Journal 0

Rahul Mittal

August 27 2019

1 Introduction

Mathematicians write combinatorial proofs, but they may make mistakes so we don't know if the proofs are correct. Researchers have developed natural language processing (NLP) libraries and techniques which can parse and organize natural language text; researchers have also developed formal proof languages in which we can express mathematical proofs and have them be computationally verified. My project aims to use NLP to convert combinatorial proofs written in English into a formal proof language so that they can be verified computationally.

2 Summer

I didn't do anything.

3 Obstacles

One of the challenges I think I will face is dealing with the variety of sentence structure as well as different ways words can be used in English. I'm not fully sure of the extent to which the Stanford parser (which I intend to use) deals with the first problem; the second problem will most likely have to be dealt with on my end somehow. I also envision that it's going to be difficult to deal with the amount of words that have specific meaning (as opposed to say words like "balls" and "urns" which are being used as arbitrary objects), how to detect the difference between those two types of words, how to organize the former and deal with special cases.

4 1st Marker of Success

My first marker of success will be for my program to be able to understand the actions being performed in the different parts of a proof (defining, making a claim, etc. like I said in the timeline). I'm fairly confident that if I am able to reach that point then what I have to do after that should be possible with similar effort by building on top of the structure generated by my program in the first step.

5 Materials

Since my project is entirely programming, there aren't any physical materials that I need. My project will use libraries at least including the Stanford parser, and I will also need to install the software necessary to compile and run code written in Coq (ideally there also exists some kind of IDE or editor I can use to more easily write code in Coq).