Social Dynamics Used for Verification of Large Proof Trees.

Jianchi Chen Ying-Yu Ho Tim Holland Kexin Rong
Qianhaozhe You
California Institute of Technology

Abstract

A mathematical proof has the structure of a DAG. However, it is typically presented in a linear fashion, and cannot practically expose the depth of its roots. We propose a web service to provide both of these by exposing part of the graph to the user and allowing them to explore down into the subproofs. Naturally it would be prohibitive for the authors to write even a useful amount of the tree in, so the service allows the introduction of new proofs to expand the reach of the proof trees. Specification of proofs could either be very formal, in which case a program could verify them or more relaxed and reviewed by other users in a social effort to find mathematical truth.

1 Introduction

We propose a web application that allows the inspection of, addition to, and potentially input of verification of collection of large proof DAGs. Interfacing with a node of the tree will present a statement of the theorem associated with that node, edges to the children and potentially to some of its parents. Text accompanies the node, and explains how the children are combined together to prove the theorem. Ideally the accompanying text is as small as possible and the nodes have a small fan out to break the proof down into digestible chunks.

We have two possible choices to proceed on how to keep the DAG logically sound. The first is more of a technical problem: require highly precise statements of additions to the tree to be parsed, the logical statements can then be verified (or rejected) automatically. Alternatively, with a more social emphasis proofs could be stated in natural mathematical language. Under this scheme, the correctness would have to be verified by users, through a combination of moderation, reputation, and democracy.

2 Technical Hurdles

At the very basic level we have to deal with all the usual problems in building a webapp: deciding upon tools (languages, databases, frameworks) and an API to communicate between client and server. Further features needed would be dynamic loading of the graph as the user scrolls through, search to look for other proofs to minimize redundancy, automated theorem verification or a voting mechanism, ability to mention other theorems to include them is lemmas in a particular proof, text entry for a theorem, inclusion of images for proofs that necessitate it, and collections of axioms for the various fields of mathematics that could be present as a starting point. Hard problems would involve the choice of data structure for the proofs, intuitive presentation of the DAG to the user, an effective mechanism of rejecting poorly worded or incorrect proofs.

3 Potential Order of Implementation

1. Persistent storage of a proof in the database.
2. Communication of a proof over the network.
3. User interface.
4. Adding a new theorem from the user interface.

5. Persistence of theorems in UI across page loads.
6. Verification system (social or formal).

4 Background

I'm not aware of any similar projects.

5 End Product

The end product would hopefully be a useful web application that would allow publishing of proofs in a graphical format subject to immediate review of other users, and a repository for a large collection of proofs to be seen in a graphical format upon request.