

DISSERTATION PROPOSAL

BENJAMIN BERMAN

1. INTRODUCTION

A long time ago my cello teacher pointed out that the way to play a difficult bit of music is not simply to grit one's teeth and keep practicing but to figure out how to make playing those notes easy. The major goal of the proposed dissertation is to reapply the same idea in the context of interactive theorem proving with Coq: I intend to show ways to make the difficult task of using Coq easier by improving the user interface. As well as solving usability problems for an important and powerful tool, many of the techniques I am developing and testing are widely applicable to other forms of coding.

Why is this work important to society?

Why is this work an intellectual contribution?

Why is this work doable?

I would like to point out that this task of improving user interfaces for theorem provers is a natural extension of work on symbolic logic and of proof assistants. Symbolic logic can make working with statements easier. Proof assistants can make working with symbolic logic easier. User interfaces can make working with proof assistants easier. Though the techniques differ, the goals are shared.

2. IDEAS

2.1. jEdit as an Interactive Theorem Proving Environment. Harness the power of Java/Piccolo/jEdit. jEdit also used with Isabelle (CITE).

2.2. Proof Previews.

2.3. Proof Tree Visualization.

2.4. Proof Transitions.

2.5. Syntax Tree Visualization.

2.6. Keyboard-Card Menus.

3. IMPLEMENTATION

3.1. Current.

3.2. Planned.

4. DESCRIPTION OF USER STUDIES

4.1. **Keyboard-Card Menu Study Results.**

5. LITERATURE REVIEW

6. TIMELINE

7. CONCLUSION

I hope to have made several points in this proposal. First, that this is important work, both because the Coq interactive theorem prover is an important tool that could benefit significantly from improved user interfaces and because many of the ideas generalize to other forms of coding. Second, that as an intellectual challenge this work is non-trivial, not only because of the normal programming problems that must be overcome but because designing good user interfaces for complicated systems, which includes the identification of tractable problems and the testing of potential solutions, is non-trivial. Finally, that, despite this non-trivial nature, the work can be accomplished.