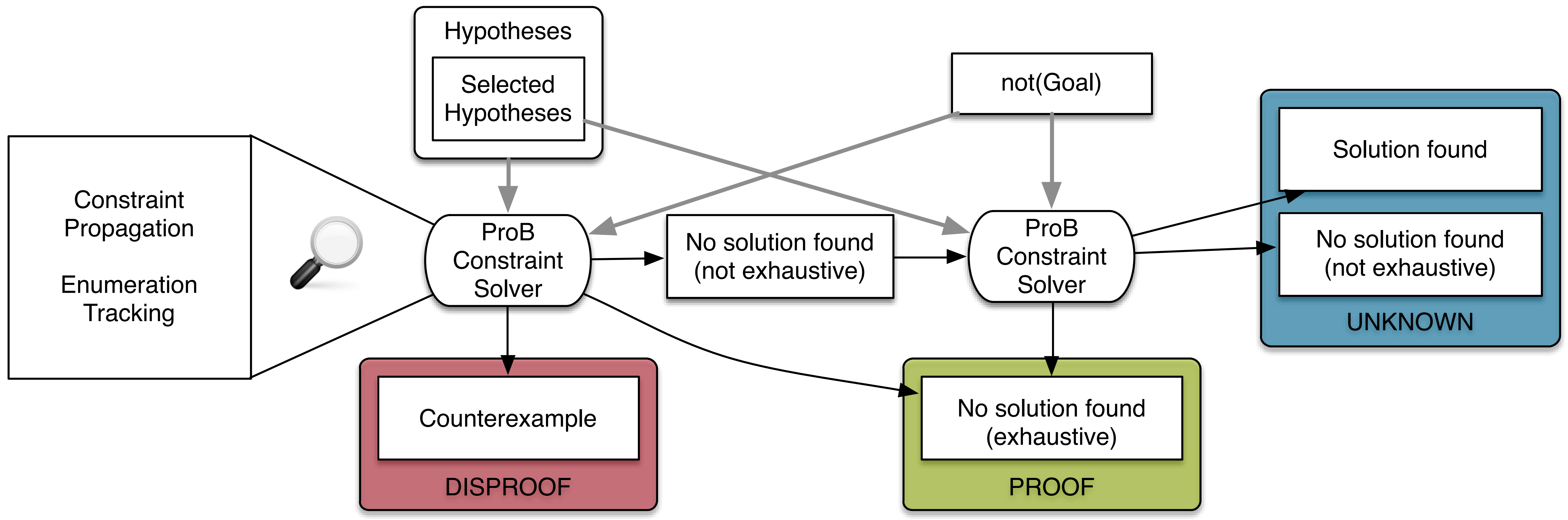
Turning Failure into Proof: The ProB Disprover

Sebastian Krings

Heinrich-Heine-University Düsseldorf, Germany

Abstract

Initially, the ProB disprover used constraint solving to try and find counterexamples to proof obligations generated from Event-B models [1]. Recently, we made the ProB kernel capable of determining whether a search was exhaustive. Hence, one can now also use it as a prover. It uses a technique which guarantees soundness in the following way: if a solution is found it is guaranteed to be correct; if not, the solver can either return the result 'definitely false' if the enumeration was exhaustive, or 'unknown' if the enumeration was aborted.



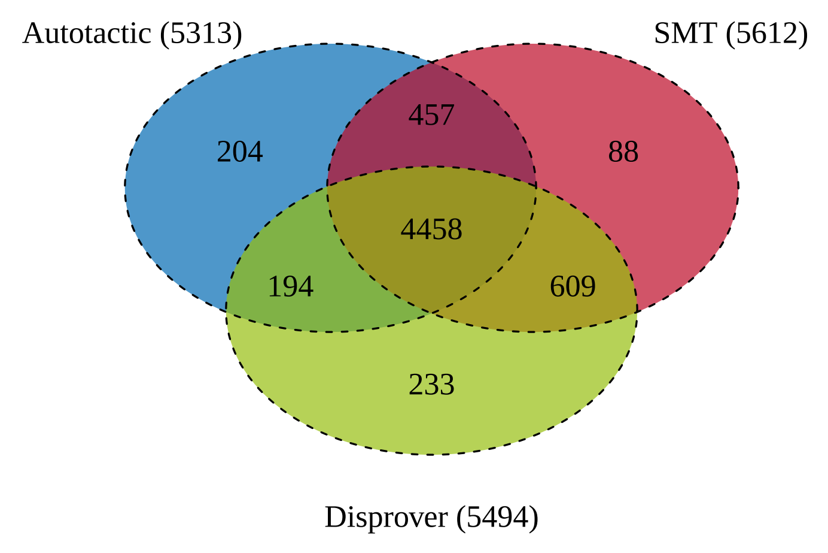
Figure

Technique

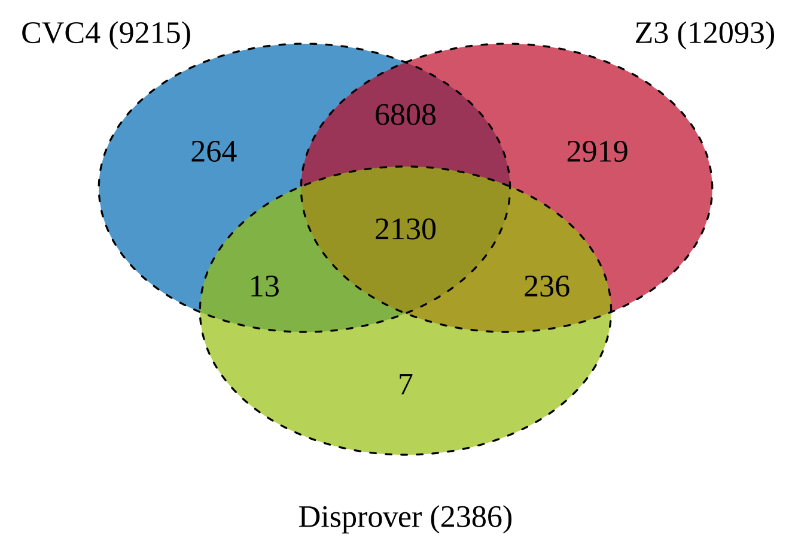
ProB's constraint solver is based on CLP(FD)-style constraint propagation and resorts to enumeration once no further propagation is possible. Hence, it relies on variables being constrained to a finite domain. While this is fine for animation and visualization, it limits the applicability of ProB as a prover. To overcome this limitation, we decided to track the enumeration of certain variables. We firstly set up an environment for each scope (i.e. quantifier, set comprehension, and so on), stating whether a variable has to be enumerated exhaustively for a proof to be successful. Then, we use this information to direct the enumeration towards certain variables, knowing whether a solution means satisfiability or unsatisfiability or if further search is necessary. With this technique, we are able to prove if certain hypotheses H imply a *Goal* by searching for a solution to $H \Rightarrow \neg Goal$.

Results

We applied the ProB disprover to Event-B proof obligations [2] and compared it to the classical B provers as well as to the SMT solver based prover. Additionally, we wrote a translation from SMT-LIB to B and used problems selected from the SMT-LIB collection of benchmarks. Here, we compared to CVC4 and Z3. The diagrams below show the number of discharged / solved problems.



Proof Obligations



SMT-LIB

Conclusion

As explained before, ProB's technique differs from classic DPLL(T)-based provers, making it an orthogonal addition to them rather than a replacement. Our empirical evaluation shows, that our approach is useful for certain kinds of problems which are otherwise hard to solve. We therefore think that it should be evaluated and refined further.

Acknowledgments & References

Joint work with Jens Bendisposto and Michael Leuschel.

[1] Olivier Ligot, Jens Bendisposto, and Michael Leuschel.
Debugging Event-B Models using the ProB Disprover Plug-in.
Proceedings AFADL'07, Juni 2007.

[2] Sebastian Krings, Jens Bendisposto, and Michael Leuschel.
Turning Failure into Proof: Evaluating the ProB Disprover.
In *Proceedings of the 1st International Workshop about Sets and Tools*, 2014.