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

Action number: CA20111

Grantee name: Dorel LUCANU

Details of the STSM
Title: Translation of Generic Symbolic Execution Steps into Dedukti Start and end date: 19/03/2023 to 02/04/2023
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.

[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.

(max. 500 words)

Grantee enters max 500 word summary here.

The proposed plan included:

- a deep understanding of how the symbolic execution is handled by K Prover;

- how the K Prove symbolic execution steps can be translated in Dedukti proofs;

- how the K Prover can benefit from the rich library system of Dedukti by borrowing proofs for the results given by the external algorithms it uses.

I did the following activities in order to achieve these objective:

- I cooperated with K team in order to find out how we can instrument the call of the K prover to obtain the trace of the steps performed by the K prover.

- I prepared and presented slides explain the K definitions in terms of Matching Logic (ML).

- I prepared and presented slides that explain the steps of the K prover and the proof system used by the K prover, and how this steps are represented as ML patterns.

- I had daily meetings with Amelie Ledein (PhD student in the Dedukti team), where I explained the K prover steps and Amelie explained me how Dedukti works and the translations. of the concrete executions of K definitions in Dedukti.

- During these meetings, we also experimented several ways of how the main K Prover steps can be checked in Dedukti.

- I had periodically meetings with Catherine Dubois and Valentin Blot to summarise the work progress and to design the main approach of how Dedukti can check proofs of the K prover. One of these meetings hold at École Nationale Supérieure d'Informatique pour l'Industrie et l'Entreprise (ENSIIE).

- I delivered a Dedukti seminar with title "A gentle introduction to Matching Logic and its Applications".

- I also had irregularly meetings with other Dedukti members (Gilles Dowek, Frederic Blanqui, Dietmar Berwanger, postdocs, PhD students).

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.

(max. 500 words)

Grantee enters max 500 word summary here.

The main achievement of the work done during this STSM is the design of an efficient methodology to check in Dedukti proofs of the K prover. This methodology consists of:

- Extracting from the trace file obtained from the K prover the essential elements needed to check the proof and the translation to Dedukti.

- Preprocess this information in order to build the proof tree.

- We observed that not all the semantics rules of the programming language are needed to check a symbolic step, and therefore we designed an algorithm that preprocesses the language definition to find these groups of rules. It is important to point out that these groups are not depending on the current configuration of the symbolic step.

- The checking of the branching steps is relatively easy (some experiments were already done on the IMP language).

- The checking of the rules for that there is no a branch in the proof tree requires an implementation of an unification modulo builtins algorithm.

- The checking of the progress condition for the circular hypotheses.

- Assembling the checks for individual steps using Consequence, Disjunction, Reflexivity and Transitivity rules in order to obtain a Dedukti term to be checked.

A prototype of this approach, based on the IMP language, follows to be implemented by joint work.

The correctness of the approach can be proved as follows:

- The preprocessing of the Kdefinition defines in fact a translation of the programming language definition into an equivalent one, but where the rules are "all paths" instead of "one path".

- A theorem will state the equivalence of the two theories and the correspondence between the corresponding proof systems.

- The checking process is done using the transformed theory, for which the inference rules are much simpler.

A sketch of a detailed technical report has already started, and from this a conference/journal paper will be produced. We hope that the prototype will be functional for checking proofs of the IMP programs at the moment of the publication.