The “𝕄ATh Tutor” project

Sebastian Sahli
Dept. of Mathematics and Statistics
University of Constance, Germany
sebastian.sahli@uni-konstanz.de

1 Introduction

The 𝕄ATh-project I'm involved in originated in an attempt to combine various aspects of high level math education in a common formal framework [1]. The name 𝕄ATh is an acronym for **M**eta-language for **M**odels, **A**lgorithms and **Th**eories. The current development is guided by the objective to support course work at various academic levels and task areas, ranging from mathematical modeling to the implementation of numerical algorithms.

One of the big goals of the 𝕄ATh-project is to provide support for beginner students struggling with the formal aspects of mathematics. In traditional beginner lectures, students have to discover the importance of these aspects and the underlying structure largely on their own on a learning-by-doing basis.

In order to provide support for those students, the idea came up to carve out all these rules and write them down explicitly in some sort of instruction manual which describes the structure precisely while being close to the mathematical reality of first year students. These precisely formulated rules, paired with an intuitive syntax, should serve as foundation for a proof checking program being used as a training tool for the students to practice the handling of these rules by providing immediate feedback. Due to our background in applied mathematics, the tool should also be used in the development of numerical algorithms and mathematical models.

2 Research questions

One could argue that there are already a lot of proof checking tools (like PVS or Mizar) that can be used for our purposes. However, our specific set of requirements makes it difficult to use one of these tools as an all-in-one solution. Our desired tool should provide an intuitive syntax and emphasize the aspects of a proof which are crucial to convey its ideas to humans. Also, having control over the data structures of the tool appeared to be mandatory for its use in numerics. This justified in our eyes the development of our own tool.

Hence, in my work I deal with the question how a language and a proof checker can be designed to meet all these requirements at once. Dealing with this task requires balancing between mechanical formalism and the human nature of being sloppy and omitting details in proofs and raises a lot of questions like the following:

> *How can one design a syntax that resembles usual mathematical notations while being parsable by a machine?*
>
> *How can a proof checker allow the users to focus on the important parts of a proof while assisting them in proving details that are usually omitted?*

The goal of my work is to answer these questions by providing an example for a proof checking tool and a language (called the 𝕄ATh-language) which (hopefully) matches all these requirements.

In: C. Kaliszyk, E. Brady, J. Davenport, W.M. Farmer, A. Kohlhase, M. Kohlhase, D. Müller, K. Pąk, and C. Sacerdoti Coen (eds.): Joint Proceedings of the FMM and LML Workshops, Doctoral Program and Work in Progress at the Conference on Intelligent Computer Mathematics 2019 co-located with the 12th Conference on Intelligent Computer Mathematics (CICM 2019), Prague, Czech Republic, July 8–12, 2019, published at http://ceur-ws.org

3 Completed research

The first step in the development of the 𝕄ATh-language was to identify and name the constructs and objects arising in mathematical texts and also the details which are likely to be omitted. This was done in cooperation with Michael Junk by observing students during lectures and exercises. These obsevations resulted in the development of a formalism described in [2].

Next, I developed a formal syntax for the important concepts of this formalism and Java data structures representing these concepts along with a prototype of a proof checker implementing the formal rules. This implementation changed the view on some aspects of the formalism and led to its improvement which again resulted in a change in the implementation, starting an iterative process of mutual improvements. This process resulted in an intermediate version presented at CICM 2018 which was, however, not released due to some barely resolvable conceptual issues.

4 Research and evaluation plans

The current goal of my project is to provide an improved implementation of the proof checker (now in the Scala language) which resolves the issues of the current version.

The performance of this version will be evaluated by translating lecture notes for beginner lectures on modelling and analysis into the new language and let them be checked by the proof checker searching for conceptual, technical and usability issues. Additionally, the tool will be used during preliminary courses and basic lectures on modeling to obtain feedback from students and make observations on the usage and acceptance of the system. The gathered information will be used to improve the system.

After the completion of the first version of the system we will investigate to what extent the system can be connected to other formal languages (like MMT) in terms of formulating the concepts of the 𝕄ATh-system in the other language and reverse.

5 Publications plans

The most important publication task will be the release of a first version of the proof checker together with its source code and a description of the language and the functionality of the program. This description will contain both an introduction from the user's point of view and an more technical insight into the syntax and semantics of the system. The latter will be the basis of my thesis.

Later publications will describe the experiences made with formalizing beginner lectures in the 𝕄ATh-language. Further papers may deal with the acceptance of the new system among students and the observations made during the use in lectures. Even studies on the influence of the system's usage on the students' performance are possible. Another option is to treat the connection with the 𝕄ATh-language to other formal systems. However, this depends on the time remaining after finishing the main task of the project.

References

[1] 𝕄ATh-Homepage, http://www.math.uni-konstanz.de/mmath/en/

[2] Junk, M., Hölle, S., Sahli, S.: *A Meta Language for Mathematical Reasoning.* In: Osman Hasan, O., Kaliszyk, C., Naumowicz, A. (eds.) Proceedings of the Workshop Formal Mathematics for Mathematicians (FMM), Hagenberg, Austria (2018)