Advisor	Bas Spitters
Students	Kira Kutscher
Languages	English
Text tools	LaTeX
Other tools	Coq

Project Description

We selected the basis for our work to consist of $\mathcal{R}$`ml`, presented by Audebaud and Paulin-Mohring in 2009 ("Proofs of randomized algorithms in Coq") as well as the `xhl` development[1], which is a Coq implementation of `pwhile`, the probabilistic imperative language used in EasyCrypt.

Our project explores similarities as well as differences between $\mathcal{R}$`ml` and `pwhile` and understanding their respective interpretations. Our aim for this project is to understand how we can translate a program, P, written in `pwhile` to a semantically equivalent one in $\mathcal{R}$`ml` and show that the results of interpreting P directly will lead to the same result as interpreting the $\mathcal{R}$`ml`-translation of P.

Along the way we will attempt to implement an interpretation of $\mathcal{R}$`ml` in Coq. If we are left with more time on our hands before the completion of this project, we will also attempt a formalisation of the translation of `pwhile` to our implementation of $\mathcal{R}$`ml` .

The goal of this project is to present some different approaches to the construction and interpretation of probabilistic programs and show their equivalence. The reader should be left with an actionable understanding of the domain theoretic approach, enabling them to start working with the basic concepts of the topic.

Provisional Table of Contents

- Abstract (10-20 lines)
- Section 1: Introduction (1-2 pages)
- Section 2: Theory and existing frameworks (6-12 pages)
- Section 3: Our approach (6-12 pages)
- Section 4: Our contribution (6-10 pages)
- Section 5: Comparison to other work and ideas for future work (2-4 pages)
- Section 6: Conclusion (1-2 pages)
- Acknowledgements (3-5 lines)
- References ($\frac{1}{2}$-1 page)
- Appendix with programming code, tables, full proofs, etc. (5-20 pages)

Provisional Time Plan

First week of February (15 hours)
Planning of activities, including the production of the Bachelor's contract.

[1] `https://github.com/strub/xhl`

Rest of February and first half of March (3×15 hours)
Read literature and research the current state of the art tools for formal cryptographic proofs; rough draft of Section 2 in Bachelor's report.

Rest of March until start of May ($2 \times 15 + 8 \times 30$ hours)
Work on the design and implementation of our contribution. Keep adding to the report along the way.

End of May and first half of June (3×30 hours)
Write the missing parts, put drafts together, make things consistent, proof reading.

31st of May
Have a draft of the section about our own implementation. It should contain all of the structure needed, but does not have to be clean yet.

3rd of June
The sections on theory as well as our own implementation should be complete modulo proof reading.

10th of June
Sections 3 and 5 should be done modulo proof reading.

12th of June
Introduction, conclusion and abstracts (one in Danish, one in English) should be in place. The rest of the time is spent on proof reading.