

# Project Proposal

Dylan Orris

September 18, 2018

## 1 Focus

The focus of this project is the production of a software which will take a mathematical query and return the truth value of the statement. This will be done through processing of the input, which will come in as plain English. This input will then be converted to some symbolic language, which then will have logical operations applied to it in an attempt to prove or disprove the claim.

## 2 Objectives

The theoretical portions of the IS will relate to Natural Language Processing, Automated Theorem Proving, and Multithreading methodologies. This will include analysis of how to do each process, along with potential alternatives and why they were not selected for this project.

This project will produce a command line software which prompts the user for a string. If this string is a valid mathematical query, the software will use a Prolog database to prove or disprove the query. If time permits, the software will ideal return a  $\text{\LaTeX}$ document along with the truth value of the query. This document would provide a step-by-step description of the proof used by the software, allowing the user to follow what occurred.

This project will give me experience both with Natural Language Processing and Automated Theorem proving. Part of this experience will be learning the languages Prolog and Perl, as well as hopefully learning how to effectively multithread programs for greater performance.

## 3 Reaching Objectives

The primary efforts for reaching objectives will come in the form of learning Prolog and Perl. As these are two languages I have never used before, some of the struggles will simply be learning how to perform desired functions within the language, and how to properly save data in variables. Additionally, Prolog is a logical programming language, which is a type of language I have no prior experience with.

## 4 Possible Issues

One major issue for this project, as discussed in the prior section, will be learning to work in Prolog and Perl. When these issues are ones that I am unable to solve on my own, I plan to turn to online communities for assistance. There is no reason to expect that these languages will prove insurmountable.

A further issue for the project may be its size. While the software will ideally consist of both a homemade NLP kit and a homemade automated theorem prover, it may prove necessary to use a prebuilt library for one of these goals, if time does not permit. The current expectation is to have the NLP kit produced first, and when it works well, allow it to produce symbolic notation in any language so that it may be used with any prebuilt system. Should it prove unfeasible to produce both parts of this project, a Perl based theorem proving library will be sought out.

## 5 Timeline

### Semester 1

- Week 6 Annotated Bibliography
- Week 7 Outline
- Week 8 Working NLP for conversion to any symbolic language
- Week 10 Multithreaded NLP
- Week 12 Chapter 1
- Week 14 Basic Automated Theorem Prover for simple claims

### Semester 2

- Week 1 Better Prover and Chapter 2
- Week 3 Multithreaded Prover
- Week 5 Final Software
- Week 7 Chapters 3 & 4
- Week 9 Chapter 5 and final edits

## References