Proof editor for structural induction

Keywords

proof editor, inductive sets, structural induction, logic

Background

In mathematics, simply induction is a very powerful tool that allows us to prove properties over the infinite set of natural numbers. Example of such a property is that

$\forall\, n \geq 1.\ 2 + 2^2 + 2^3 + 2^4 + \ldots + 2^n = 2^{(n+1)} - 2.$

In computer science, recursive data types such as lists or trees play a very important role in programming. When we want to prove properties over (finite) elements of such data types, structural induction is the right tool to use. Example of a property that can be proved by structural induction over the structure of a list could be that

$\forall$ xs ys. len (xs ++ ys) = len (xs) + len (ys)

for any lists xs and ys, where len is the function computing the length of a list and ++ is the function computing the concatenation of two lists.

Project description

Every inductively defined set has an associated inductive principle that can be used to prove properties over the elements of the set. The induction principle associated with an inductive set follows the structure of the set. Proofs using this principle are said to be done by structural induction.

The set of Natural numbers is such an inductive set and its induction principle is known as simple induction (a.k.a. mathematical induction). However, any other inductive set such as lists or trees have also associated induction principles.

This project is about constructing a proof editor that allows to prove properties over arbitrarily defined inductive set. This in turn requires that the proof editor deals with:

- first order logic
- incomplete proofs
- inductive definitions of sets
- recursive definitions over elements of inductive sets
- graphical presentation of proofs
- error messages when an incorrect step is attempted

Suggested reading:
https://en.wikipedia.org/wiki/First-order_logic
https://en.wikipedia.org/wiki/Natural_deduction
https://en.wikipedia.org/wiki/Structural_induction

https://chalmers.instructure.com/courses/15028/files/folder/Recap%20Reading?preview=1541559

Target group

DV, IT, D, TM, F
but even other students interested in the subject can take part

Special Pre-requisites

Students should be familiar with basic concepts in logic and (simple) induction at the level taught in a discrete mathematics course. Good programming skills are necessary.

It is an advantage, but not necessary, if students are familiar with functional programming, structural induction (for example via the course on Finite automata and formal languages) and more advance logic (for example via the course Logic in Computer science).

Proposal author

Ana Bove (bove@chalmsers.se)

Supervisor

Possibly Ana Bove. Could be others in the ProgLog group/LT unit.