

Proofs in Vampire

Ashutosh Gupta

IST Austria

Various applications of theorem proving not only require to check if a theorem holds true but also need a proof of the theorem, e.g., spurious counterexample analysis in verification [1].

The proofs can be presented in various formats to a user. Not all proof formats are conducive to the need of a user. The user may prefer one proof system over another. A user may prefer to have concise proofs, and other may like to have detailed proofs that allow inexpensive proof checking. Furthermore, there is no commonly acceptable API or file format for proof output. This situation allows the provers to produce proofs that are mutually incompatible and do not match the need of the users. Producing a proof after a run of an efficient and complex theorem prover is not a straight forward task. For example, the provers may apply various simplifications on input formulas before running the core proving algorithms. Such simplifications should be added as proof steps in the generated proofs. In long and arduous development process of a theorem prover, developers may forget to provide proper annotations to all the deduction steps such that a sufficiently detailed proof can be generated.

Currently, a VAMPIRE proof only contains the information about the antecedents of the deduced clauses and the proof rule that is used to produce the clauses. This proof format leaves out various details. For example, an application of the superposition rule does not specify the terms that were unified to obtain the consequences. We propose to develop a comprehensive proof format and API that allows proofs with different level of granularity and are easy to use. Towards this end, the VAMPIRE team should work towards the following objectives.

Proof Format VAMPIRE should support a flexible proof format that allows broad range of proof rules and also extendability with new proof rules.

Proof API VAMPIRE should provide an API that allows users to access and manipulate the proofs. This feature will be greatly helpful to the subsequent research about generating proofs with certain structure that needed for certain applications [2].

Proof Checker A generated proof is meaningless until its semantic is clear to the user. It is usually hard to convey the semantics of a ever evolving proof format in text. Therefore, VAMPIRE may provide an open source proof checker such that users can understand the invariants and expected structure of the VAMPIRE proofs.

References

- [1] T. A. Henzinger, R. Jhala, R. Majumdar, and K. L. McMillan. Abstractions from proofs. In *POPL*, 2004.
- [2] K. Hoder, L. Kovács, and A. Voronkov. Playing in the grey area of proofs. In *POPL*, pages 259–272. ACM, 2012.