Static Methods for Quantitative Program Analysis

Introduction

Winter Semester 2018/19; 10 October, 2018

Thomas Noll et al.
Software Modeling and Verification Group
RWTH Aachen University

`https://moves.rwth-aachen.de/teaching/ws-1819/qpa/`

Software Modeling and Verification Chair

RWTH AACHEN UNIVERSITY

Outline

What Is It All About?

Static (Program) Analysis

Static analysis is a general method for reasoning on artefacts such as requirements, design models, and programs.

Here, “static” means: based on source code, not on (dynamic) execution
(in contrast to testing, profiling, or run-time verification)

Overview

What Is It All About?

Static (Program) Analysis

Static analysis is a general method for reasoning on artefacts such as requirements, design models, and programs.

Here, “static” means: based on source code, not on (dynamic) execution
(in contrast to testing, profiling, or run-time verification)

(Main) Applications

Optimising compilers: exploit program properties to improve runtime or memory efficiency of generated code

- dead code elimination, constant propagation,...
- usually fully automated

Software validation: verify program correctness

- bytecode verification, shape analysis, functional correctness, ...
- varying degrees of automation

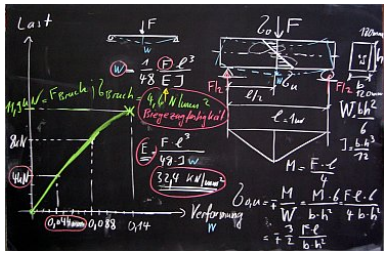
Overview

Dream of Program Analysis

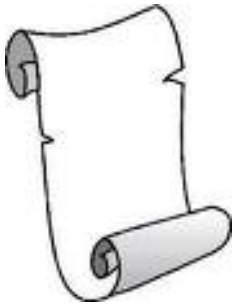
Program

```
                print "ncfiles: Socket error (%s) for host %s (%s)" % (errno,
for h3 in page.findAll("h3"):
    value = (h3.contents[0])
    if value != "Afdeling":
        print >> txt, value
        import codecs
        f = codecs.open("alle.txt", "r", encoding="utf-8")
        text = f.read()
        f.close()
        # open the file again for writing
        f = codecs.open("alle.txt", "w", encoding="utf-8")
        f.write(value+"\n")
        # write the original contents
```

Analyzer



Result



Property specification

Why “Quantitative”?

Classical setting

Addresses “yes/no problems”:

- Is control location c reachable?
- Is the value of variable x always positive?
- When we send a request to the server, will we eventually get an answer?

Why “Quantitative”?

Classical setting

Addresses “yes/no problems”:

- Is control location c reachable?
- Is the value of variable x always positive?
- When we send a request to the server, will we eventually get an answer?

Thomas A. Henzinger [Comp. Sc. – Research and Development 28(4), 2013]

“the Boolean partition of software into correct and incorrect programs falls short of the practical need to assess the behavior of software in a more nuanced fashion [...]”

Why “Quantitative”?

Classical setting

Addresses “yes/no problems”:

- Is control location *c* reachable?
- Is the value of variable *x* always positive?
- When we send a request to the server, will we eventually get an answer?

Thomas A. Henzinger [Comp. Sc. – Research and Development 28(4), 2013]

“the Boolean partition of software into correct and incorrect programs falls short of the practical need to assess the behavior of software in a more nuanced fashion [...]”

Interesting quantitative aspects

- Execution time
- Resource consumption (energy, heap space, ...)
- Probabilistic properties (reliability, expected execution time, ...)
- Note: many “qualitative” properties pose “quantitative” questions
  - e.g., balancedness of trees refers to height information

Outline

Aims of this Seminar

Goals

Aims of this seminar

- Independent understanding of a scientific topic
- Acquiring, reading and understanding scientific literature
  - given references sufficient in most cases
- Writing of your own report on this topic
  - far more that just a translation/rewording
  - usually an “extended subset” of paper
    - “subset”: present core ideas and omit too specific details (e.g., extension of partial to total correctness)
    - “extended”: more extensive explanations, examples, ...
    - discuss with supervisor!
- Oral presentation of your results
  - can be “proper subset of report”
  - generally: less (detailed) definitions/proofs and more examples

Requirements on Report

Your report

- Independent writing of a report of 10–15 pages
- First milestone: detailed outline
  - not: “1. Introduction/2. Main part/3. Conclusions”
  - but: overview of structure (section headers, main definitions/theorems) and initial part of main section (one page)
- Complete set of references to all consulted literature
- Correct citation of important literature
- Plagiarism: taking text blocks (from literature or web) without source indication causes immediate exclusion from this seminar
- Font size 12pt with “standard” page layout
- Language: German or English
- We expect the correct usage of spelling and grammar
  - $\geq$ 10 errors per page $\Longrightarrow$ abortion of correction
- LaTeX template will be made available on seminar web page

Requirements on Talk

Your talk

- Talk of 30 minutes
- Available: projector, presenter, [laptop]
- Focus your talk on the audience
- Descriptive slides:
  - $\leq$ 15 lines of text
  - use (base) colors in a useful manner
  - number your slides
- Language: German or English
- No spelling mistakes please!
- Finish in time. Overtime is bad
- Ask for questions
- Have backup slides ready for expected questions
- LaTeX/beamer template will be made available on seminar web page

Outline

Important Dates

Deadlines

- 5 November: Detailed outline of report due
- 3 December: Full report due
- 14 January: Presentation slides due
- 29 January (?): Seminar

Important Dates

Deadlines

- 5 November: Detailed outline of report due
- 3 December: Full report due
- 14 January: Presentation slides due
- 29 January (?): Seminar

Missing a deadline causes immediate exclusion from the seminar

Selecting Your Topic

Procedure

- You obtain(ed) a list of topics of this seminar.
- Indicate the preference of your topics (first, second, third).
- Return sheet here or by Sunday (14 October) via e-mail (`noll@cs.rwth-aachen.de`) or to secretary.
- We do our best to find an adequate topic-student assignment.
  - disclaimer: no guarantee for an optimal solution
- Assignment will be published on web site next week.

Important Dates

Selecting Your Topic

Procedure

- You obtain(ed) a list of topics of this seminar.
- Indicate the preference of your topics (first, second, third).
- Return sheet here or by Sunday (14 October) via e-mail (`noll@cs.rwth-aachen.de`) or to secretary.
- We do our best to find an adequate topic-student assignment.
  - disclaimer: no guarantee for an optimal solution
- Assignment will be published on web site next week.

Withdrawal

- You have up to three weeks to refrain from participating in this seminar.
- Later cancellation (by you or by us) causes a not passed for this seminar and reduces your (three) possibilities by one.

Outline

The Topics

Worst-Case Execution Time Analysis [Noll]

Goal

Analysis methods for establishing (safe and tight) timing constraints for (embedded) software, e.g. for ensuring schedulability

- challenge: execution times of instructions dependent on the execution history (caches, pipelines, speculative execution, ...)

The Topics

Worst-Case Execution Time Analysis [Noll]

Goal

Analysis methods for establishing (safe and tight) timing constraints for (embedded) software, e.g. for ensuring schedulability

- challenge: execution times of instructions dependent on the execution history (caches, pipelines, speculative execution, ...)

Topics

1. Timing Analysis by Integer Linear Programs [B]
2. Timing Analysis by Abstract Segment Trees [B]
3. Power-Aware Worst Case Execution Time Analysis [B]
4. Timing Analysis by Abstract Interpretation
5. Semantics-Based Worst Case Execution Time Analysis

The Topics

Heap Resource Analysis [Matheja]

Goal

Logical methods for reasoning about memory resources (variables, heap)

The Topics

Heap Resource Analysis [Matheja]

Goal

Logical methods for reasoning about memory resources (variables, heap)

Topics

6. Introduction to Separation Logic [B]
7. Reasoning about Complexity of Data Structure Operations [B]
8. Permission Accounting for Concurrent Threads [B]
9. Graph-Based Reasoning about Relational Properties of Data Structures [B]
10. Variables as Resources [B]
11. Dataflow Analysis for Quantitative Properties of Tree Data Structures

The Topics

Static Analysis of Probabilistic Programs [Kaminski]

Goal

Reasoning about (expected) values of quantities such as reliability, termination and sizes of data structures in the presence of probabilistic behaviour (e.g., failures)

The Topics

Static Analysis of Probabilistic Programs [Kaminski]

Goal

Reasoning about (expected) values of quantities such as reliability, termination and sizes of data structures in the presence of probabilistic behaviour (e.g., failures)

Topics

12. Quantitative Separation Logic [B]
13. Abstract Interpretation of a Probabilistic λ-Calculus [B]
14. Estimating Probabilities of Program Assertions [B]
15. Static Quantitative Reliability Analysis [B]
16. Probabilistic Invariants and Almost-Sure Termination [B]
17. Abstract Interpretation of Imperative Probabilistic Programs I
18. Abstract Interpretation of Imperative Probabilistic Programs II

Outline

Some Final Hints

Hints

- Take your time to understand your literature.
- Be proactive! Look for additional literature and information.
- Discuss the content of your report with other students.
- Be proactive! Contact your supervisor on time.
- Prepare the meeting(s) with your supervisor.
- Forget the idea that you can prepare a talk in a day or two.

Some Final Hints

Hints

- Take your time to understand your literature.
- Be proactive! Look for additional literature and information.
- Discuss the content of your report with other students.
- Be proactive! Contact your supervisor on time.
- Prepare the meeting(s) with your supervisor.
- Forget the idea that you can prepare a talk in a day or two.

We wish you success and look forward to an enjoyable and high-quality seminar!