Short-Term Scientific Mission Grant
- APPLICATION FORM[1] -

Action number: CA20111

Applicant name: Konstantinos Kogkalidis

Details of the STSM

Title: Neural Premise Selection for Agda

Start and end date: 19/06/2023 to 23/06/2023

Goals of the STSM

The goal of the project is to build a neural premise selection tool for the automated theorem prover and functional programming language Agda. The project differentiates itself from traditional approaches in the field of automated theorem proving, such as using off-the-shelf LLMs or relying on a neural network to produce proof tactics. Instead, it aims to leverage the structural invariances and symmetries using modern graph neural networks, creating "deep" representations of type- and term-structure. Furthermore, the problem is reformulated as merely suggesting valid lemmas for a program hole, which can then be verified and further completed by the type checker. This unique approach is expected to have significant benefits, resulting in a significant publication at a high-quality conference, and paving the road for future research in this area.

Working Plan

Over the last few weeks, we have been collaborating remotely, and have already achieved a great deal; we have an initial collection of training data (consisting of the Agda standard library), a prototype model is currently in the works, and preliminary experiments are being performed. We believe that our collaboration would be even more productive if we were able to have a focused sprint in person. The workplan would involve a combination/selection of the following:

1. Review and refine the existing data collection, model prototyping, and initial experiments that have already been completed.
2. Find ways to align the shape and structure of the data with appropriate graph neural networks.
3. Evaluate our need for more data and gather more as needed (e.g., crawling for online git repositories) -- these can find use to practically assess the real-world potential of our tool.
4. Experiment with alternative/simpler architectures and establish baselines.

[1] This form is part of the application for a grant to visit a host organisation located in a different country than the country of affiliation. It is submitted to the COST Action MC via e-COST. The Grant Awarding Coordinator coordinates the evaluation on behalf of the Action MC and informs the Grant Holder of the result of the evaluation for issuing the Grant Letter.

5. Develop the specifications and identify the requirements for an easy-to-use interface between our tool and Agda.
6. Write up the results of the project aiming for a high-quality publication, highlighting the unique approach and significant impact of the project on the field of automated theorem proving.

Expected outputs and contribution to the Action MoU objectives and deliverables.

Expected Project Outputs:

1. A neural premise selection tool for the automated theorem prover and functional programming language Agda using state-of-the-art graph neural architectures.
2. A dataset of program holes and associated lemmas, as well as a methodology for collecting and curating such a dataset.
3. The bases for a universal, framework-agnostic methodology to structure-aware neural representations of the types and terms of any functional programming language.
4. A set of trained, custom-tailored graph neural network models for predicting lemmas for program holes based on the goal type and program scope.
5. A methodology for integrating the neural premise selection tool into the Agda type checker and automated theorem prover.
6. A research paper describing the project's methodology, experiments, and results, potentially to be submitted to a high-quality conference in the field of machine learning (e.g., ICLR)

The project outlined aligns with several of the objectives outlined in the Research Coordination Objectives, including:

1. Promoting the output of detailed, checkable proofs from automated theorem provers.
2. Making techniques for program verification more effective and more accessible to all stakeholders.
3. Developing the use of artificial intelligence and machine learning techniques on proofs.