Compressed Combinatory Proof Structures and Blending Goal- with Axiom-Driven Reasoning: Perspectives for First-Order ATP with Condensed Detachment and Clausal Tableaux

Christoph Wernhard

University of Potsdam, Germany
info@christophwernhard.com

1 Background

Goal-driven first-order provers such as *leanCoP* [14] or *SETHEO* [9], which may described as based on clausal tableaux [8], the connection method [1, 3] or model elimination [12], in essence enumerate tree-shaped proof structures, interwoven with unification of formulas that are associated with nodes of the structures. While they do not compete with state-of-the art systems in the range of solvable problems, they have merits that are relevant in certain contexts: Proofs are typically emitted as data structures of simple and detailed forms, making them suitable as inputs for further processing. Through iterative deepening, proofs tend to be short. The provers facilitate comparing alternate proofs of a problem or influencing the shape of proofs. Implementations can be manageable and small [18], making the approach attractive for adaptation to specific logics [15, 16, 17] and novel combinations with other techniques [7, 31, 32, 6].

Here we aim to preserve the merits of that approach, while moving on to stronger proving capabilities. Our concrete starting point is a view of condensed detachment as a specialization of the connection method [29]. It provides a simplified variant of first-order ATP that still has many of its essential characteristics and seems suitable as basis for the development and study of new techniques. Emphasis is on the explicit consideration of proof structures in a simple form, as full binary trees or terms. Condensed detachment has dedicated applications in the investigation of propositional logics [24], reflected in about 200 such *TPTP* problems [27], and can be more generally used as inference rule for arbitrary first-order Horn problems.

The contribution is based on [29] as well as ongoing work [27, 28]. It is backed by an implemented system, *CD Tools*, available as free software from

http://cs.christophwernhard.com/cdtools/.

The system website also provides detailed result tables for experiments, including graphical proof visualizations.

2 Theses

In the contribution we elaborate the following two theses.

Thesis 1: Compressed Combinatory Proof Structures. Representing a proof tree by a combinator term [23, 30] that normalizes to the tree lets subtle forms of duplication within the tree materialize as duplicated subterms of the combinator term. In a DAG representation of the combinator term these straightforwardly factor into shared subgraphs. As an example, consider the proof tree

$$2(2(2(2(2(2(2(21))))))). \tag{i}$$

Here $1, 2$ are axiom identifiers and inner nodes correspond to a condensed detachment step with left and right children as proofs of major and minor, resp., premises. The notation for the inner detachment nodes follows the common notation for application in λ- and combinator terms, i.e., as juxtaposition of left and right subtree, with parentheses according to left associativity. Tree (i) can be expressed with the combinator term

$$\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)(\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)1), \tag{ii}$$

where $\mathbf{B}22$ and $\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)$ each have two incoming edges in the minimal DAG. To search for proofs, combinator terms can be enumerated, like clausal tableaux, with simultaneously relating formulas associated with components of the enumerated structures through unification. From the clausal tableaux or connection method point of view, the approach realizes compressions known from the connection structure calculus [4, 5, 2], which was never implemented. As a refinement to restrict the search space, the enumeration of combinator terms can be based on *proof schemas*, pattern terms such as $\mathsf{r}(p, q)$, with an associated semantics defined by a combinator term with parameters p, q.

Thesis 2: Blending Goal- with Axiom-Driven Reasoning. The goal-driven proof structure enumeration by clausal tableau methods can be generalized beneficially to a method that blends with axiom-driven enumeration in configurable ways. The core method then enumerates proof structures, paired with the most general theorems (MGTs) [29] proven by them from given axioms, on a given *level*, characterized for example by the number of tree nodes or tree height. In axiom-driven mode, the proof structure and the corresponding MGT are both outputs whose values can be cached. In goal-driven mode, only the proof structure is an output, while the goal formula is an instantiated input. The overall operation is an iterated interplay of both modes on increasing levels: Basically the axiom-driven mode is performed, but before solutions for a new level are computed and cached, the goal-driven mode is invoked, at the new level and, depending on the configuration, possibly at a number of increasingly higher levels. In both modes, subproblems on lower levels can be solved by accessing the cache with previously computed proof-and-lemma pairs.

The extreme of a purely goal-driven configuration acts much like a goal-driven clausal tableau prover with iterative deepening. A purely axiom-driven configuration just generates lemmas, consequences, from the given axioms. The axiom-driven component in particular enables heuristic restrictions based on the MGTs, the lemma components of the cache entries, e.g., with limiting their size, restricting the number of different proofs per MGT, or limiting the overall number of cache entries where some ordering based on the MGT may determine the entries to be kept. In purely goal-driven configurations – and in goal-driven clausal tableau provers alike – MGTs are not materialized and hence not available as basis of heuristic restrictions. For condensed detachment problems, the blending of goal- and axiom-driven structure enumeration leads to success rates that drastically improve on goal-driven clausal tableau provers and compare to the lower end of state-of-the-art provers, while proofs are relatively short.

3 Implementation and Experiments

CD Tools includes two provers, *SGCD* (*Structure Generating theorem proving for Condensed Detachment*) and *CCS* (*Compressed Combinatory Structures*) that roughly address Theses 2 and 1, respectively. Most experiments so far were performed on the 196 problems in *TPTP 8.0.0* that are condensed detachment problems satisfying certain further constraints [27]. The *TPTP* rates 189 of these lower than 1.00 and 151 with 0.00. Clausal tableau provers are known to prove 92 of the 196 problems [27].[1]

With the approach of Thesis 2, 176 problems can be proven in different configurations of *SGCD* [27, 28] for level characterizations by number of tree nodes and height. The resulting proofs are typically rather small. The set of 89 problems provable by two purely goal-driven configurations of *SGCD* is, as expected, very similar to the set of 92 problems provable with clausal tableaux. In further experiments, *SGCD* was configured with a novel level characterization of the full binary trees used as proof structures that was motivated by observations at a human formal proof [29, 27]: The trees at level 0 are single nodes representing axioms. The trees at a level $n + 1$ are those where the left or right child is the root of a tree at level n and the other child is the root of a (not necessarily strict) subtree of its sibling or an arbitrary tree at level 0. In largely axiom-driven configurations this leads to 153 proven problems, apparently with proofs of small *compacted size* (size of the minimal DAG for the tree, or number of distinct compound subterms [29, 25]), also for problems where systematic search for minimal compacted size seems not feasible.[2]

CCS, the second prover in *CD Tools*, performs iterative deepening on compacted size of the proof structures and can incorporate, as suggested by Thesis 1, compressions with combinators and proof schemas, proof structure patterns defined by combinator terms. So far it was tried with exhaustive search, i.e., without heuristic restrictions, in purely goal-driven mode. Search for proofs with guaranteed minimal compacted size [25] succeeds for 86 problems. For 79 problems it is, moreover, possible to obtain all proofs with minimal compacted size. To get an idea of compression possibilities with the combinator approach and to see which particular combinators seem useful for proofs from applications, proofs obtained by *SGCD* and *CCS* for 176 problems were first compressed into tree grammars with *TreeRePair* [11], an advanced tool targeted at XML compression, and then, converted via λ-terms to combinator terms with a method from the implementation of functional programming languages [19, Chap. 16].

Concerning proof search with combinators, experiments were performed with configurations characterized by sets of proof schemas, which succeeded on 88 problems, including 6 on which the search for an "uncompressed" proof with minimal compacted size failed. Proof search with *CCS* was also tried on general Horn problems, the 562 problems of *TPTP specialist class* `CNF_UNS_RFO_NEQ_HRN`, of which 549 are rated lower than 1.00, 425 with 0.00, and around 430 are provable by clausal tableaux.[3] In five configurations with sets of proof schemas, some corresponding to specific forms of resolution, *CCS* – configured for goal-driven exhaustive search with iterative deepening upon compacted size – proves 421 of these, including 67 rated between 0.25 and 0.50, with a large overlap with those provable by clausal tableaux.

Acknowledgments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 457292495. The work was supported by the North-German Supercomputing Alliance (HLRN).

[1] *SETHEO 3.3* [13], *S-SETHEO* [10], *lazyCoP 0.1* [22] and *SATCoP 0.1* [21] together prove 76 problems according to the *ProblemAndSolutionStatistics* document of the *TPTP*. *leanCoP 2.1* proves 50 problems and *CMProver* [26] in different configurations proves 89 problems [27].

[2] Problem *LCL038-1* belongs to these. For this problem, which, upon suggestion in [20], was considered often in ATP and whose human proofs were analyzed in [29], *SGCD* found a proof with compacted size 22 [27].

[3] E.g., 414 by the four provers accounted in *ProblemAndSolutionStatistics* that were mentioned in footnote 1.

References

[1] W. Bibel. *Automated Theorem Proving.* Vieweg, Braunschweig, 1982. Second edition 1987. doi:10.1007/978-3-322-90102-6.

[2] W. Bibel and E. Eder. Methods and calculi for deduction. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1, chapter 3, pages 67–182. Oxford Univ. Press, 1993.

[3] W. Bibel and J. Otten. From Schütte's formal systems to modern automated deduction. In R. Kahle and M. Rathjen, editors, *The Legacy of Kurt Schütte*, chapter 13, pages 215–249. Springer, 2020. doi:10.1007/978-3-030-49424-7_13.

[4] E. Eder. A comparison of the resolution calculus and the connection method, and a new calculus generalizing both methods. In E. Börger, H. Kleine Büning, and M. M. Richter, editors, *CSL '88*, volume 385 of *LNCS*, pages 80–98. Springer, 1989. doi:10.1007/BFb0026296.

[5] E. Eder. *Relative Complexities of First Order Calculi.* Vieweg, Braunschweig, 1992. doi:10.1007/978-3-322-84222-0.

[6] M. Färber, C. Kaliszyk, and J. Urban. Machine learning guidance for connection tableaux. *J. Autom. Reasoning*, 65(2):287–320, 2021. doi:10.1007/s10817-020-09576-7.

[7] C. Kaliszyk and J. Urban. FEMaLeCoP: Fairly efficient machine learning connection prover. In M. Davis, A. Fehnker, A. McIver, and A. Voronkov, editors, *LPAR-20*, volume 9450 of *LNCS (LNAI)*, pages 88–96. Springer, 2015. doi:10.1007/978-3-662-48899-7_7.

[8] R. Letz. *Tableau and Connection Calculi. Structure, Complexity, Implementation.* Habilitationsschrift, TU München, 1999. Available from http://www2.tcs.ifi.lmu.de/~letz/habil.ps, accessed Jun 30, 2022.

[9] R. Letz, J. Schumann, S. Bayerl, and W. Bibel. SETHEO: A high-performance theorem prover. *J. Autom. Reasoning*, 8(2):183–212, 1992. doi:10.1007/BF00244282.

[10] R. Letz and G. Stenz. Model elimination and connection tableau procedures. In A. Robinson and A. Voronkov, editors, *Handb. of Autom. Reasoning*, volume 1, pages 2015–2114. Elsevier, 2001.

[11] M. Lohrey, S. Maneth, and R. Mennicke. XML tree structure compression using RePair. *Inf. Syst.*, 38(8):1150–1167, 2013. System available from https://github.com/dc0d32/TreeRePair, accessed Jun 30, 2022. doi:10.1016/j.is.2013.06.006.

[12] D. W. Loveland. *Automated Theorem Proving: A Logical Basis.* North-Holland, Amsterdam, 1978.

[13] M. Moser, O. Ibens, R. Letz, J. Steinbach, C. Goller, J. Schumann, and K. Mayr. SETHEO and E-SETHEO – the CADE-13 systems. *J. Autom. Reasoning*, 18(2):237–246, 1997. doi:10.1023/A:1005808119103.

[14] J. Otten. Restricting backtracking in connection calculi. *AI Communications*, 23(2-3):159–182, 2010. doi:10.3233/AIC-2010-0464.

[15] J. Otten. MleanCoP: A connection prover for first-order modal logic. In S. Demri, D. Kapur, and C. Weidenbach, editors, *IJCAR 2014*, volume 8562 of *LNCS (LNAI)*, pages 269–276. Springer, 2014. doi:10.1007/978-3-319-08587-6_20.

[16] J. Otten. Non-clausal connection-based theorem proving in intuitionistic first-order logic. In C. Benzmüller and J. Otten, editors, *ARQNL 2016*, volume 1770 of *CEUR Workshop Proc.*, pages 9–20. CEUR-WS.org, 2016. URL: http://ceur-ws.org/Vol-1770/paper1.pdf.

[17] J. Otten. The nanoCoP 2.0 connection provers for classical, intuitionistic and modal logics. In A. Das and S. Negri, editors, *TABLEAUX 2021*, volume 12842 of *LNCS (LNAI)*, pages 236–249. Springer, 2021. doi:10.1007/978-3-030-86059-2_14.

[18] J. Otten and W. Bibel. leanCoP: lean connection-based theorem proving. *J. Symb. Comput.*, 36(1-2):139–161, 2003. doi:10.1016/S0747-7171(03)00037-3.

[19] S. L. Peyton Jones. *The Implementation of Functional Programming Languages.* Prentice Hall, 1987.

[20] F. Pfenning. Single axioms in the implicational propositional calculus. In E. Lusk and R. Overbeek, editors, *CADE-9*, volume 310 of *LNCS (LNAI)*, pages 710–713. Springer, 1988. doi:10.1007/BFb0012869.

[21] M. Rawson and G. Reger. Eliminating models during model elimination. In A. Das and S. Negri, editors, *TABLEAUX 2021*, volume 12842 of *LNCS (LNAI)*, pages 250–265. Springer, 2021. doi:10.1007/978-3-030-86059-2_15.

[22] M. Rawson and G. Reger. lazyCoP: Lazy paramodulation meets neurally guided search. In A. Das and S. Negri, editors, *TABLEAUX 2021*, volume 12842 of *LNCS (LNAI)*, pages 187–199. Springer, 2021. doi:10.1007/978-3-030-86059-2_11.

[23] M. Schönfinkel. Über die Bausteine der mathematischen Logik. *Math. Ann.*, 92(3–4):305–316, 1924. doi:10.1007/BF01448013.

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

[25] R. Veroff. Finding shortest proofs: An application of linked inference rules. *J. Autom. Reasoning*, 27(2):123–139, 2001. doi:10.1023/A:1010635625063.

[26] C. Wernhard. The PIE system for proving, interpolating and eliminating. In P. Fontaine, S. Schulz, and J. Urban, editors, *PAAR 2016*, volume 1635 of *CEUR Workshop Proc.*, pages 125–138. CEUR-WS.org, 2016. URL: http://ceur-ws.org/Vol-1635/paper-11.pdf.

[27] C. Wernhard. CD Tools – Condensed detachment and structure generating theorem proving (system description). https://arxiv.org/abs/2207.08453, 2022. doi:10.48550/ARXIV.2207.08453.

[28] C. Wernhard. Generating compressed combinatory proof structures – an approach to automated first-order theorem proving. In B. Konev, C. Schon, and A. Steen, editors, *PAAR 2022*, volume 3201 of *CEUR Workshop Proc.* CEUR-WS.org, 2022. Preprint: http://cs.christophwernhard.com/papers/css.pdf.

[29] C. Wernhard and W. Bibel. Learning from Łukasiewicz and Meredith: Investigations into proof structures. In A. Platzer and G. Sutcliffe, editors, *CADE 28*, volume 12699 of *LNCS (LNAI)*, pages 58–75. Springer, 2021. doi:10.1007/978-3-030-79876-5_4.

[30] S. Wolfram. *Combinators – A Centennial View*. Wolfram Media Inc, 2021. Accompanying webpage: https://writings.stephenwolfram.com/2020/12/combinators-a-centennial-view/, accessed Jun 30, 2022.

[31] Z. Zombori, J. Urban, and C. E. Brown. Prolog technology reinforcement learning prover (system description). In N. Peltier and V. Sofronie-Stokkermans, editors, *IJCAR 2020*, volume 12167 of *LNCS (LNAI)*, pages 489–507. Springer, 2020. doi:10.1007/978-3-030-51054-1_33.

[32] Z. Zombori, J. Urban, and M. Olsák. The role of entropy in guiding a connection prover. In A. Das and S. Negri, editors, *TABLEAUX 2021*, volume 12842 of *LNCS (LNAI)*, pages 218–235. Springer, 2021. doi:10.1007/978-3-030-86059-2_13.