Proof Reconstruction

Proposal Report

Andrés Sicard-Ramírez
Diego Alejandro Montoya-Zapata

EAFIT University

Department of Mathematical Sciences

Mathematical Engineering

Medellín

Colombia

2015

1 Problem Statement

In order to make an accurate description of the problem, let us begin by explaining the context to which the problem belongs.

Agda is a proof assistant. It is an interactive system for writing and checking proofs. Agda is also a functional language with dependent types. "*Dependent types are types that depend on elements of other types*"[1].

Automated Theorem Proving (ATP) deals with the development of computer programs that show that some statement (*conjecture*) is a *logical consequence* of a set of statements (the *axioms* and *hypotheses*)[2].

Currently, Professor Andrés Sicard-Ramírez is working on a new approach in computer-assisted verification of lazy functional programs where functions can be defined by general recursion. He is working in first-order theories of functional programs and he is using Agda for formalizing his theories as well as the Automated Theorem Proving systems (ATPs) for proving the properties of the programs. In order to use the ATPs, he provided a translation of the Agda representation of first-order formula into TPTP -a language understood by many ATPs-. This translation was performed by the **Apia** program.

Then, we can now present our problem. In this moment, the communication between Agda and the ATPs is uni-directional because the ATPs are being used as oracles. A first-order conjecture represented in Agda is sent to the ATPs via the Apia program, and the ATPs prove or disprove it (using a fixed timeout). This is unacceptable for most Agda users due to (*i*) the consider a theorem proved only if it has been verified by the proof assistant and (*ii*) the user must also trust that translation from Apia into ATPs logic is being done adequately. Thus, in order to increase the reliability of this process, it would be highly desirable to establish the communication in the other direction, by the reconstruction of the Agda proof terms associated with the proved conjectures, from their ATPs proofs.

2 Goal

The goal of this project is to do a contribution in the field of the formal verification of programs, specifically in the computer-assisted verification of lazy functional programs. We think that this goal can be achieved by performing the reconstruction of proof term in Agda and its posterior integration to the Apia program.

3 State of the Art

Blanchette et. al. [3] describes the three main approaches that are being taken to perform *proof reconstruction:*

- *Replay the ATP proof directly, inference by inference, inside the proof assistant.* This is the approach used by PRocH, by the Isabelle proof methods *metis* and *smt*, and by the SMT bridge for HOL4.
- *Check the ATP proofs using a verified checker.* This is the approach implemented in the SMT integration in Coq and the Waldmeister integration in Agda.
- *Translate the ATP proofs to the proof assistant's source form.* In this approach, the translation need not preserve all the steps of the ATP proofs. This was the original approach implemented for Sledgehammer.

SMTCoq is a Coq tool that checks proof witnesses coming from external SAT and SMT solvers. SMTCoq was developed inside Coq and in [4] can be found a description of how it was built as well as the results of some experiments that were performed in order to check its behavior.

Sledgehammer is a component of the Isabelle/HOL proof assistant that integrates external ATPs to discharge interactive proof obligations. Something impressive is that Sledgehammer transforms the proofs by contradiction into direct proofs [5]. Unlike the SMTCoq, Sledgehammer does reconstruct the whole proof, while the other one just verify some pieces of the proof.
Foster and Struth integrated the Waldmeister ATP to Agda [6]. They implemented equational logic for reconstructing the Waldmeister proofs step-by-step with in Agda.

4 Justification

The formal verification of software has been of importance since the last century. This affirmation can be supported by the words of Hoare: "I hold the opinion that the construction of computer programs is a mathematical activity like the solution of differential equations, that programs can be derived from their specifications through mathematical insight, calculation, and proof, using algebraic laws as simple and elegant as those of elementary arithmetic" [7]. Nowadays, the formal verification is still of interest for the researchers who think as Hoare. For this reason, the solution of the proposed problem would impact positively mathematics and computer science as well as it might be helpful for those researchers who focus their work in computer-assisted verification.

5 Scope

During the last years a huge number of ATPs have been developed and improved; furthermore, a library of test problems for ATPs, named TPTP, has been developed. The TPTP library has provided the community with standards for input and output for ATPs [8]. Nevertheless, it does not exist a standard for the way the proof is printed which make it difficult to try to do a program to reconstruct the proofs for all of the ATPs. For this reason, we are focusing our efforts in formulating the demonstration in Agda just for the **SPASS** ATP system.

6 Methodology

In order to accomplish the objective of the project, it will be invested 5 hours per week in the development of the activities mentioned in the next section as well as in literature review. Furthermore, weekly meetings between the tutor and the student have been scheduled. The aims of these meetings is to discuss the results obtained, try to solve problems or doubts regarding to the project, as well as to promote the team work.

7 Schedule of Activities

In Table 1 is shown a list of activities whose fulfillment would allow us to achieve the main goal.

Main activity	Related activities	Start Week	End Week
Proposal report	Proposal presentation	3	4
Study of SPASS	First tests with the basic rules of inference, documentation of those tests	4	8
Progress report	Oral progress report	8	9
Translate the proofs into Agda	Translation of the proofs by ourselves, construction of a library to automatize this	10	16
Project report	Project presentation	16	17

Table 1: Detailed schedule of activities.

8 Intellectual Property

Based on the Intellectual Property Regulation of the University, these project belongs to these entities: the authors, the Logic and Computation Research Group and the University. In Table 2 are specified the proportions in which the utilities generated by this project would be distributed between these entities.

Subject	Percentage of participation
Authors	30 %
Research group	15 %
University	55 %

Table 2: Distribution of the utilities generated by the project.

References

[1] A. Bove and P. Dybjer, "Dependent types at work," in *Language Engineering and Rigorous Software Development* (A. Bove, L. Barbosa, A. Pardo, and J. Pinto, eds.), vol. 5520 of *Lecture Notes in Computer Science*, pp. 57–99, Springer Berlin Heidelberg, 2009.

[2] G. Sutcliffe, "An Overview of Automated Theorem Proving."

[3] J. C. Blanchette, C. Kaliszyk, L. C. Paulson, and J. Urban, "Hammering towards QED," *Journal of Formalized Reasoning*, 2015.

[4] M. Armand, G. Faure, B. Grégoire, C. Keller, L. Théry, and B. Werner, "A modular integration of sat/smt solvers to coq through proof witnesses," in *Certified Programs and Proofs*, pp. 135–150, Springer, 2011.

[5] J. C. Blanchette, S. Böhme, M. Fleury, S. J. Smolka, and A. Steckermeier, "Semi-intelligible Isar Proofs from Machine-Generated Proofs," 2015.

[6] S. Foster and G. Struth, "Integrating an automated theorem prover into agda," in *NASA Formal Methods*, pp. 116–130, Springer, 2011.

[7] C. A. R. Hoare, "An axiomatic basis for computer programming," *Communications of the ACM*, vol. 12, no. 10, pp. 576–580, 1969.

[8] G. Sutcliffe and C. Suttner, "The TPTP Problem Library for Automated Theorem Proving," 2001.

[9] A. Sicard-Ramírez, *Reasoning about Functional Programs by Combining Interactive and Automatic Proofs*. PhD thesis, Universidad de la República - Uruguay, Montevideo, Uruguay, 2014.

[10] EAFIT University, "Reglamento de Propiedad Intelectual," 2009.