Proposal for STSM

Details of the STSM

Title: Reconstructing AletheLF Proofs in Lambdapi

Dates: 01/04/2024 - 14/04/2024

Goals of the STSM

AletheLF (ALF) is a new fine-grained proof format based on SMT-LIB 3, which is going to be supported by SMT-solvers such as cvc5 in their proof-producing module. This recent format uses a stronger typing system making possible the production of proofs for bit-vector operations, List, and other features that could not be expressed by the current Alethe proof format. Nevertheless, AletheLF proofs may contain errors due to bugs in the SMT-solvers generating incorrect proofs. Therefore, we would like to develop an automatic procedure for verifying ALF proof with a proof assistant. In the core language of ALF, SMT-LIB 3 theories are defined as ALF signatures and rewrite rules. Lambdapi is an interactive proof system featuring dependent types like in Martin-Lőf's type theory, but allows to define objects and types using oriented equations (rewriting rules), and reason modulo those equations. We believe that the native support of rewriting logic and dependent type in Lambdapi make it a suitable choice for reconstructing ALF proofs due to their similar semantics. We thereby would like to develop a tool that will reconstruct ALF proofs into equivalent Lambdapi proofs, and verify it with the Lambdapi inference kernel.

Working plan

Similar to Alethe, a proof produced in ALF takes the form of list of indexed steps that can reference steps appearing before them in the list. Steps without references are tautologies and assumptions. The last step is always the deduction of the empty clause. Furthermore, ALF supports subproofs, which are used for local assumptions and to reason about bound variables. The syntax is based on SMT-LIB 3. The core language of ALF does not assume any definitions of standard SMT-LIB 3 theories. Instead, SMT-LIB 3 theories are defined as ALF signatures and rewrite rules. We intend to develop a shallow embedding of ALF logic and its type system in Lambdapi. Besides, we want to show the embedding is sound i.e. derivable in Lambdapi inferences kernel. The core language of ALF will be translated into corresponding symbols in Lambdapi, and rewrite rules in ALF which will be converted into Lambdapi rewrite rules. As a first step, we plan to shallow embed the standard SMT-LIB 3 theories e.g. resolutions of clauses and computations operators, into a Lambdapi library. Subsequently, we aim to develop a binary that will translate an AletheLF proof script into a Lambdapi proof. Carcara is a proof checker and elaborator for SMT proofs in the Alethe format. We aim to adapt Carcara to the new ALF proof format and add a Lambdapi proof translating module in

Carcara. In the case where ALF proof does not provide enough fined-grained information to reconstruct the proof, Carcara will elaborate on the proof by inferring the missing information. Elaboration is a key ingredient for the success rate of reconstruction, and hence the usefulness of this approach depends on the quality of the generated proofs.

Expected outputs and contribution to the Action MoU objectives and deliverables

Proof assistants allow users to invoke an external automatic theorem prover such as an SMT-solver. This allows the user to focus on the core of their argument by reducing the burden of manual proof. In the proof assistant Isabelle/HOL this approach is implemented in the smt tactic, and the tactic hammer in CoqHammer, an automated reasoning tool for Coq. However, as we mentioned these proofs might be incorrect, so the generated proof needs to be reconstructed inside proof assistant. We believe that the fine-grained proofs format proposed by AletheLF might allow for a higher success rate in reconstruction. Moreover, the new feature supported in ALF would enlarge the proof finding of input problems. This work covers two goals of Research Coordination Objectives, with the first goal being to “Promote the output of detailed, checkable proofs from automated theorem provers”, and the second will be to extend the database of proof systems translation of Dedukti logical framework by adding ALF. Moreover, the translation to Lambdapi will provide a way for users to export ALF prooifs in other proof systems such as Isabelle, HOL, or Coq. That contribution is intended to be part of WG2 Automated Theorem Provers works, and WG1 Tools on Proof Systems Interoperability.