Learning to Advise an Equational Prover*

Chad E. Brown, Bartosz Piotrowski, and Josef Urban

Czech Technical University, Prague

We describe a simple first-order equational theorem prover, `aimleap`, capable of interacting with an advisor. The prover takes 87 (fixed) quantified unit equations as axioms (all of which are true in AIM loops [2]). The conjecture is then given as an equation and `aimleap` attempts to find a proof by applying paramodulation using one of the axioms to one term occurrence in either the left or right side of the current goal. An external advisor can be used to filter and rank possible paramodulation steps. We report results of using an advisor trained using XGBoost on data described below.

Proof Search We breifly describe the way `aimleap` searches for a proof of $s = t$. Note that s and t may contain constants and variables, where variables are allowed to be instantiated. The search is limited by a bound n which is always initialized to 10.

1. If s and t are unifiable, then report success.
2. If $n = 0$, then report failure.
3. Compute a finite set of paramodulants $s_2 = t_2$, obtaining a representative up to renaming of variables for each possible paramodulant.
4. Order these paramodulants using an advisor, filtering out those which the advisor deems to require more than $n - 1$ paramodulation steps to complete the proof, and for each one ask if $s_2 = t_2$ is provable in $n - 1$ steps.

Primary Dataset Veroff has obtained a large number of AIM proofs using Prover9 [3]. Analysing some of these proofs it was possible to obtain 3468 equations provable from the 87 axioms within 2-10 paramodulation steps. This gave us our initial data for training and testing. Roughly half of the examples, 47.3%, was the result of 2 paramodulation steps. Another 25% resulted from 3 paramodulation steps and 10.2% resulted from 4 paramodulation steps. The remaining 18.5% resulted from 5-10 paramodulation steps, with fewer examples as the number of steps increased. We later augmented the training data with roughly 10,000 synthetically generated data, while still restricting testing to the original 3468 equations.

Rote Learner As a sanity test, an “oracle” advisor was used first. This advisor returns the known distance for goals and subgoals that were seen in the training proofs. For other equations it returned a high distance of 50 (essentially forcing these equations to be pruned from the paramodulant options). We call this the “rote learning” advisor. As expected, the prover could reprove all 3468 problems given the full rote learned information. In some cases, shorter proofs than the training proofs were found. In 132 cases, new proofs involving only one paramodulation step were found. The rote learner also gave us a way to create negative training data by recording each time the rote learner returned 50.

*Supported by the ERC Consolidator grant no. 649043 AI4REASON

Cross-validation We split the 3468 problems into ten parts. Using a rote learner that is allowed to only look up values from the other nine parts, `aimleap` was only able to prove 800 (21.9%) problems.

Constant Distance We also tested an advisor that always returns a constant distance whenever the left and right hand sides of the current goal were not equal (giving 0 if they are equal). This gave poor results except when the constant is 9. When the constant is different from 9, the maximum number of problems solved was 138 (4%). When the distance is given as 9, `aimleap` considers all possible 1 step proofs and some 2 step proofs. Since just over half the problems actually do have a 1 or 2 step proof, an advisor giving a constant distance can solve 1739 problems (50.1%).

Training the Advisor

Search Results using the Advisor

First-Order Automated Provers For further comparison we gave the problems to three automated provers with a timeout of 60s: Prover9 [3], E [4] and Waldmeister [1]. E proved 2684 problems (77.4%), Waldmeister proved 2170 problems (62.6%) and Prover9 proved 2037 problems (58.7%).

Conclusion We have described an equational prover (`aimleap`) capable of interacting with an advisor. This provides a framework for testing the degree to which an advisor (typically one trained using machine learning techniques) is helpful during proof search. With perfect advice all the problems are easily provable by `aimleap`. The current preliminary results appear promising, outperforming rote learning on cross-validation. However, the advisor cannot yet compete with existing ATPs.

References

[1] T. Hillenbrand, A. Jaeger, and B. Löchner. Waldmeister - Improvements in Performance and Ease of Use. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction*, number 1632 in Lecture Notes in Artificial Intelligence, pages 232–236. Springer-Verlag, 1999.

[2] Michael K. Kinyon, Robert Veroff, and Petr Vojtechovský. Loops with abelian inner mapping groups: An application of automated deduction. In Maria Paola Bonacina and Mark E. Stickel, editors, *Automated Reasoning and Mathematics - Essays in Memory of William W. McCune*, volume 7788 of *LNCS*, pages 151–164. Springer, 2013.

[3] W. McCune. Prover9 and mace4. `http://www.cs.unm.edu/~mccune/prover9/`, 2005–2010.

[4] Stephan Schulz. System Description: E 1.8. In Ken McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *Proc. of the 19th LPAR, Stellenbosch*, volume 8312 of *LNCS*. Springer, 2013.