程序理论

詹博华

中国科学院软件研究所 / 中国科学院大学

2023 年 12 月

课程介绍

数理逻辑和程序理论

- 六次课程：12/13, 20, 27, 1/3, 10, 17
- 作业总共 30 分，课程的 30%：
  - 手写作业：15 分，50%
  - Isabelle 证明：15 分，50%

自我介绍

詹博华

- 2018.08 – 现在：中国科学院软件研究所，副研究员
- 2017.08 – 2018.07：慕尼黑工业大学，博士后
- 2014.09 – 2017.07：麻省理工学院，博士后
- 2010.09 – 2014.06：普林斯顿大学，博士

主要研究方向：交互式定理证明、程序验证、形式化数学

程序分析

分析程序（计算机系统），证明正确性和安全性

主要方法：

- 静态分析（编译理论）
- 模型检验（自动机理论）
- 基于推理的验证（应用逻辑）

应用领域

主要应用于需要极高安全保障的安全攸关系统

- 航空航天、高速列车系统
- 医疗系统
- 核电站控制系统
- 操作系统内核
- 硬件设计

应用领域

主要应用于需要极高安全保障的安全攸关系统

- 航空航天、高速列车系统
- 医疗系统
- 核电站控制系统
- 操作系统内核
- 硬件设计

How can we provide people with cyber-physical systems they can bet their lives on? — Jeannette Wing

Intel Pentium FDIV bug

- 影响一款早期（1994 年）的 Intel 芯片
- 平均每 90 亿个除法计算中有一个出错
- 对 Intel 造成的损失约 $475 million

Ariane 5 Disaster

- 火箭在升空 40 秒后被摧毁（1996 年）
- 调查表明事故原因是从 64 位浮点数到 16 位整数转换造成的溢出
- 火箭和装载的卫星价值约为 $500 million

Therac-25 Bug

- 由计算机控制的放射线医疗系统
- 先后6次事故，造成3名病人死亡（1985-1987 年）
- 错误原因是并发程序造成的数据竞争

近期案例

- Boeing 737 MAX 飞行控制系统：造成两架飞机坠毁，346 人死亡
- DAO Attack: 针对以太坊智能合约的攻击
- Meltdown & Spectre: 芯片安全漏洞
- ...

基于推理的验证（Deductive Verification）

- 为程序/系统的正确性提供数学证明。
- 功能正确性：程序/系统的行为完全满足规约。
- 与普通的测试相比：通过有限多的计算覆盖所有（潜在无限多个）可能的程序状态。
- 与静态分析相比：能够证明更强、更准确的属性。
- 与模型检验相比：不受状态空间大小的限制。
- 局限：证明需要人工参与，需要大量时间和人力。

程序验证过程

基于推理的验证主要分为以下几步：

1. 建立严格的程序语言语义（实际行为是什么？）
2. 设置严格的系统规约（期望的行为是什么？）
3. 将程序正确性表达为数学定理，通过程序逻辑（例如霍尔逻辑或其扩展）证明。

(半) 自动程序验证

- 给定程序和需要验证的性质，自定/半自动地完成验证。
- 基于 SMT 求解器的使用。
- 更加容易使用，但难以验证很深/很复杂的性质。
- 主要 SMT 求解器: Z3、CVC4、等等。
- 主要程序验证工具: Dafny、Why3、FramaC、等等。

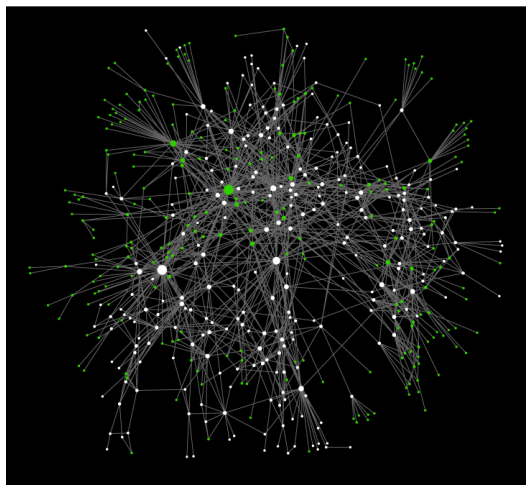
交互式定理证明

- 在人的引导下完成定理和程序正确性的验证。
- 两个最常用的工具：Coq 和 Isabelle

- 其他工具：HOL Light, HOL4, Mizar, PVS, ACL2, Lean...

seL4：微型操作系统内核的验证

- 首个具有完整正确性证明的微型操作系统内核 (micro-kernel)。
- ~8,700 行 C 代码，600 行汇编代码。
- >200,000 行 Isabelle 证明。
- 耗时 20-30 人年。



Klein et al. seL4: Formal verification of an OS Kernel. SOSP 2009.

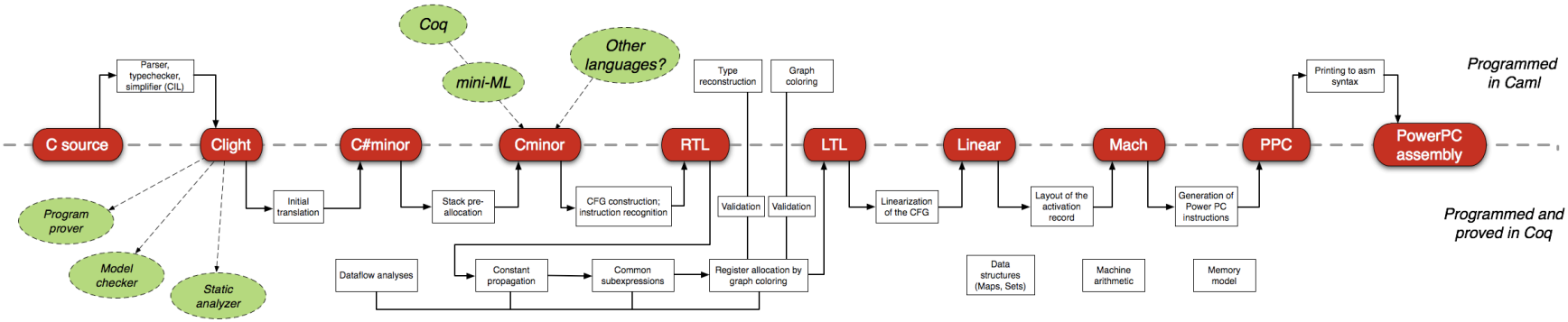
seL4：微型操作系统内核的验证

在后续的研究中，将 seL4 操作系统应用于无人驾驶的直升机，证实系统可以有效防御黑客攻击。

Klein et al. Formally verified software in the real world, Comm. of the ACM

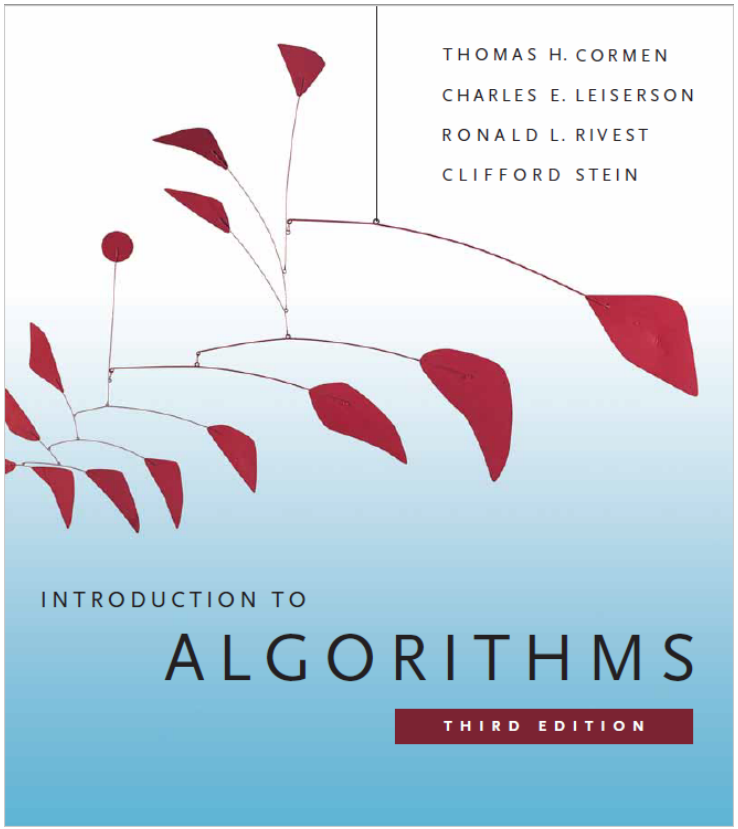
CompCert：编译器的验证

- 在给定源代码语言（例如 C 语言）和目标语言（例如汇编语言）的语义后，可以验证一个编译器的正确性。
- CompCert：第一个形式化验证的 C 编译器。
- 大约相当于 gcc -O1 的优化程度。
- 在一次编译器测试研究中，CompCert 是 11 个编译器里**唯一没有发现中后端错误的编译器**（但仍在未验证的前端发现了错误）。



算法和数据结构的验证

- 经典教科书“算法导论”的前半部分算法基本上都已验证。
- 验证包括算法返回正确的结果，以及时间复杂度分析。
- 后半部分大约一半的内容已被验证。



课程大纲

理论部分：

- 操作语义和指称语义
- 霍尔逻辑（Hoare logic, **公理语义**）
- 分离逻辑（Separation logic）

实践部分：

- 高阶逻辑（higher-order logic）
- Isabelle 基础
- 使用 Isabelle 验证简单程序

参考书目

- Mike Gordon: Specification and Verification I
  https://www.dcc.fc.up.pt/~nam/web/resources/vfs13/Notes.pdf
- Tobias Nipkow, Gerwin Klein: Concrete Semantics
  http://www.concrete-semantics.org/
- 周巢尘、詹乃军：形式语义学引论