BSc thesis

Zeimer

March 2019

Abstract

This is a simple paragraph at the beginning of the document. A brief introduction to the main subject.

Contents

1 TODO

1. Abstract.
2. Introduction: functional programming, formally verified programming and proving.
3. Approaches to computational effects: chaos, ML-style, monads, algebraic effects.
4. A description of the inner workings of the library: design choices, file structure, implementation.
5. Examples: some from Just Do It, maybe some custom ones.
6. Safety: some theorems and proofs.
7. Theoretical comparison of the ease of use with Haskell and Idris.
8. Practical comparison with MERC.
9. Cite some literature: some Coq papers, Moggi, Just Do It, Experimenting with Monadic Equational Reasoning in Coq
10. Technical matters:
    (a) Mention where's the implementation and put it to Coq's repository of user libraries.
    (b) Installation guide.
    (c) Tools: why no ssreflect?
    (d) Documentation (it's in the source code).
11. More: a case study in proof engineering - how do the tactics hs, monad and (maybe) the one for reflective functor simplifcation work?
12. Deficiencies, conclusion and further work.
13. Points to make: this is a library for general purpose programming, without some deep goal.

2 Introduction

This is the first section.

Unnumbered Section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisissem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi necante...

3 Second Section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisissem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi necante...