Seminar *Probabilistic Programming*

Introduction

Winter Semester 2020/21; October 2020

Thomas Noll et al.

Software Modeling and Verification Group

RWTH Aachen University

https://moves.rwth-aachen.de/teaching.ws-20-21/propro/
Outline

Overview

Aims of this Seminar

Important Dates

The Topics

Final Hints
Probabilistic Programming

Probabilistic programs

Probabilistic programs = classical programs + probabilistic choice + conditioning

- choice: “execute program $P$ with probability $\frac{2}{3}$ and $Q$ with $\frac{1}{3}$”
- conditioning: “observe that value of variable $x$ is positive”
- describe posterior probability distributions over variable output values

Applications

- Randomised algorithms (e.g., randomised Quicksort)
- Computer vision (e.g., image generation)
- Security (cf. seminar)
- Biology, coding theory, cryptographic protocols, machine learning, quantum computing, reliability analysis, ...
An Example

Virus infection

```c
bool alicelNfectious = true
bool bobInfected = false
while alicelNfectious {
    prob 0.1 {
        bobInfected = true
    }
    prob 0.6 {
        alicelNfectious = false
    }
}
```

- What is the probability of Bob becoming infected?
- How long is Alice likely to be infectious?
## Areas Covered in this Seminar

### Topic areas

- **Semantics** [handbook]
  - mathematical approaches to formally define precise meaning of programs
- **Verification** [handbook]
  - providing correctness proofs for programs (termination, ...)
- **Logic** [handbook]
  - systematic inference for reasoning about probability distribution of program
- **Security** [handbook]
  - analysis of information leakage
- **Programming languages** [handbook]
  - new probabilistic and probabilistic extensions of classical programming languages
- **Static analysis** [papers]
  - analysis of reliability, resource consumption, ... based on source code
Outline

Overview

Aims of this Seminar

Important Dates

The Topics

Final Hints
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 than just a translation/rewording
  - usually an **“extended subset”** of original literature
    - “subset”: present core ideas and omit too specific details (e.g., related work)
    - “extended”: more extensive explanations, examples, ...
    - discuss contents 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 12–15 pages
- First milestone: detailed outline
  - rather: 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
  - \texttt{\LaTeX} template will be made available on seminar web page
- Language: German or English
- We expect the correct usage of spelling and grammar
  - $\geq$ 10 errors per page $\implies$ abortion of correction
Requirements on Talk

Your talk

- Talk of **30 minutes**
- Organised as Zoom meeting
- 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

Overview

Aims of this Seminar

Important Dates

The Topics

Final Hints
Important Dates

Deadlines

- 30.10.2020: Topic preferences due
- 30.11.2020: Detailed outline due
- 11.01.2021: Full report due
- 01.02.2021: Presentation slides due
- 08./09.02.2021 (?): Seminar talks

Important

Missing a deadline causes **immediate exclusion** from the seminar
Selecting Your Topic

Procedure

- Check out Foodle poll at https://terminplaner.dfn.de/qTBd2JXsC9i5fzGP
- Please give at least three “Yes” votes ✓
- Preferably additional “Maybe” votes (√)
- Topics classified according to BSc/MSc level
  - MSc students please choose at least one “M-only” topic
- Give as comment:
  - Preference of topics (if desired)
  - Language of report and talk (English/German)
- **Fill form by Friday, October 30**
- We do our best to find an adequate topic-student assignment
  - disclaimer: no guarantee for an optimal solution
- Assignment of topics and supervisors will be published on web site by mid-week 45

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.
Semantics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 1. | **Semantics of Probabilistic Programming** (B/M)  
   – semantics of an imperative language with discrete and continuous distributions  
   – operational: reduction (Markov chain)  
   – denotational: transformation of probability distributions |
| 2. | **Probabilistic Programs as Measures** (M)  
   – compositional measure-theoretic semantics for statistical modelling languages |
| 3. | **Computable Distributions** (M)  
   – semantics with focus on Type-2 computability (on infinite words, e.g., real numbers)  
   – encoding of computable distributions in functional programming language Haskell |
| 4. | **Probabilistic \( \lambda \)-Calculi** (B/M)  
   – probabilistic version of \( \lambda \)-calculus (core language for functional programming languages)  
   – two variations: randomised and Bayesian \( \lambda \)-calculi |
### Verification

5. **Probabilistic Couplings from Program Logics (M)**
   - use of couplings for verifying probabilistic programs
   - enables clean proofs of probabilistic relational properties (equivalence, convergence, ...)

6. **Expected Runtime Analysis (B/M)**
   - weakest precondition calculus (à la Dijkstra) for determining expected run-times of probabilistic program
   - fully automatable for Bayesian networks

7. **Termination Analysis (M)**
   - methods for checking (positive) almost-sure termination
   - based on martingales (special type of stochastic process)

8. **Quantitative Analysis (M)**
   - quantitative analysis of probabilistic programs
   - uses concentration of measure inequalities to characterize how functions of random variables deviate from expected value
9. *The Logical Essentials of Bayesian Reasoning* (B/M)
   - introduces channel-based approach to Bayesian probability theory
   - inspired by semantics of classical programming languages ("predicate transformer")

10. Quantitative Equational Reasoning (M)
    - extension of classical theory of equational reasoning to quantitative setting
    - involves some algebra and category theory
11. *Probabilistic Abstract Interpretation* (M)
   - application of abstract interpretation techniques to probabilistic programming in the context of security
   - yields estimation of knowledge an adversary can achieve by observations

12. *Quantitative Information Flow* (M)
   - presents an embedded domain-specific language in Haskell to compute hyper-distributions induced by programs
   - used to compute amount of information leakage of a program
Programming Languages

Programming languages

13. **Luck: A Probabilistic Language for Testing (B/M)**
   – probabilistic domain-specific language for test generation
   – framework for property-directed testing of functional programs

14. **Rely: Programming Unreliable Hardware (B/M)**
   – programming language for reasoning about the probability that a program produces the correct result when executed on unreliable hardware
   – application: approximate computing

15. **Tabular: Probabilistic Inference from the Spreadsheet (B/M)**
   – domain-specific programming language for expressing probabilistic models and performing probabilistic inference over relational data
   – programs and data stored as spreadsheet tables (MS Excel interface)
# Static Analysis

## Static analysis

16. *Reduction Methods for Reliability Analysis* (B/M)
   - goal: combat state-space explosion of probabilistic programs
   - introduces two reduction methods that operate on syntactic level

17. *Slicing* (B/M)
   - goal: identify part of program that affects “interesting” variables
   - takes control/data/observe dependences into account

18. *An Algebraic Framework for Static Analysis* (M)
   - framework for designing, implementing, and proving correctness of static analyses
   - supports probabilistic programs with recursion, unstructured control-flow, nondeterminism, and continuous distributions

19. *Resource Analysis* (B/M)
   - static analysis for deriving upper bounds on expected resource consumption of probabilistic programs
   - derives symbolic bounds represented as multivariate polynomials in inputs
Outline

Overview

Aims of this Seminar

Important Dates

The Topics

Final Hints
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!