NAME
Zsuzsanna Sinka
(Hungary)

ADVISOR
O.Univ.-Prof. Dr.Dr.h.c.mult.
Bruno Buchberger

INSTITUTE
Research Institute for Symbolic Computation

UNIVERSITY OF APPLIED SCIENCES OBERÖSTERREICH
JOHANNES KEPLER UNIVERSITY LINZ
softwarepark hagenberg
ISI HAGENBERG International Studies in Informatics

Interactive Tools for Inspecting Proofs in Theorema

Automated theorem proving · Theorema · Interactive proving
Focus windows · Rule inspector · Research

This thesis is the result of the author's work with the Theorema theorem proving system. We want to aid mathematics and computer science students in studying logic.
The tool we presented here is designed to check the user's understanding of the inference rules during a (logical) proof. On the other hand, it can be also used to check the correctness of an automatically generated proof.
The application of our work is illustrated by examples of increasing difficulty.

Introduction

Nowadays, logic is conquering a more and more important position.This universal language provides an opportunity to share knowledge without being able to speak a common natural language.Those who speak' logic' can formulate the whole world of mathematics, although learning logic is not so easy.

The Theorema system already provides tools for the inspection of proofs generated by (the Theorema) provers.

In this work, we illustrate tools for training predicate logic proving with mathematics and computer scientist students. In addition, we implement and describe a slight extension of one of the Theorema proof inspection tools. This extension allows the student to choose at each proof step, from a library of inference rules, the appropriate inference rule, which then is compared with what the automated prover was doing. Several case studies are presented in detail. Also, for beginning students who did not have any exposure to proving and the Theorema system, an easy-to-read manual is presented.

The Theorema project, initiated and directed by Bruno Buchberger, aims at supporting the entire mathematical exploration cycle including the proving phase. Built on top of Mathematica, it implements a two-dimensional syntax of mathematical formulae similar to the syntax used by mathematicians, and a formal text language for defining new notions, properties of notions, problems, and algorithms, and for combining mathematical formulae in knowledge bases and for using it in proofs. The system contains various general provers (which can be used for generating proofs in arbitrary mathematical theories) and special provers (which can be used for generating proofs in specific theories).

In the default setting, a Theorema reasoner tries to solve a given reasoning problem automatically by repeated inference rule application, until either a successful proof is obtained or no more inference rules can be applied. An inference rule takes as input a reasoning situation consisting of a reasoning goal and a knowledge base and returns a new reasoning situation.

The RuleInspector is an extension of the Focus Windows technique. We designed it to help check the user's understanding of the inference rules of the prover being studied. It can be also used to check the correctness of an automatically generated proof. By calling a proof, another window appears besides the classical Focus Window, the Inference Rule Library. This is a collection of the possible inference rules of the prover used for proving. The user can click on the name of the inference rule in order to check if it was used in the current proof step or not. The RuleInspector uses the two-phase style of the Focus Window but it has additional features depending on its style (strict or free).

In a mathematical proof it is usually not written which proof rule was applied in a certain proof step. Although the ones who created the proof know exactly which one of the rules are necessary, for a new person on the field it takes time to acquire the methods, to get used to use them and to learn the small tricks. We tried to reduce this time to gain a quicker taste of the proving methodology. The RuleInspector was created to investigate the proofs, although it is also capable to help debugging the functioning of the provers.

The inspection has as many steps as the proof has. One can follow the proof steps in the Focus Window and one should choose from the list of inference rules the one used in the current step. The Focus Window presentation shows two phases for each proof step. The 'Attention Window' shows the user the formulae that have been utilized by the applied rule. The 'Transformation Window' adds to this any formulae that have been inferred by the inference rule applied.

Independent of the two focus windows phases, the rule inspector can be used in two modes: free - the default one - and strict. By default, a user can freely navigate through the proof, without necessarily choosing an inference rule in the Inference Rule Window. In the strict mode reaching a situation in a Transformation Window the user cannot continue navigating through the proof until the inference rule actually used is picked from the inference rule window. In the strict inspection mode the navigation buttons are deactivated.