

Fine-grained mathematical justifications

Jesse Alama

Center for Artificial Intelligence, New University of Lisbon

2011-07-21

Introduction

- Interest: representing mathematical knowledge formally and computing with it.
- Aim: discover necessary and sufficient conditions for (formalized) mathematical theorems
- Method: produce fine-grained dependency information from the **mizar** Mathematical Library



Automated Theorem Proving

- The task of an automated theorem prover (ATP) is to find a deduction of a given proposition from given assumptions.
 - Or: show that a given formula is unsatisfiable (and possibly produce a proof that it is so);
 - Or: show that a given formula is satisfiable (and possibly produce a description of a model of the formula).



Interactive theorem proving

- We are often interested in proving theorems that are many steps away from our basic assumptions.
- ATPs are generally not well-suited to such tasks, because the search space of deductions is often (extremely!) large and complex.
- Interactive theorem provers (ITPs) focus the reasoning task to the construction of proofs.
 - Proof formalisms: — natural deduction (in various styles: J askowski, Fitch, Gentzen, etc.)
 - sequent calculus
- ITPs generally use an ATP in some fashion to check the the most basic steps of arguments expressed in the ITP's proof language.



What's the difference?

- ITPs are generally used to construct **human-readable** proofs.
- The proof formalisms employed by ATPs often produce proofs that take unexpected, counterintuitive paths from the assumptions to the desired conclusion.
 - Analyzing proofs found by ATPs is usually nettlesome.
- One writes a proof with the assistance of an ITP by breaking down the argument of interest, and then querying the ITP to check whether the proof is acceptable.



A zoo of ITPs

Numerous ITPs in various styles are available. Some major ones:

- Isabelle: based on a weak, extensible logic that admits many possible ‘instantiations’ (e.g., classical and intuitionistic logic, higher-order logic vs. set theory, etc.), tactic or natural deduction proof style
- HOL, HOL light: higher-order logic based on the so-called LCF (logic for computable functions) style, λ -calculus, intuitionistic logic (via the Curry-Howard correspondence), tactic proof style
- Coq: based on the calculus of inductive constructions, tactic proof style
- Mizar: first-order classical set theory, natural deduction



Focus on mizar

- **mizar** is one of the oldest ITP that enjoys widespread use.
- Its foundations—classical first-order set theory—are the most attractive from the perspective of traditional foundations of mathematics.
 - **mizar** is actually based on Tarski-Grothendieck set theory, which is stronger than ZFC.
- Rich formalism: captures many natural aspects of mathematical practice, such as various kinds of notational conventions and extensions by definitions.
- Moreover, among comparable ITPs, the lion's share of the knowledge formalized in the mizar system is based on **pure mathematics**.



Aim: computing precise necessary and sufficient conditions

- From a formalized proof of a theorem, we can say, with complete precision, what is **sufficient** for the theorem.
- We can also compute refinements to a theorem, thereby giving a sharper set of sufficient conditions.
- When the set S of sufficient conditions is suitably refined, we can attempt to go backwards and find out whether S is actually necessary for the theorem.



Dependency graph

- **Nodes:** mizar items (theorems, definitions, lemmas, implicit information, etc.)
- **Edges:** The set of outgoing edges of a node u is the set of nodes $\{v_1, v_2, \dots, v_n\}$ such that the justification of u depends on all of v_1, v_2, \dots, v_n .

The dependency graph of a set of deductions gives information about what is sufficient for the success of theorems.



Philosophical aside: intensional vs. extensional mathematical knowledge

- Intension: the method of dependency graphs can be used to say something about new proofs **in mizar**:
 - Assumptions used in a proof of a mathematical theorem **as formalized in mizar**;
 - Necessary conditions for the success of a particular **mizar** text.
- Extension: claims about mathematical theorems that do not depend on how they are expressed/formalized.
 - Evidently something more is needed to answer extensional questions than working with a **mizar** dependency graph.
 - Candidates methods: 'thinking hard'/traditional mathematics, automated theorem provers.

Refining proofs from dependency graphs

(example)

- Every path commencing with theorem T ending at axiom A passes through lemma L .
- Thus, $S - \{A\} + L \vdash T$.
- If we know that $S - \{A\} + L$ differs from S , by our calculation we have found a weakening of the sufficient conditions for theorem T .



A challenge

- 2008: my formalization of Euler's polyhedron formula (" $V - E + F = 2$ ")
- The **mizar** formalization shows that **TG** \vdash Euler's polyhedron formula, so that **TG** is sufficient for (a formalization of a particular proof of) this theorem.
- Challenge: Tarski-Grothendieck set theory is clearly much more than what is needed for Euler's formula. What do we really need?
 - **Use the text.** Calculemus!



Tarski's universe axiom

- For every set a there exists a set U such that:
 - $a \in U$,
 - U is closed under taking power sets,
 - U is closed under union,
 - for every subset A of U for which $|A| < U$, we have that $A \in U$
- Consequences of the universe axiom:
 - axiom of infinity
 - axiom of choice
 - power set

'Eliminating' the universe axiom

- By analyzing the dependency graph for Euler's polyhedron formula, one finds that all paths from the theorem to the universe axiom go through either:
 - axiom of infinity
 - axiom of choice
 - power set
- This calculation shows that, from a witness to $\mathbf{TG} \vdash \text{EPF}$, we have a witness for $\mathbf{ZFC} \vdash \text{EPF}$, which is a considerable refinement, in light of the strength of \mathbf{TG} .

'Eliminating choice': from ZFC to ZF

- We can do even better, using the dependency graph of Euler's formula, but using a slightly different method:
- All paths from Euler's formula to the axiom of choice pass through the theorem "Every linearly independent set of vectors in a vector space can be extended to a basis".
- But the proof of Euler's formula uses only finite-dimensional vector spaces, which always have bases (by definition).
- Hence, choice can be eliminated, and we have that **ZF** \vdash Euler's polyhedron formula.

'Eliminating' replacement: from **ZF** to **Z**

- **Challenge:** Gather all instances of the scheme of replacement (the characteristic axiom that distinguishes **Z** from **ZF**) that occur anywhere in the complete justification of Euler's polyhedron formula.
- Show, for each such instance, that it is a theorem of **Z**.
- (Such a proof could be carried out either 'by hand' or with the help of an automated theorem prover.)

Necessary conditions

- Reverse mathematics: discover axioms from theorems (rather than the other way around).
- Example: Bolzano-Weierstrass theorem (every bounded sequence of real numbers has a convergent subsequence).
 - the theorem can be formally proved in **ACA**₀ (**RCA**₀ + scheme of arithmetical comprehension)
 - show that, working in the much weaker **RCA**₀, that every instance of the scheme of arithmetical comprehension can be proved from the Bolzano-Weierstrass theorem.
 - Conclusion: **ACA**₀ is equivalent to, or necessary and sufficient for, the Bolzano-Weierstrass theorem.



Reverse Mathematics 'in the small'

- Program: find tractable cases of mathematical interest where one can compute, using an automated theorem prover, that certain conditions are necessary for certain concrete theorems.
- Applications:
 - Elimination of axioms (e.g., universe, choice, replacement, etc.)
 - Isolating the need for the axiom of infinity
 - Discovering reversals, à la reverse mathematics (e.g., the necessity for the axiom of choice for certain theorems)

References

- A. Arana, “On formally measuring and eliminating extraneous notions in proofs”, **Philosophia Mathematica** **17** (2009), 189-207
- **The mizar homepage**
- A. Quaife, **Automated Development of Fundamental Mathematical Theories**, Kluwer Academic Publishers, 1992
- S. Simpson, **Subsystems of Second-Order Arithmetic**, 2nd edition, Cambridge University Press, 2010
- J. Urban, “MPTP—Motivation, implementation, first experiments”, **Journal of Automated Reasoning**, **33**(3-4) (2004), pp. 319–339
- **The mizar-items site**