COM S/CPR E 513x/413x: Foundations and Applications of Program Analysis

Iowa State University

Spring 2018

Instructor: Wei Le (weile@iastate.edu), Atanasoff 210

1 Course Description

Algorithms and tools for automatically reasoning about code and program executions to predict software behavior. Theory and foundations related to control flow analysis, dataflow analysis, abstract interpretation and symbolic execution. Applications of program analysis to improve software security, performance and testing. Concepts, algorithms, tools, benchmarks, methodologies for solving problems using program analysis and for preparing research in program analysis.

2 Course Objectives

After successfully completing this course, students of COM S 413X are expected to:

1. know terminologies needed to read program analysis literature,
2. implement classical program analysis algorithms (you are encouraged to open source your implementation),
3. understand the connections between program analysis and other fields such as software engineering, security, programming languages, machine learning, artificial intelligence, natural language processing and statistics, and
4. apply program analysis tools to solve a formulated software problem.

After successfully completing this course, students of COM S/CPR E 513X are expected to:

1. know terminologies and mathematical frameworks needed to read program analysis literature,
2. learn and implement classical program analysis algorithms (you are encouraged to open source your implementation),
3. understand the connections between program analysis and other fields such as software engineering, security, programming languages, machine learning, artificial intelligence, natural language processing and statistics,
4. formulate software problems to program analysis, and
5. practice research methods in the area of program analysis

3 Prerequisites

- COM S 331: Theory of Computation
- COM S 342: Principles of Programming Languages

4 Textbooks and Other Resources

The course does not have a required text book. We will use lecture notes and papers to teach and learn, though the following books are considered classical to learn program analysis:

- *Principles of Program Analysis* by Chris Hankin, Flemming Nielson, and Hanne Riis Nielson, published by Springer, ISBN 9783662038116: a theoretical, static analysis book.
- *Advanced Compiler Design and Implementation* by Steven Muchnick, published by Morgan Kaufmann, ISBN 9781558603202: it is a compiler book that covers the topics of control flow analysis, dataflow analysis, alias analysis, and the applications of program analysis in compiler optimizations.

You are also welcomed to check out the program analysis courses taught by other instructors:

- Alex Aiken, Stanford, CS 357 Techniques for Program Analysis and Verification
- Monica Lam, Stanford, CS243 Program Analysis and Optimization
- Jonathan Aldrich, CMU, 15-819 O Program Analysis
- Jens Palsberg, UCLA, CS232 Static Program Analysis
- Mayur Naik, Georgia Tech, CS6340 Software Analysis and Testing
- Stephen Chong, Harvard, CS252r Advanced Topics in Programming Languages
- Mooly Sagiv, Tel Aviv University, Program Analysis
- Evan Chang, University of Colorado Boulder, CSCI7135 Program Analysis: Theory and Practice

5 Course Work for COM S 413x

- Survey (20%):
  - Learn how to write a survey by reading a given survey and submitting a summary (2%)
  - Read the given papers and answer questions related to papers (10%)
  - Writing 1-2 sections of the survey (8%)
- Assignments (50%): 5 sets of assignments (10% each) related to implementation of program analysis or using the tools of program analysis.
- Implementation project (30%):
  - Midpoint presentation
  - Project document, code of the analysis tools, test cases, output scree shot

6 Course Work for COM S 513x

- Survey (30%):
  - Learn how to write a survey by reading and presenting a given survey in class (3%)
  - Find a list of papers for the survey topic (3%)
  - Read the papers and answer questions related to papers (10%)
  - Write an outline of the survey, i.e., what sections we should have in the survey and what are the key points to include for each section (6%)
  - Writing sections of the survey (8%)
- Assignments (40%): 4 sets of homework on implementing program analysis algorithms and using program analysis tools
- Research project (30%):
  - Research proposal, formulate a software problem to program analysis research
  - Midpoint presentation
  - Research report

7 Tentative Topics

A typical class will consist of lecture and discussion time:

Lecture topics, including but not limited to:

1. Theoretical complexity of program analysis
2. Abstract interpretation: theory of abstraction, soundness and completeness
3. Control flow analysis: predicting execution paths
4. Data flow analysis: predicting data access patterns (definition, use, dependencies)
5. Chop, slice, dice: extension of data flow analysis
6. Taint analysis: extension of data flow analysis
7. Value flow analysis: predicting which expressions produce the same values
8. Pointer analysis: predicting which pointers points to the same memory location (value flow of the pointers)
9. Symbolic execution: representing program states using symbols (sometimes symbols mixed with concrete values)
10. Interprocedural analysis: scope of the analysis

Open questions to discuss, including but not limited to:

1. Which program analysis frameworks to use?
2. Analyzing binary level, source level or intermediate level of the code?
3. What problems are solved using program analysis?

4. What other techniques are used with program analysis?
5. What are the future directions of program analysis?
6. What are the mathematical problems the program analysis problems are reduced to? For example, points-to analysis is reduced to reachability on graphs.
7. What are the program analysis challenges for different programming languages, e.g., C, Python and Java?
8. A key spirit of static analysis – abstraction and approximation
9. What are the models and representations of programs?
10. What are the popular program analysis tools and why they are successful?
11. what are the similarities of different approaches that appear to be very unrelated approaches

8 Course Policies

Late homework: We do not grade late homework. Please submit your homework in time.

Academic dishonesty: Suspected academic misconduct will be reported to the dean of students office http://www.dso.iastate.edu/ja/academic/misconduct.html