# Training Data Extraction for Identifying Useful Lemmas

Michael Rawson[1], Christoph Wernhard[2], and Zsolt Zombori[3]

[1] TU Wien, Austria michael@rawsons.uk
[2] University of Potsdam, Germany info@christophwernhard.com
[3] Alfréd Rényi Institute of Mathematics, Hungary zombori@renyi.hu

We investigate learning to identify useful lemmas for ATP, where usefulness is defined in terms of 1) reducing proof search and 2) shortening the length of the overall proof. How can ATP performance be improved by the generation and selection of useful lemmas before search begins? The presented work and ideas are guided by four theses.

1. Restriction to a Class of Problems with Accessible and Simple Proof Structures is Helpful

   Interested in novel techniques, we work with a restricted class of first-order problems, *condensed detachment* (CD) problems [10, 8], due to Carew A. Meredith [6]. Inference steps can be characterized by detachment (modus ponens) combined with unification. Proof structures are particularly simple and well accessible: full binary trees, or terms with a binary function symbol $\mathsf{D}$, which we call D-terms. Constants in these terms label axioms. As examples of D-terms consider 1, a constant representing a use of the axiom labeled by 1; $\mathsf{D}(1,1)$, representing a detachment step applied to axiom 1 as major and minor premise; or $\mathsf{D}(1,\mathsf{D}(1,1))$, representing a proof with two detachment steps. These proof terms are closely related to proof structures of the connection method [1, 2].

2. A Utility Model Allows Learning from Successful as well as Failed Proof Attempts

   Our aim is to assist lemma selection via machine learning by building a *utility model* that takes a lemma as input and outputs a predicted utility. We present our data extraction mechanism, in particular our attempts at maximising the training signal coming from a single proof attempt. This approach is in marked contrast with most learning-assisted theorem proving systems (e.g. [4, 5], considering only learning applied to lemmas), that typically learn from steps of the proof found and discard everything else from the search. In particular, very little work exists on how to learn from failed proof attempts.

3. Subtree or Unit Lemmas are Interesting and Powerful

   Given conjecture $C$, if search is successful a D-term proof $P$ is found. Any subtree[1] $P'$ of $P$ turns out to be the proof of some unit lemma, an atomic lemma formula used in the proof. These unit or subtree lemmas are typically used in applications of CD, where the D-terms are then considered in their DAG representation. This includes advanced applications, out of the scope of modern ATP. We overview comprehensive recent results from [7] on learning such lemmas, their generation, selection and application, addressing learning aspects and novel proving techniques that are adequate for these tasks.

4. Even More Powerful Lemmas are Possible and Potentially Useful

   Any connected subgraph $P'$ of $P$, i.e., a tree whose leaves are not all constants but include variables, proves a Horn clause whose head is the term proven by the root of $P'$ and whose body consists of the terms proven by subtrees of $P$ under the leaves of $P'$, or at the variable positions of $P'$. This situation has correspondences in resolution, the connection structure calculus [3] and compression with combinators [9]. Open key questions are the assessment of the usefulness of such powerful lemmas and in which way provers (of different types) benefit from supplementing them with such powerful lemmas (in different ways, as additional axioms or as proof schemas).

[1] We use *subtree* of a tree $P$ to refer to a node in $P$ and all of its descendants in $P$.

## References


[1] Bibel, W.: Automated Theorem Proving. Vieweg, Braunschweig (1982). https://doi.org/10.1007/978-3-322-90102-6, second edition 1987

[2] Bibel, W., Otten, J.: From Schütte's formal systems to modern automated deduction. In: Kahle, R., Rathjen, M. (eds.) The Legacy of Kurt Schütte, chap. 13, pp. 215–249. Springer (2020). https://doi.org/10.1007/978-3-030-49424-7_13

[3] Eder, E.: A comparison of the resolution calculus and the connection method, and a new calculus generalizing both methods. In: Börger, E., Kleine Büning, H., Richter, M.M. (eds.) CSL '88. LNCS, vol. 385, pp. 80–98. Springer (1989). https://doi.org/10.1007/BFb0026296

[4] Kaliszyk, C., Urban, J.: Learning-assisted theorem proving with millions of lemmas. Journal of Symbolic Computation **69**, 109–128 (2015). https://doi.org/https://doi.org/10.1016/j.jsc.2014.09.032, https://www.sciencedirect.com/science/article/pii/S074771711400100X, symbolic Computation in Software Science

[5] Kaliszyk, C., Urban, J., Vyskočil, J.: Lemmatization for stronger reasoning in large theories. In: Lutz, C., Ranise, S. (eds.) Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015, Wroclaw, Poland, September 21-24, 2015. Proceedings. LNCS, vol. 9322, pp. 341–356. Springer (2015). https://doi.org/10.1007/978-3-319-24246-0_21, https://doi.org/10.1007/978-3-319-24246-0_21

[6] Prior, A.N.: Logicians at play; or Syll, Simp and Hilbert. Australasian Journal of Philosophy **34**(3), 182–192 (1956). https://doi.org/10.1080/00048405685200181

[7] Rawson, M., Wernhard, C., Zombori, Z., Bibel, W.: Lemmas: Generation, Selection, Application (2023), preprint, soon to appear on arXiv

[8] Ulrich, D.: A legacy recalled and a tradition continued. J. Autom. Reasoning **27**(2), 97–122 (2001). https://doi.org/10.1023/A:1010683508225

[9] Wernhard, C.: Generating compressed combinatory proof structures — an approach to automated first-order theorem proving. In: Konev, B., Schon, C., Steen, A. (eds.) PAAR 2022. CEUR Workshop Proc., vol. 3201. CEUR-WS.org (2022), https://arxiv.org/abs/2209.12592

[10] Wernhard, C., Bibel, W.: Investigations into proof structures (2023), preprint, http://cs.christophwernhard.com/papers/investigations/