Comparison of theorem provers

Artem Yushkovskiy
artem.yushkovskiy@aalto.fi

Tutor: Stavros Tripakis

Abstract

One of the useful applications mathematical logic theory is the Automated theorem proving. This is a set of techniques that allow one to verify mathematical statements mechanically using logical reasoning. Although, it can be also used to solve engineering problems, for instance to prove security properties for a software system or an algorithm. Furthermore, automated theorem proving is an essential part of the Artificial Intelligence theory, which became highly evolving these days. In this paper, we describe bases of formal systems and automated deduction theory, and compare two widespread tools for automated theorem proving, Coq [1] and Isabelle [2].

KEYWORDS: logic, automated theorem prover, Coq, Isabelle

1 Introduction

In general terms, *formal proof* is the sequence of statements, based on finite set of fundamental axioms and satisfying the rules of logical inference. *The axiom* is a statement claimed to be true evidently. *The logical inference* is the transfer from one statement (premise) to another (consequence), which preserve truth, while the rule of logical inference is a principle that allows one to infer the validity of such transfer. In formal logic, inference is based entirely on the structure (i.e., form) of those statements, which allows

one to apply basic logical rules to any type of proof and thus construct the formal system.

The main goal of the formal system is to be verifiable, i.e. one could *check* its validity. At present, a lot of tools are being developing to automate the process of such checking to run it on the computer. In particular, the systems *Isabelle/ZF* [???], *Coq* [???], PVS [???], ACL [???] work in a form of axiomatic set theory and allow the user to enter theorems and proofs into the computer, which then verifies that the proof is correct (these are also called sometimes "proof assistants"). Another goal of constructing the formal system is having the computer to *discover* formal proof. This goal is different from the previous one since the system must be optimised for efficient search. The output proofs can rely on induction, or on meta argument, or on higher-order logic. McCune's systems *Otter* [???] and *Prover9* [???] are commonly recognized as the state-of-the-art tools [Com00].

In current paper we consider only the systems, which are built to achieve the first goal, i.e. to verify existing proof, since <...>. Two aforementioned theorem provers Coq and Isabelle are examined for the purpose of revealing expressiveness, computation power and usability. These properties are described in detail in section <???>. Section 2 gives an overview of the history of logic providing thorough definitions, typology and properties of formal system and formal proof. Issues related to theoretical limitations of formal systems are discussed further as well. <... about comparison, results and author's personal contrubution>

2 Theory of logical calculi

// TODO

set theory, Zermelo–Fraenkel, briefly 1st and 2nd ordered logic,

an example from wiki: A formal system or logical calculus is any well-defined system of abstract thought based on the model of mathematics. A formal system need not be mathematical as such; for example, Spinoza's Ethics imitates the form of Euclid's Elements. Spinoza employed Euclidiean elements such as "axioms" or "primitive truths", rules of inferences etc. so that a calculus can be built using these. For nature of such primitive truths, one can consult Tarski's "Concept of truth for a formalized language".

2.1 History of logic calculi

// TODO

A long time ago in a galaxy far, far away...

Hilbert's programme, ...

Mock text from wiki:

- The embryonic period from Leibniz to 1847, when the notion of a logical calculus was discussed and developed, particularly by Leibniz, but no schools were formed, and isolated periodic attempts were abandoned or went unnoticed.

- The algebraic period from Boole's Analysis to Schröder's Vorlesungen. In this period, there were more practitioners, and a greater continuity of development.

- The logicist period from the Begriffsschrift of Frege to the Principia Mathematica of Russell and Whitehead. The aim of the "logicist school" was to incorporate the logic of all mathematical and scientific discourse in a single unified system which, taking as a fundamental principle that all mathematical truths are logical, did not accept any non-logical terminology. The major logicists were Frege, Russell, and the early Wittgenstein.[92] It culminates with the Principia, an important work which includes a thorough examination and attempted solution of the antinomies which had been an obstacle to earlier progress.

- The metamathematical period from 1910 to the 1930s, which saw the development of metalogic, in the finitist system of Hilbert, and the non-finitist system of Löwenheim and Skolem, the combination of logic and metalogic in the work of Gödel and Tarski. Gödel's incompleteness theorem of 1931 was one of the greatest achievements in the history of logic. Later in the 1930s, Gödel developed the notion of set-theoretic constructibility. The period after World War II, when mathematical logic branched into four inter-related but separate areas of research: model theory, proof theory, computability theory, and set theory, and its ideas and methods began to influence philosophy.

2.2 Limitations of logic systems

// TODO
first-order logic

2.3 Emphasizing text

Italics is a good way to emphasize printed text. However, **boldface** looks better when converted to HTML.

Paragraphs are separated by an empty line in the Latex source code. Latex puts extra space between sentences, which you must suppress after a period that does not end a sentence, e.g. after this acronym.

Cross-references to figures (Fig. 1), tables (Table 1), other sections (Sec. 2.4) are easy to

create.

2.4 Mathematics

In the mathematics mode, you can have subscripts such as K_{master} and superscripts like 2^x. Longer formulas may be put on a separate line:

$$\emptyset \in \emptyset \ \Rightarrow \ E \neq mc^2.$$

You may also want to number the formulas like Eq. (1) below.

$$C = E_{K_{public}}(P) = P^e. \qquad P = D_{K_{private}}(C) = C^d. \tag{1}$$

2.5 Make a list

Lists can have either bullets or numbers on them.

- one item
- another item, which is an exceptionally long one for an item and consequently continues on the next line.

  Lists can have several levels. Item 1 below contains another list.

1. the fist item
   (a) the first subitem
   (b) the second subitem
2. the second item

3 More complex stuff

This section provides examples of more complex things.

3.1 Data served on a table

Table 1 presents some data in tabular form.

Protocol	Year	RFC
TCP	1981	793
ISAKMP	1998	2408
Photuris	1999	2522

Table 1. A table with some protocols

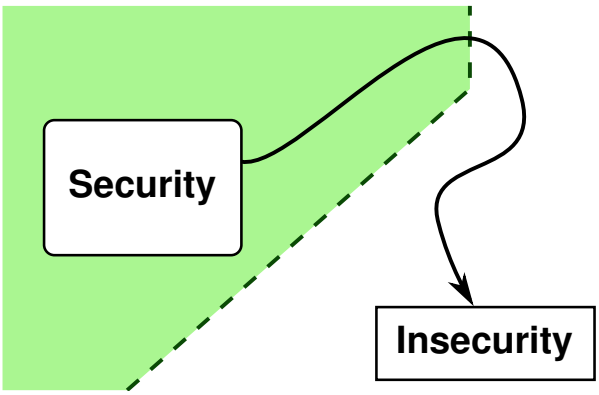

Figure 1. An embedded picture

3.2 Adding references

Do not forget to give pointers to the literature. If you are listing stuff related to your topic, you can give several references once [1, 5, 3]. However, usually you should give only one, for example the standard describing the stuff [2] and if you want to directly use someone else's words, use both quotation marks and refer to the source, for example that "the developer does not need to know all about the framework to develop a working implementation" [4]. Remember also to mark references to your pictures if they are not created by your own mind!

If you plan to write with Latex regularly, create your own BibTeX database and use BibTeX to typeset the bibliographies automatically. In the long run, it will save you a lot of time and effort compared to compiling reference lists by hand.

3.3 Embedded pictures

Fig. 1 is an embedded picture. The supported formats for pictures depend on the actual LaTeX command used. For instance, regular LaTeXsupports pictures in EPS (Embedded PostScript) format, while pdfLaTeXsupports PDF (Portable Document Format), PNG (Portable Network Graphics) and JPEG (Joint Photographic Experts Group). It is recommended to use either EPS or PDF for diagrams as well as for any picture which includes vector images.

4 Yet another section title

To be added.

5 Conclusion

To be added.

References

[1] Douglas E. Comer. *Internetworking with TCP/IP, Volume I*. Prentice-Hall Inc, 4th edition, 2000.

[2] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol (ISAKMP). RFC 2408, The Internet Engineering Task Force, November 1998. http://ietf.org/rfc/rfc2408.txt.

[3] Pekka Nikander. *An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems*. PhD thesis, Helsinki University of Technology, March 1999.

[4] Sanna Suoranta. An Object-Oriented Implementation of an Authentication Protocol. Master's thesis, Helsinki University of Technology, November 1998.

[5] Tim Hsin-ting Hu and Binh Thai and Aruna Seneviratne. Supporting Mobile Devices in Gnutella File Sharing Network with Mobile Agents. In *IEEE International Symposium on Computers and Communication*, volume 3, pages 1530–1346, April 2003.