系统分析与验证

主讲：郭建
jguo@sei.ecnu.edu.cn

电话：62224039　　办公室：数学馆308

上课地点：数学馆113　时间：周一下午17:00-18:30

参考资料

- Christel Baier and Joost-Pieter Katoen, Principles of Model Checking
- Handbook of Model checking, E.M.clarke etc.2018.
- Concepts, Algorithms, and Tools for Model Checking by Joost-Pieter Katoen
- ……

课程目标

- 学生能够对嵌入式系统进行高层建模、规范描述和自动化、半自动化验证
  - Methods for modeling systems, such as embedded systems.
  - Logical formalisms for expressing properties related to the safety, security and performance of systems.
  - Validation and verification tools that determine the relationship between models and properties in order to establish strong guarantees related to safety, security, and performance.

预备知识

- 离散数学
- 数据结构
- 编译原理
- ……

- 为什么要验证？

验证方法分类

- 模拟:动态验证，简单易行，主流方法
- 仿真:依赖于硬件
- 形式化验证:静态验证，完备性高，重要补充
- 半形式验证:两者结合，越来越重要的方法
  - 基于断言的验证

形式化验证（Formal Verification）

- 形式化验证
  - 从硬件设计上，是指从数学上完备地证明或验证电路的实现方案是否确实现了电路设计描述的功能
  - 从软件设计上，是指从数学上完备地证明或验证系统的软件是否满足系统的需求规范

形式化验证的分类

按照验证的内容或需要分类

- 性质检验(property checking)

  一个设计是否包含某些设计和功能？

- 等价性验证(equivalence verification)

  两个设计在功能上是否等价？

  - 硬件设计：功能及性质说明→行为设计→结构设计→物理设计
  - 软件设计：高级语言→汇编语言→机器指令

形式验证的分类

按照验证的方法分类

- 定理证明(Theory Proving)
- 模型检验(model checking)
- 等价性验证（equivalence proving）

等价性检验

- 问题：
  - 给定两个组合电路，检验在所有输入下，这两个电路的输出是否一致。
- 输入分类:使输出为0;使输出为1;不管项
- 等价条件：使它们输出不同的每个输入向量,至少是其中一个电路的不管项。
- 工具
  - Candence的Affirma，
  - Verplex的Logic Equivalence Checker，
  - Synopsys的 Formality，
  - Mentor Graphics的FormalPro

等价性检验

- 途径：符号方法和增量方法
- 符号方法：
  - 将问题形式化成为符号表示，然后用特定的问题求解方法，如BDD，SAT。
  - 适用：系统缺乏结构相似性；不能得到系统设计的内部情况时。

等价性检验

- 增量方法：
  - 利用被检验的系统在结构上存在的相似性，逐步检验内部局部子系统的等价性，进而最终得到系统在整体上的等价性。
  - 适用于相似性比较大的系统。

定理证明

- 使用定理证明器
- 方法：
  - 首先从原始设计中抽取模型，表示成形式逻辑的命题、谓词、定理、推理规则等
  - 需要验证的性质被表示成定理的形式
  - 在**验证者的引导**下，定理证明器不断地对公理、已证明的定理施加推理规则，产生新的定理，直至推导出所需要的定理
- 形式逻辑
  - 一阶逻辑
  - 高阶逻辑
  - 时态逻辑

定理证明

- 优点：
  - 高度抽象，强大的逻辑表达能力
  - 应用不受限制，可以表示和验证几乎所有的系统行为特性
- 缺点：
  - 需要人工引导
  - 需要验证者有良好的数学训练和经验
  - “There are no fully automatic **theorem provers”.**

定理证明系统

- 建立在某种形式逻辑基础上
- 内置各种推理规则、推理对策、元对策等，成为验证复杂系统的有力工具
- 从公理开始，寻找一个证明序列
- 该序列中的每个元素都是从前面的公理和定理推导出来的定理
- 证明序列最终以证明的结果出现而结束
- 著名系统：PVS, LCF, ACL2, HOL, isabelle，coq 。其中，HOL,PVS系统为高阶逻辑证明系统

不同形式逻辑之间的比较

- 命题逻辑：传统布尔代数，变量∈{0，1}
- 一阶逻辑（谓词逻辑）：包含∀，∃量词
- 高阶逻辑：包括对集合和函数的推导
- 时态逻辑：包括关于时间操作符

模型检验

- 模型检验：模型检验是针对状态并发系统的一种自动验证技术
- 由美国的Clarke, Emerson, and Sistla等和法国的Queille and Sifakis等在1981年初期分别独立提出
- 特点
  - 系统用有限状态结构表示
  - 被验证的性质采用时态逻辑公式表示
  - 验证过程就是对设计的状态空间的全搜索过程，确定被验证的性质在状态空间中的可达性/不可达性

模型检验的优缺点

- 优点
  - 快速
  - 无须人机交互，自动进行
  - 对于给定的性质不存在**bug**
  - 断定性质不满足时，能给出反例
- 挑战
  - 如何解决状态爆炸问题

模型检验发展

- 显式模型检验（EMC）
  - 基于直接搜索
  - **1981， Clarke， Queille**
- 符号模型检验（SMC）
  - 基于**OBDD**隐式搜索
  - **1987， McMillan**
- 定界模型检验（BMC）
  - 基于**SAT**
  - **1999， Biere ， Clarke**
- 无界模型检验（UMC）
  - 基于**SAT**
  - **2002， McMillan**

模型检验技术

- 离散系统
  - 基于Kripke 结构模型检验
    - SMV, nuSMV, SPIN, VIS,
  - 基于时间自动机的模型检验
    - UPPAAL
  - 基于概率(统计)的模型检验
    - UPPAAL
    - Prism
- 混成系统
  - 离散、连续模型

形式化验证研究的发展

- 70年代

  形式逻辑，断言归纳，

- 80年代

  时态逻辑， 模型检验， BDD等价性验证

- 90年代

  门级等价性验证工具

  模型检验工具

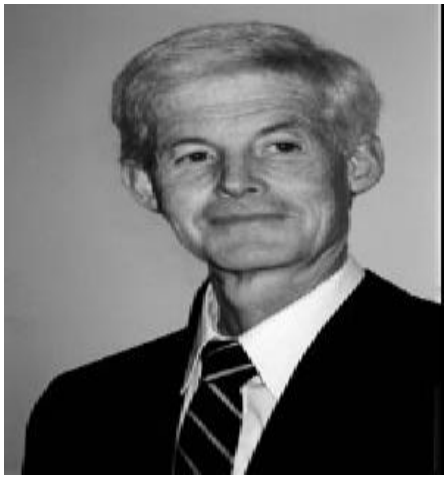
  模型检验发展

- 2000年以来

  RTL等价性验证与模型检验

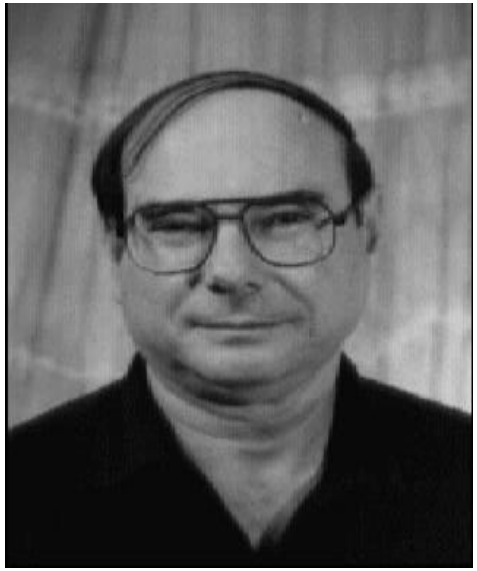
  出现专门的验证公司

形式化验证相关的其他图灵奖得主



- 1982年图灵奖获得者， Stephen Arthur Cook
  - NP完全性理论的奠基人
  - 1971年，“The Complexity of Theorem Proving Procedures.”

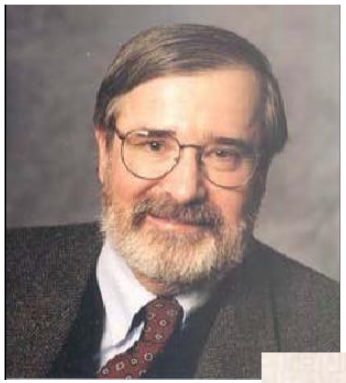
形式验证相关的其他图灵奖得主

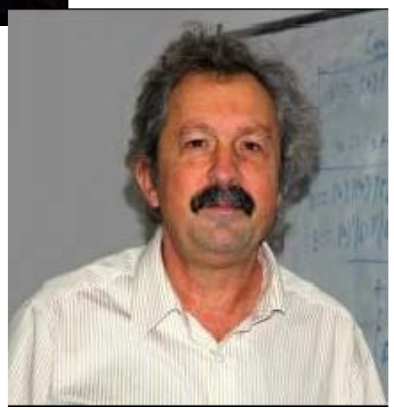


- 1996年图灵奖获得者，Amir Pnueli
  - 1977年，把时态逻辑引入计算机科学，把它作为开发反应式系统和并发式系统时规范说明（specification）和验证的工具。
  - 命题线性时态逻辑（PLTL）

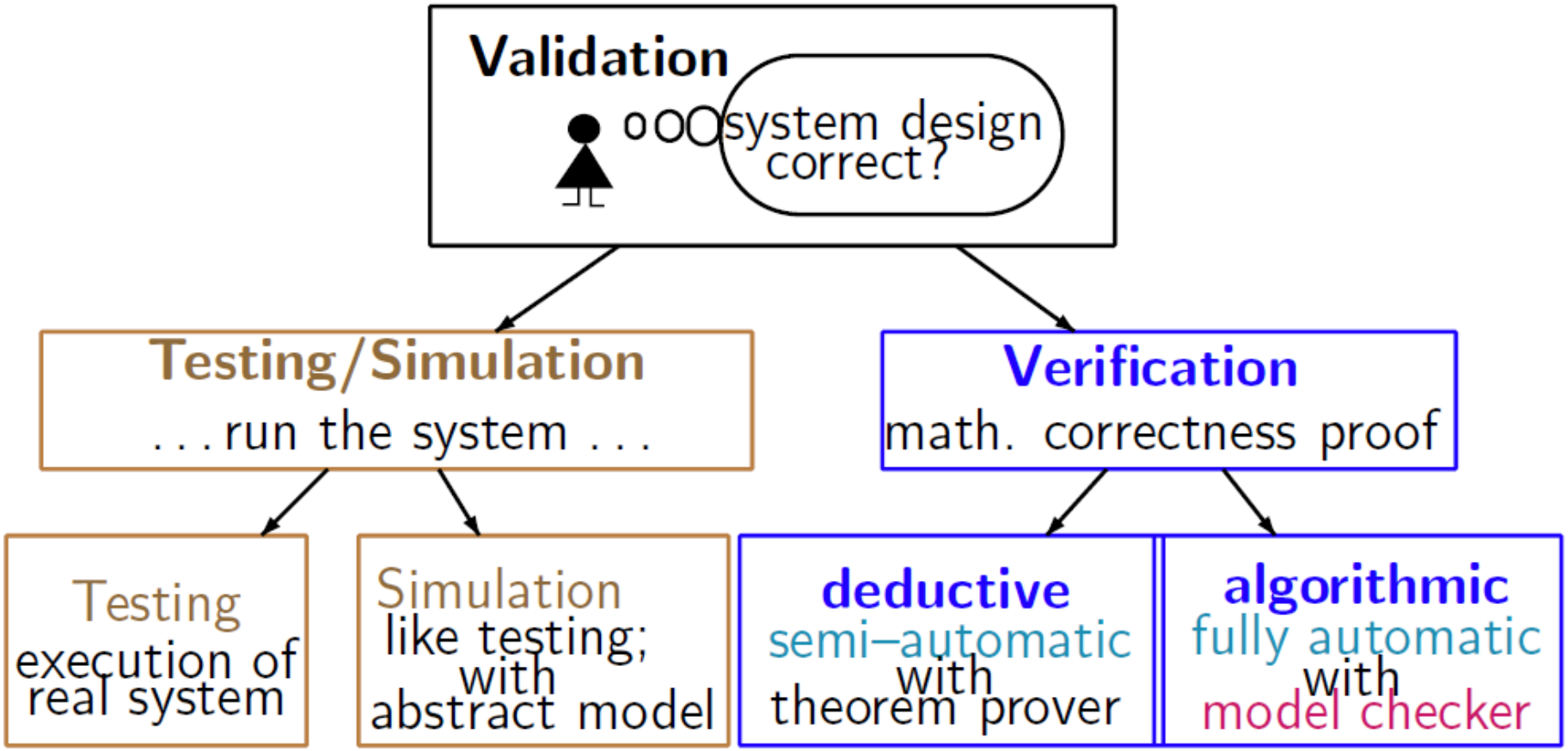
2007年图灵奖

- Edmund M. Clarke，Allen Emerson和Joseph Sifakis荣获2007年的图灵奖。
- “For his role in developing Model-Checking into a highly effective verification technology, widely adopted in the hardware and software industries”
- （“在将模型检验发展为被硬件和软件业中所广泛采纳的高效验证技术上的贡献”）





validation methods



This figure is from Christel Baier (TU Dresden) and Joost-Pieter Katoen

Model checking

- Verify finite state concurrent system
- Perform automatically
- Given **sufficient resources**, terminate with a yes/no answer
- Application
  - operating systems
  - control systems for machines；hardware systems
  - communication protocols
- Restricting Unbounded data structure to specific instances

The process of model checking

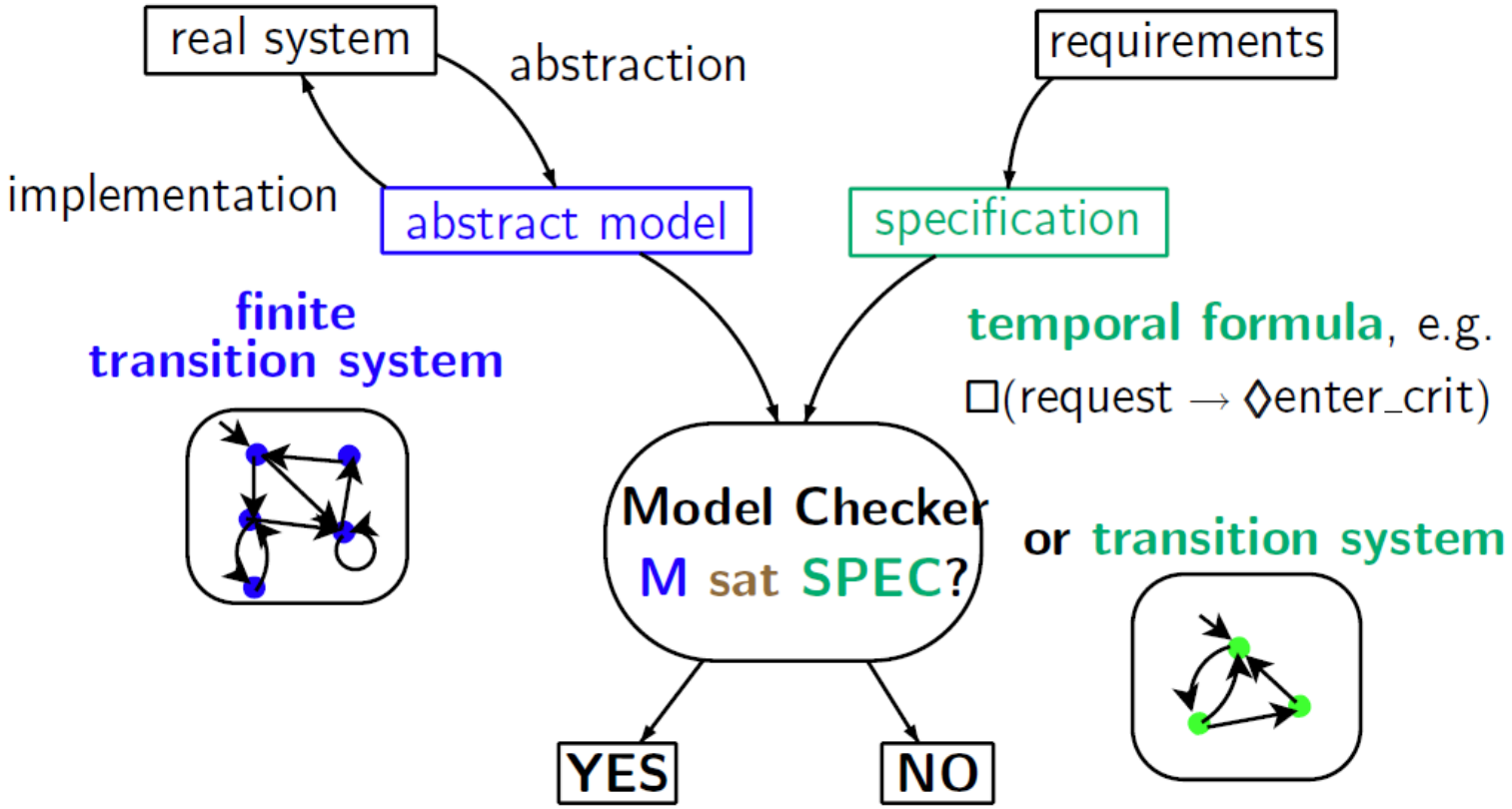
- Modeling
  - Compilation task: convert a design into a formalism
  - The use of abstraction to eliminate irrelevant or unimportant
- Specification
  - The property that the design must satisfy, in some logical formalism
  - Temporal logic
  - Completeness: specification can not cover all properties that the system should satisfy

The process of model checking (cont)

- Verification
  - Whether the system model satisfies the properties
  - The analysis of the verification result----human assistance
  - For a negative result, provide an error trace (counterexample)
    - incorrect modeling of the system
    - incorrect specification----false negative

The process of model checking(cont')

- Verification



This figure is from Christel Baier (TU Dresden) and Joost-Pieter Katoen

The process of model checking (cont)

- Verification (cont)
  - Due to the size of the model, the verification will fail to terminate normally
    - Using abstractions
    - Adjusting the model

State explosion

- Given n processes represented by program graphs $P_1, \ldots, P_n$ with 2 locations each
- on the set of variables $Var = \{x_1, \ldots, x_m\}$, with $Dom(x_i) = \{0, 1\}$
- How many states has the transition system T: $P_1 ||| \ldots ||| P_n$?
- $2^n \cdot 2^m$

Temporal logic and model checking

- Temporal logic
  - Linear structure
  - Branching structure

Temporal logic and model checking

- Temporal logic
  - Describe the ordering of events in time without introducing time explicitly
  - Burstall, Kroger, Pnueli using temporal logic for reasoning about computer programs
  - Pnueli uses temporal logic for reasoning about concurrency
    - A set of axioms that described the behavior of individual statements in the program
    - Extend to sequential circuits by Bochmann and Malachi and Owicki

Temporal logic and model checking (con't)

- By Clarke and Emerson---CTL
  - A single model satisfies a formula
  - Complexity: the polynomial in both the size of the model determined and the length of its specification.
  - Handle fairness without changing the complexity
  - An improved algorithm—linear in the product of the length of the formula and the size of model—EMC
  - states—10^4~10^5 at a rate of about 100 states/second
  - Examples :network protocols, sequential circuits

Temporal logic and model checking (cont)

- Model checking(cont)
  - Model checking for a variety of temporal logics
    - LTL is PSPACE-complete—Clarke and Sistla
    - LTL model checking is still acceptable for short formula.– Pnueli and Lichtenstein
    - Tableau method based verification system for LTL formulas--Fujita

Temporal logic and model checking (cont)

- Model checking (cont)
  - Model checking for a variety of temporal logics (cont)
    - CTL*--combine both LTL and CTL—Clarke,Emerson and sistla
  - Other techniques for verifying concurrent system
    - Automata theory—Vardi， Wolper
      - Use automata for the spec and an impl
      - Check whether the impl conforms the spec—language containment
      - Verifier—COSPAN

Symbolic algorithm

- McMillan used a symbolic representation for the state transition graphs--1987
  - OBDD—proposed by bryant in1986
  - Transition relation—assignments of two sets of variables
  - Computing fixpoints of predicate transformers from the transition relation
- Tool—SMV --NuSMV

总结

- 形式化方法概述
- 模型检验技术概述