬件集成度和容量的快速提升，现代计算机软件的规模又何止于100个函数。更加糟糕的是，一旦知道错在哪里，软件错误修改起来相当容易，这也就意味着，在运行阶段出现的软件错误通常是其编写者自己发现不了的，或者说，是在其认知能力之外的。这就是所谓“计算机系统的正确性问题”，一个长期困扰着计算机科学家和工程师的难题。为解决这一难题而提出的“形式化方法”就是一种试图通过符号逻辑的方法和技术来表述和证明计算机系统的正确性的技术途径。在“形式化方法”的诸多支撑技术中，“交互式定理证明”具有不可替代的独特作用。由于计算机系统的非平凡性质通常是不可判定的，像“模型检验”、“静态检查”这样全自动验证技术不能用于解决所有的非平凡性质的验证。对于那些用全自动技术无法解决的验证问题，只能通过“交互式定理证明”所特有的人机交互功能，以半自动化的方式，在人类智慧的引导下来解决。这就是“交互式定理证明”真正的用武之地。

2 交互式定理证明简史

利用计算机进行符号逻辑推理的工作始自上世纪70年代，早期具有代表性的系统有Automath[1]，Stanford LCF[2]和Mizar[3]，其最初动机通常是对某种数学

理论进行形式化，但随着计算机系统的广泛应用，人们逐渐认识到其在解决计算机系统自身正确性方面的作用，从而为其发展注入了新的、更为强劲的动因。这一研究传统延绵至今，形成了一系列具有重要影响的交互式定理证明系统，如HOL[4], Isabelle[5]，Coq[6]，PVS[7]，ACL2[8]，Nuprl[9]，AGDA[10]等。如今，人们已经可以在这些系统中对诸如数据库系统[11]、编译程序[12]、操作系统[13]等大型软件系统的正确性进行形式验证，相关的结果也都发表在POPL，PLDI，SOSP等CCF推荐的A类国际会议上[3]。其中，[13]中所描述的，由来自澳大利亚国家信息科学研究中心（NICTA）的科学家采用交互式定理证明技术对操作系统微内核seL4所进行的形式验证工作被2011年度的“MIT技术评论”（MIT Technolgy Review)列为该年度10大突破性技术之一[14]。该项技术被认为有可能在未来10年改变信息技术的全貌。而采用交互式定理证明技术对研究结果进行形式化正在成为顶级会议上某种新的学术标准。

ITP会议（及其前身TPHOLs）就是从事交互式定理证明研究的学者每年聚会交流、发表最新研究结果的主要渠道之一。该会议自1988年以来，每年交替在欧洲或者北美的学术机构举行，互联网上有详细的记录[4]。对这些记录的研究可以发现，前述各顶级国际会议的程序委员会中不乏来自ITP社区的重要学者，有不少研究成果往往是先在ITP（或TPHOLs）上发表，再经过深化拓展后被前述顶级国际会议录用。可以说，ITP是通往这些顶级会议的重要通道。

3 交互式定理证明研究趋势

当前，交互式定理证明的前沿有两个主要热点：“代码级别的程序验证”和“复杂数学理论的形式化”。代码级别的验证是计算机工作者最为关心的问题。虽然已经可以对大规模的软件进行形式验证，但需要大量的人工干预，工作量太大，有碍于推广。为此，需要对现有的方法和工具进行改进，找到更加合适的人机交互方式和更多可以自动化的成分从而降低代码级验证的难度。目前的验证方法多集中于顺序程序的验证，如何对并发程序进行验证，也是一个有待解决的问题。数学理论的形式化是对任何其它系统进行形式化所必须的基础，也是对交互式定理证明技术的发展及其可用性的一块试金石。在目前的发

[3] http://www.ccf.org.cn/sites/ccf/paiming.jsp
[4] http://www.cs.uwyo.edu/~ruben/itp-2014/Main/History

展水平下，已经可以对很多复杂的数学定理（如四色定理，奇序定理等）进行形式化，引起了学界广泛的重视，相关的研究正处于快速发展之中。

4 ITP 会议简介

“交互式定理证明”会议（ITP）是机器定理证明方向的主要国际会议之一，它与CADE[15]，TABLEAUX[16]一起构成了“自动推理联合国际会议”（International Joint Conference on Automated Reasoning，简称IJCAR[17]）的三个主要子会议。同时也是四年一度的“联合逻辑会议（Federated Logic Conference，简称FLoC[18]）的固定子会议。会议内容包括机器定理证明系统的基础理论、实现方法及其在计算机系统的形式验证和数学理论形式化中的应用等诸多论题。

该会议始于1988年，原名“高阶逻辑中的定理证明”（Theorem Proving in Higher Order Logics ，简称TPHOLs ），2010年更名为ITP。自1990年始，会议论文集均作为Springer的LCNS（Lecture Notes in Computer Science）系列出版。根据CiteseerX[19] 在2008年对581个信息类国际会议影响因子的统计，TPHOLs得分0.07，居于第87位（前14.9%）。

每期 ITP 会议论文要经过3至4位专家匿名评审，最终录用25篇左右的正式论文，其中通常有两篇“证明珍珠”（Proof Pearl），代表了精巧而富于创意的证明技术。除正式论文之外，会议还开辟了“粗钻石”（Rough Diamonds）专栏，用于发表那些虽然不尽完善，但富有创意和研究前景的工作。每篇正式论文有30分钟的报告时间和10分钟的答辩时间。“粗钻石”论文也有15分钟的报告时间和简短的答辩环节。

References

1. Automath. `http://www.cs.ru.nl/~freek/aut/`.
2. R. Milner. LCF: A Way of Doing Proofs with a Machine. In *Mathematical Foundations of Computer Science,*, pages 146–159–72. Springer, 1979.
3. A. Naumowicz and A. Kornilowicz. A Brief Overview of Mizar. In *Proc. of TPHOLs 2009*, pages 67–72, 2009.
4. M. Gordon. From LCF to HOL: a short history. In *Proof, Language, and Interaction*, pages 169–185. MIT Press, 2000.
5. M. Wenzel T. Nipkow, L. C. Paulson. Isabelle/HOL: A Proof Assistant for Higher-Order Logic. Springer, 2002.
6. The Coq Proof Assistant. `http://coq.inria.fr`.

7. S. Owre, J. M. Rushby, and N. Shankar. PVS: A Prototype Verification System. In *Proc. of CADE 1992*, 1992.
8. ACL2. http://www.cs.utexas.edu/users/moore/acl2/.
9. PRL Project. http://www.nuprl.org/.
10. Agda wiki page. http://wiki.portal.chalmers.se/agda/.
11. G. Malecha, G. Morrisett, A. Shinnar, and R. Wisnesky. Towards A Verified Relational Database Management System. In *Proc. of POPL'10*, 2010.
12. X. Leroy. Formal Verification of a Realistic Compiler. *Communications of the ACM*, 52(7):107–115, 2009.
13. G. Klein, J. Andronick, K. Elphinstone, G. Heiser, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood. seL4: Formal Verification of an OS Kernel. *Communications of the ACM*, 53(6):107–115, 2010.
14. W. Bulkeley. Crash-proof code, making critical software safer. http://www2.technologyreview.com/article/423692/crash-proof-code/, 2011.
15. Conference on Automated Deduction (CADE). http://www.cs.albany.edu/~nvm/cade.html.
16. International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX). http://i12www.ira.uka.de/TABLEAUX/.
17. International Joint Conference on Automated Reasoning. http://en.wikipedia.org/wiki/International_Joint_Conference_on_Automated_Reasoning.
18. Federated Logic Conference. http://www.floc-conference.org.
19. CiteSeerX. http://citeseerx.ist.psu.edu/index.