Tomáš Lávička, Ph.D.

Technology Nexus Secured Business Solutions AB

Evropská 2588/33a, 160 00 Praha 6

Phone: +420 777 695 694

Email: lavicka.thomas@gmail.com

Supervisor´s opinion of the diploma thesis

Bc. Jáchym Šimon

"Automated proof-checking of the Rose-Rosser's proof of completeness of Łukasiewicz propositional logic"

The goal of the thesis was to formalize Rose-Rosser's proof of completeness of Łukasiewicz infinitely valued propositional logic in Lean theorem prover, which was recently rewritten with modern notation in a bachelor thesis of P. Fejl. The thesis was a natural continuation of student's bachelor thesis were he formalized analogous result for classical logic. The completeness proof itself is rather long, complex, and relies on heavy syntactic manipulations that are very challenging for human reader to verify. Such a proof provides motivation and reason to use automated tools in proof verification process. Without a doubt such a tools are gaining importance in modern mathematics.

There are two parts of the thesis. The first and the more important one is the proof formalization, i.e. a code in Lean language. The second part is the thesis itself, which provides the reader with guidelines to the first part. The second part is clearly formulated and explains the key concepts used within the formalization well. It presents main definitions, theorems and techniques used in their formalization in Lean as well as providing useful examples further explaining the involved concepts. Nevertheless, the main contribution is the Lean code, which I discuss in the remaining part of the review.

Overall, the code consists of almost 4000 lines separated into five files. It already gives the impression of the tremendous amount of work involved in it, especially when we realize all the challenges such an endeavor entails. First important observation when one attempts to formalize a mathematical proof is the rigor required. Slightly different formulation of one's definition may completely change how easy and naturally the given notion is used later. This requires a deep insight into the concept of formal proof (type theory) and capabilities of Lean. Often, while diving deeper into Lean, one discovers better ways to formalize the key concepts, which lead to many iterations to the code. Case in point being one of the fundamental notions of the thesis, the notion of provability in Łukasiwicz logic. In the version of the code presented previous year in a form of research assignment, the notion of proof in Łukasiwicz was based on a sequence of formulas (the usual formulation in mathematical logic). However, this turned out to be extremely inefficient, causing the entire code to compile[1] for little less than an hour. Replacing the concept with computationally more convenient notion of tree of formulas enabled us to simplify every consequent notion as well as

[1] Succesful compilation of the code = validity of the formalized proofs.

reducing the compile time to roughly 30 seconds. This required rewriting the whole thing. It is a great demonstration of difficulty of finding the right balance between simplicity, readability, succinctness and performance of the code. Overall I am extremely satisfied with the final state of the submitted code. It reflects great progress the student made since his bachelor thesis. In fact, the result greatly exceeds my original expectations. Particularly, I want to highlight the interactive apply_PF tactic which is often used in the second part of the thesis. The tactic unifies the goal of a proof with one of the current assumptions whose type is only provably equal. It simplifies the related proofs by a great deal. Moreover, its most important feature is that the tactic is designed for efficiency. It is achieved by a clever use of pattern matching on a tactic level. The proofs in the second part of the code compile in reasonable time thanks to this tactic.

To conclude, the student demonstrated great amount of intelligence and diligence and even though the student did not manage to formalize the whole completeness proof, the result is definitely to be considered very valuable and exceeding requirements for a master thesis. In fact, not so much remains for it to be finished and publishable (the student intends to see it through). Especially the syntactic part might serve as a guideline to extend Lean' mathlib library by results in mathematic/algebraic propositional logic. The formalization is designed to be easily generalizable to other propositional logics, including the invaluable proof_verifier tactic (e.g. one could formalize a class of logics with equivalence connective[2], which would immediately inherit the capability of to handle congruence).

Given the above, I clearly propose to evaluate this thesis with a grade of A-excellent and I recommend it to defend an engineering degree.

In Prague 19. May 2022

Tomáš Lávička, Ph.D

---

[2] Such a class of logic is, in fact, natural and deeply studied (see e.g. the hierarchy of Protoalgebraic logics).