Report on the outcomes of a Short-Term Scientific Mission[1]

Action number: CA20111

Grantee name: Maximilian Doré

Details of the STSM

Title: Conjecture and proof search in Agda with large language models

Start and end date: 10/03/2024 to 14/03/2024

Description of the work carried out during the STSM

Description of the activities carried out during the STSM. Any deviations from the initial working plan shall also be described in this section.

The primary goal of my STSM was to start a collaboration with Moa Johansson and others at Chalmers on developing a Copilot for Agda, i.e., an system which helps Agda users formalise mathematics based on a large language model. We have had many fruitful discussions to carry this project further, the STSM was hence a full success.

On Tuesday 12 March I gave a seminar talk titled “Teaching LLMs mathematical reasoning through programming” for which was attended both by the types and data science group. The talk was well-received and we had a fruitful discussion on the objectives for an Agda Copilot afterwards.

On the following days, I got together with different researchers in the department for more focussed discussions. With Moa and her postdoc Jonathan Thomas I discussed different encodings for Agda code and how we could adapt the large language model “Llemma” for our purposes. Rocío Mercado, who is an expert on using AI for biomolecular engineering, sketched how molecular structures are encoded in her research field, which provided very interesting impetus for our thoughts on encoding Agda code for an LLM. Jean-Philippe Bernardy, whose research used to be on the symbolic side but who is now interested in neural methods, introduced me to his recent paper on encoding the abstract structure of Agda programs. We had a very fruitful discussion on the advantages and disadvantages of using an LLM versus other neural machine learning methods. In our discussion we were joined by Andreas Abel, one of the main developers of Agda, who provided very useful insight in the automated reasoning capabilities of Agda. We concurred that the outputs of Agda’s automated reasoning system provides a good first benchmark for an Agda Copilot.

[1] This report is submitted by the grantee to the Action MC for approval and for claiming payment of the awarded grant. The Grant Awarding Coordinator coordinates the evaluation of this report on behalf of the Action MC and instructs the GH for payment of the Grant.

Moa also introduced me to other members of the department. Wolfgang Ahrendt introduced me to his work on using LLMs in formal methods, and we discussed the different learnings that formal methods for software engineering and formalised mathematics can draw from each other. I also had a long conversation with Andrea Silvi, one of Moa's PhD students, about his project. He works on devising a transformer to learn regex rewriting.

Lastly, I presented past work of mine on Wednesday 13 March with a seminar talk titled "(Automatically) verifying Morse reductions in Cubical Agda" at the Proglog series, which is organised by the Types group at Chalmers. The audience raised very useful questions and ideas for future work relating to the formalisation of topological data analysis in Cubical Agda.

Description of the STSM main achievements and planned follow-up activities

Description and assessment of whether the STSM achieved its planned goals and expected outcomes, including specific contribution to Action objective and deliverables, or publications resulting from the STSM. Agreed plans for future follow-up collaborations shall also be described in this section.

With Moa and Sólrún Halla Einarsdóttir, a PhD student of Moa, I am currently conducting a first case study on the capabilities of Llemma, the biggest LLM tailored to mathematics, for generating Agda code. This will provide a good baseline for how well general purpose LLMs can generate Agda code. Sólrún is currently setting up Llemma on a Chalmers server. Interacting with a live system will be crucial to get an idea what kinds of prompts lead to useful results. I am working on curating a training set of Agda proofs that can be used to fine-tune Llemma.

In parallel, I will investigate in which way the Copilot could best be integrated into the Agda system. The emacs-mode of Agda is how most users interact with Agda; my plan is that a user-defined tactic (implemented via reflection, Agda's meta-programming facility) is invoked via the emacs-mode and calls the LLM to generate proofs/programs, conjectures/types and definition/datatypes. I intend to implement this tactic to work for an arbitrary LLM. As Jonathan has suggested, I will try to use the platform LangChain, which provides a framework to develop a pipeline independently of a concrete LLM. This would allow us to easily switch between Llemma; a fine-tuned version of Llemma; or any other model that might be released in the future.

As a side project, I am currently getting up to speed with Andrea's project on letting a neural network learn regex rewriting. Since I have plenty of experience with regular languages, I hope to provide some feedback for Andrea from the theoretical CS side.

I am currently exploring funding opportunities to go to Chalmers for a postdoc. The STSM has been very fruitful to gather and refine research ideas, and I would be excited to continue working with Moa and others at Chalmers.