大作业

课堂演示

- 分组
  - 3人
- 演示ppt
  - 讲述论文解决的问题和方法
  - 讲述论文的缺点和改进方向
  - 每位组员自己研究方向和问题的简介
  - 自己研究方向和论文或领域特定语言的思考
- 课堂演示安排：
  - 从第9周开始
  - 每次5组，介绍25分钟，讨论5分钟

阅读报告

- 阅读报告：
  1. 选题说明
  2. 论文要解决的问题
  3. 论文的解决方法
  4. 论文的主要创新之处
  5. 论文对你的启发
  6. 论文的不足之处及可改进
  7. ***自己研究方向和问题介绍***
  8. ***结合自己方向的进一步工作***

- 作业提交时间：第16周之前

会议列表

- SOSP
- OSDI
- ATC
- EMSOFT
- DAC
- MODELS
- ICCPS
- OOPSLA
- ICSE
- FSE
- ASE

论文列表

清華大學 Tsinghua University

- SOSP
  - Snowboard: Finding Kernel Concurrency Bugs through Systematic Inter-thread Communication Analysis SOSP'21
  - HEALER: Relation Learning Guided Kernel Fuzzing SOSP'2021
  - Optimizing data-intensive computations in existing libraries with split annotations SOSP'19
  - Finding Semantic Bugs in File Systems with an Extensible Fuzzing Framework SOSP’19
  - Rudra: Finding Memory Safety Bugs in Rust at the Ecosystem Scale SOSP’21
  - Efficient scalable thread-safety-violation detection: finding thousands of concurrency bugs during testing SOSP’19
- OSDI
  - Detecting Transactional Bugs in Database Engines via Graph-Based Oracle Construction OSDI’23
  - Finding Consensus Bugs in Ethereum via Multi-transaction Differential Fuzzing OSDI'21
  - Testing database engines via pivoted query synthesis OSDI'20
  - Gauntlet: finding bugs in compilers for programmable packet processing OSDI'20

论文列表

清華大學 Tsinghua University

- ATC
  - Pinolo: Detecting Logical Bugs in Database Management Systems with Approximate Query Synthesis ATC’23
  - KSG: Augmenting Kernel Fuzzing with System Call Specification Generation ATC’22
  - MLEE: Effective Detection of Memory Leaks on Early-Exit Paths in OS Kernels ATC'21
  - FuZZan: Efficient Sanitizer Metadata Design for Fuzzing ATC'20
- EMSOFT
  - Mercury: Instruction Pipeline Aware Code Generation for Simulink Models EMSOFT’22
  - Rtkaller: State-aware Task Generation for RTOS Fuzzing EMSOFT’21
  - Specification-Guided Automated Debugging of CPS Models EMSOFT'20
  - Statistical Verification of Hyper properties for Cyber-Physical Systems EMSOFT'19

论文列表

清華大學 Tsinghua University

- OOPSLA
  - Towards Better Semantics Exploration for Browser Fuzzing OOPSLA ’23
  - Coverage-guided tensor compiler fuzzing with joint IR-pass mutation OOPSLA ’22
  - Satisfiability modulo fuzzing: a synergistic combination of SMT solving and fuzzing. OOPSLA ’22
- DAC
  - STCG: State-Aware Test Case Generation for Simulink Models DAC’23
  - HCG: Optimizing Embedded Code Generation of Simulink with SIMD Instruction Synthesis DAC’22
  - Trusting the trust anchor: towards detecting cross-layer vulnerabilities with hardware fuzzing DAC'22

论文列表

- ICSE
  - Automated Repair of Programs from Large Language Models . ICSE’23
  - Analysing the Impact of Workloads on Modeling the Performance of Configurable Software Systems. ICSE’23
  - Syntax and Domain Aware Model for Unsupervised Program Translation. ICSE’23
  - JITfuzz: Coverage-guided Fuzzing for JVM Just-in-Time Compiler. ICSE’23
  - Easy modelling and verification of unpredictable and preemptive interrupt-driven systems. ICSE'19
  - Practical GUI testing of Android applications via model abstraction and refinement. ICSE'19
  - Automatic model generation from documentation for Java API functions. ICSE'16
  - Automated test suite generation for time-continuous simulink models. ICSE'16

论文列表

清華大學 Tsinghua University

- FSE
  - SEDiff: scope-aware differential fuzzing to test internal function models in symbolic execution. FSE’22
  - Minerva: browser API fuzzing with dynamic mod-ref analysis. FSE’22
  - RoboFuzz: fuzzing robotic systems over robot operating system (ROS) for finding correctness bugs.FSE’22
  - Detecting Simulink compiler bugs via controllable zombie blocks mutation. FSE’22
  - Generating automated and online test oracles for Simulink models with continuous and uncertain behaviors. FSE'19
  - Model-based testing of breaking changes in Node.js libraries. FSE'19
  - DeepStellar: model-based quantitative analysis of stateful deep learning systems. FSE'19
  - Evaluating model testing and model checking for finding requirements violations in Simulink models. FSE'19

论文列表

- ICCPS
  - Platform for model-based design and testing for deep brain stimulation ICCPS'18
- ASE
  - VITAS : Guided Model-based VUI Testing of VPA Apps. ASE’23
  - Trace-Checking Signal-Based Temporal Properties: A Model-Driven Approach ASE’20
- MODELS
  - Efficient reordering and replay of execution traces of distributed reactive systems in the context of model-driven development MODELS'20
  - Supporting robotic software migration using static analysis and model-driven engineering MODELS'20
  - On-the-Fly Translation and Execution of OCL-Like Queries on Simulink Models MODELS'19

谢谢