Ov 0 1 2 3 4 5 6 R 7 8 9 10 11 12 13 RD E

Week 0:

Mon, 5 August 2019 - Sat, 10 August 2019

Week 1: Public Holiday

Mon, 12 August 2019 - Fri, 16 August 2019

The monday of week 1 is a public holiday and lecture is canceled.

An informal make-up session will be held on Wednesday August 14th at 2 pm in COM2 02-26 Meeting Room 3.

Reading

http://www.wisdom.weizmann.ac.il/~playbook/

LSCs: Breathing life into Message Sequence Charts, Damm and Harel, Formal Methods in System Design, 2001.

File: synthesis.pdf
Paper explaining how Live Sequence Charts can be seen as visual temporal logics. This is a supplementary reading only.

File: LSCs.pdf
Live Sequence Charts: Breathing Life into Message Sequence Charts by Damm and Harel, published 2001. It captures formal visual requirements for software.

File: Lec1-Req.docx
Example requirements excerpted from a real-life air traffic control system

File: Lec1-2.pptx

Week 2: Models and Specifications

Mon, 19 August 2019 - Fri, 23 August 2019

Formal description of software requirements and temporal logic specifications

Readings:

See files below.

Also see http://spinroot.com/gerard/pdf/marktoberdorf.pdf

SPIN Model checking tool from http://spinroot.com/spin/whatispin.html

File: Lec2.2MC.pptx

File: Holzmann2018_Chapter_Explicit-StateModelChecking.pdf
Describes model checking of state machines by converting Linear Time Temporal Logic properties to finite-state automata over infinite strings. Model checking of LTL is by search.

File: TLMC.PDF
Reading on Temporal Logics and Model Checking excerpted from the book "Model Checking" by Clarke, Grumberg and Peled, published 1999

File: Lec2.1TL.ppt

Week 3: Software Verification

Mon, 26 August 2019 - Fri, 30 August 2019

Description of software verification flows and level of automation.

Reading:

A decade of software model checking with SLAM, Ball, Levin and Rajamani, Communications of the ACM, 2011.

+

References in the above CACM article.

File: Lec2.2MC.pptx

File: SWMC1.pdf
Communications of the ACM article giving an overview of software model checking.

File: SWMC2.pdf
An article describing software model checking published in 2001 by researchers in Microsoft Research. Covers how reasoning about state machines can be adapted to programs

File: Lec3.pptx

Week 4: Symbolic Program Analysis

Mon, 2 September 2019 - Fri, 6 September 2019

Foundations of symbolic analysis and how it blurs the lines between testing, comprehension and verification

Readings:

Symbolic Execution and Program Testing, King, Communications of the ACM, 1976.

Symbolic Execution for software testing: three decades later, Communications of the ACM, 2013.

+

References in the 2013 CACM article.

File: Lec4.pptx

File: klee-osdi-08.pdf
Paper describing the KLEE tool for symbolic execution. This tool constructs a symbolic execution tree, and the constraints at the leaf nodes of the tree can be solved to get tests

File: Cadar.pdf
This Communications of the ACM article describes the use of symbolic execution and symbolic program analysis for automatically generating tests.

File: DART.pdf
Directed Automated Random Testing by Godefroid, Klarlund and Sen, PLDI 2005.

File: klee-stanford-2009.pdf

Week 5: Program Repair

Mon, 9 September 2019 - Fri, 13 September 2019

Further discussions on how symbolic analysis can be used for inferring specifications and enable automated program repair.

Readings:

Automated Program Repair, Le Goues, Pradel and Roychoudhury, Communications of the ACM, 2019.

File: cacm19.pdf
This review article from Communications of the ACM gives an overview of the state-of-the-art in automated program repair, or self-healing software.

File: Lec5.pptx

Week 6: Program Synthesis (and discussion on SAT)
Mon, 16 September 2019 - Fri, 20 September 2019

A discussion on program synthesis advances will be conducted in class in the first part of the class.

+

Advances in SAT solvers fueling research in PL and SE (second part. if this part is not covered I will pick it up in weeks 9 and 10.)

Reading:

Search-based Program Synthesis, Alur, Singh, Fisman and Solar-Lezama, Communications of the ACM, 2018 (first part)

+

Conflict -driven Clause Learning SAT solvers (second part)

File: program_synthesis_now.pdf
Program Synthesis, a book by Sumit Gulwani et al, see https://www.microsoft.com/en-us/research/publication/program-synthesis/

File: CACM'18_Search-based_Program_Synthesis.pdf
This Communications of the ACM paper gives an overview of the state-of-the-art in program synthesis.

Recess Week
Sat, 21 September 2019 - Sun, 29 September 2019

Week 7: Group presentation (with Feedback) + In_class short exam
Mon, 30 September 2019 - Fri, 4 October 2019

Groups of 2, attendance is compulsory

+

IN_CLass Short Exam

Week 8: Group presentation (peer-reviewed)
Mon, 7 October 2019 - Fri, 11 October 2019

Groups of 2, Attendance is compulsory.

Week 9: Software Debugging and Specification Inference
Mon, 14 October 2019 - Fri, 18 October 2019

Techniques for software debugging and use of symbolic methods

Reading

Formula-based Software Debugging

Abhik Roychoudhury, Satish Chandra

Communications of ACM (CACM), 59(7), July 2016.

File: lec9.pptx

Re-uploaded lecture notes on Software Debugging

File: cacm16-roychoudhury.pdf
This Communications of the ACM paper gives an overview of methods which view software debugging as an inference of specifications of intended software behavior.

Week 10: Round up and big picture
Mon, 21 October 2019 - Fri, 25 October 2019

We will conduct this class as a discussion on fuzzing but the main aim is to revise and round up many of the topics on vulnerability detection covered in past weeks, and take an overall outlook.

Readings:

Coverage-based Greybox Fuzzing as Markov Chain Marcel Böhme, Van Thuan Pham, Abhik RoychoudhuryIEEE Transactions on Software Engineering (TSE), 45(5), pp 489-506, May 2019.

+

An overview paper on Fuzzing appears below, see weblink from EPFL

+

Directed Automated Random Testing, Patrice Godefroid, Nils Klarlund and Koushik Sen, PLDI 2005.

Weblink: Short overview paper on Fuzzing

File: Lec10Fuzz.pptx

File: Fuzzing.pdf
This paper describes the state-of-the-art today in software vulnerability detection by employing fuzz testing, a biased random testing method regularly used by corporations.

File: DART.pdf
Directed Automated Random Testing by Godefroid, Klarlund and Sen, PLDI 2005.

Week 11: Presentation by students (Round 2)
Mon, 28 October 2019 - Fri, 1 November 2019

Student led presentations.

Attendance is compulsory.

This lecture will be held on Tuesday 29th October 3 pm at Meeting Room 1 COM1 03-19, since Monday 28 October is a public holiday.

papers presented:

1. CodePhage paper on transplantation, PLDI 2015
2. Cloud9 paper on testing, EuroSys 2011
3. Performance Problems you can Fix, OOPSLA paper
4. BugRedux paper on debugging, ICSE 2012
5. Finding bugs in Model checkers, FSE 2019 paper

Week 12: Presentation by students (Round 2)
Mon, 4 November 2019 - Fri, 8 November 2019

Presentation of research papers by students at classroom

Attendance is compulsory.

Papers presented:

1. Context-aware program repair, ICSE 2018
2. SketchFix paper on program repair, ICSE 2018
3. Probabilistic programming, FSE 2010 and follow-up
4. SQL synthesis paper, PLDI 2017

Week 13: In class examination

Mon, 11 November 2019 - Fri, 15 November 2019

An in class examination will be held. Attendance is compulsory.

File: 6217assessment - Answers.docx
Sample Answers

Reading Week

Sat, 16 November 2019 - Fri, 22 November 2019

Examination Week

Sat, 23 November 2019 - Sat, 7 December 2019