Experiments With Connection Method Provers

Wolfgang Bibel[1] and Jens Otten[2]

[1] Darmstadt University of Technology, bibel@gmx.net
[2] University of Oslo, jeotten@ifi.uio.no

In this presentation we are focussing on Automated Theorem Proving (ATP). ATP traditionally has been part of AI, a discipline which is in the headlines these days as never before. It is therefore a natural question whether or not ATP might take advantage of the current exorbitant efforts to advance AI R&D. We begin with presenting arguments which favor an ongoing substantial role for ATP within AI. The arguments engage a perspective of natural intelligence and its structure as evolved over mankind's thousands of years long history.

In recent years, AI often is identified just with learning systems, not only in the public but also within IT. In fact, learning systems have achieved such impressive results that the mainstream in the research community currently expects the creation of an artificial intelligence to be achieveable more or less exclusively by way of learning.

In the early days of AI it was said that it would be much easier to build a system simulating the work of a math professor than the intelligence of a child. After the recent breakthroughs in learning this judgment is no more tenable since the child's learning could now be simulated to some extent by some appropriate learning system. The math professor, in addition to his or her abilities learned as a child, invests a lot into his or her competence to reason logically in a correct and scientific way on the basis of knowledge acquired in a disciplined manner. The simulation of a math professor hence will require a combination of techniques including learning as well as knowledge representation and reasoning (KR) in whatever form, not least in that of ATP. Thus, ATP will remain of central importance for AI also in the forseeable future.

A more detailed exposition of this analysis of the development of the human behavior in everyday circumstances as well as in professional achievements leads to new lines of research into the details of the composition forming the human mind and in consequence the human behavior. Such insights influence the way ATP is realized and integrated in the simulation of mathematical research. One of the goals of the remaining presentation consists in pointing to some of the directions of research deriving from these insights.

Thus encouraged to continue research in a *suitably modified ATP*, we present the basics of the Connection Method (CM), its implementation through a family of theorem provers, results of experiments with these and challenges for future research in the directions just indicated. The history of ATP has led to at least three different proof methods: Resolution, Tableaux, and CM. Among these the CM features the following advantages.

It has been shown [1] that (an extended version of) the CM can linearly simulate Resolution. Thus, the performance of any resolution prover could as well be realized on the basis of the CM. Also, in contrast to Tableaux, the CM performs an extremely compressed proof search [5]. So in terms of performance, the CM in principle is at least as powerful as its main competitors. In addition, it offers the following striking advantages.

The CM is unique in that it performs the proof search in terms of the structural features of the very formula which is to be proved (illustrated in Fig. 1) rather than destroying this original structure by adding instances of subformulae (in case of resolution or instance-based methods) to or decomposing (in case of tableaux) the original formula. This *formula-orientedness* minimizes the data to be manipulated in the proof search and thus, in principle, offers a comparatively optimal mechanism in comparison with the competing proof methods (just compare the manipulated data in a proof with your favorite method with that in Fig. 1 for the same formula). In other words, the CM offers a potentially higher *performance* than its competitors. Due to a strict *goal-orientedness* during the *connection-driven* proof search, both suggested by the formula-orientedness, current CM systems do exhibit a comparatively strong performance.

Apart from performance the CM proof search is covering many logics in a completely uniform way, rendering *uniformity* as the second striking advantage. There is a third, yet hardly exploited major advantage of the CM over the competitive methods. Namely, the formula-orientedness enables the proof process to take a *global* view over the object of analysis, an aspect which still offers a great potential for being tapped and exploited in future research.

The talk introduces the CM in an illustrative way referring the reader to the literature such as [2, 3, 1, 5] for any details. In this abstract we just show a connection proof displayed in Fig. 1 for a simple first-order formula F_1, consisting of a spanning set of two connections along with a substitution unifying the connected literals.

$$\exists a (Pa \vee \neg \exists x\, Qx) \rightarrow \exists y\, Py \vee \forall b\, \neg Qfb \qquad \text{with} \quad \sigma = \{x_1 \backslash fb, y_1 \backslash a\}$$

Fig. 1. The connection proof for F_1.

We then briefly discuss connection calculi, the adequate tools for proof search in the CM, and their underlying principles leading to the goal-orientedness mentioned above. A basic connection calculus for first-order logic (fol) involving backtracking was developed and implemented in high-level PROLOG [12, 18]. The program, called leanCoP, although comprising only a few PROLOG clauses, shows a performance comparable to competitive theorem provers consisting of hundreds of thousands of lines of code [18, 11]. Due to its high-level code its correctness, in contrast to the large systems, can be and has been verified which we regard as an important issue for proof systems [19]. leanCoP operates on formulas in clause form, not on fol formulas in their original form as illustrated in Fig. 1. A variant of leanCoP has been realized as an OCaml version [9].

A connection calculus for skolemized, but non-clausal fol formulas has also been developed and implemented in the style of leanCoP as a system called nanoCoP [13, 16] showing again a comparatively impressive performance. nanoCoP performs the proof search directly on the structure of the original formula, no translation steps to any clause form are necessary. It combines the advantages of more natural non-clausal calculi (eg. sequent calculi), with the goal-oriented efficiency of a connection-driven proof search.

Hence, it is also significantly easier, to translate the resulting non-clausal proofs back into a human-readable form.

The CM's unique uniformity bears its fruits in porting the proof-search technology developed in one logic into other logics. This way the fol technologies built into leanCoP and nanoCoP have been ported to many other logics resulting in a uniform set of systems such as ileanCoP and nanoCoP-i for intuitionistic fol, and MleanCoP and nanoCoP-M for modal fol, thus forming a large family of uniformly designed and powerful provers [11, 14, 15, 17]. The CM is the unique and unrivalled proof-method in ATP which features such a wide variety of systems, many of which are outperforming any of its competitors.

At the outset of this abstract we have argued for a *suitably modified ATP*. In the presentation we will discuss a few elements in the modifications needed for an ATP that takes into account the progress in the pertinent scientific disciplines. First of all, ATP needs to base its research on the best possible proof-method available. As our research, briefly outlined above, has demonstrated in various ways the CM offers the best possible basis for the development of powerful systems. So, the time has come for the community to take a major step towards a change in the method of choice.

This step involves many different aspects. One of those is a greater flexibility in terms of the underlying logic in particular applications. As we demonstrated the CM supports this flexibility in the best possible way. Another important aspect refers to the form of the problems to be proved. While it has become standard in ATP to simplify matters by using clause form, more advanced techniques have exposed the considerable disadvantages going along with these simplifications (see eg. [20]). As we have seen, the CM again supports the avoidance of these disadvantages.

In the latter respect, we have experimentally compared problems stated in non-clausal form with their transformed versions in clausal form with respect to runtime and resulting proof size. As one of the results it turns out that the length of the returned proofs, in terms of the number of connections, is – on average – significantly shorter in the case of the non-clausal CM of nanoCoP compared to the clause form CM of leanCoP. Furthermore, nanoCoP does not only prove problems not proved by leanCoP, but also a few hundred problems not proved by one of the fastest provers available today.

Among the modifications towards a modern ATP certainly is the integration of learning techniques into ATP systems, as initiated by members of our research group already in the 1980's in the context of our proof system SETHEO [6, 7]. Although the integration of learning into ATP by now has become a rather active area of research [21, 8], it is far from clear how exactly learning techniques could best support the work in ATP. It is clear that the way of their integration may strongly differ from one kind of problems to another.

In order to illustrate this difficulty, we discuss a problem due to Łukasiewicz [10], $\forall xyzu\, Pi(i(ixy,z),i(izx,iux)) \;\wedge\; \forall vw\,(Pv \wedge Pivw \rightarrow Pw) \;\rightarrow\; \forall abc\, Pi(iab,i(ibc,iac))$. While Łukasiewicz was able to find a proof for it by hand, current provers need tens of thousands or even millions of search steps to discover a proof. Also, the effect of applying a standard learning system to the prover E for learning a better strategy leads to limited improvement only [4]. What kind of artificial intelligence is needed to improve our systems to a point where they are able to find the proof of this or other problems in a more direct way similar to that taken by Łukasiewicz?

References

1. Bibel, W., Eder, E.: Methods and calculi for deduction. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 1, chap. 3, pp. 71–193. Oxford University Press, Oxford (1993)
2. Bibel, W.: Automated Theorem Proving. Vieweg Verlag, Braunschweig, second edn. (1987), first edition 1982.
3. Bibel, W.: Deduction: Automated Logic. Academic Press, London (1993)
4. Bibel, W.: A vision for Automated Deduction rooted in the connection method. In: Schmidt, R., Nalon, C. (eds.) The 26th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2017). LNAI, vol. 10501, pp. 3–21. Springer (2017), dOI: 10.1007/978-3-319-66902-1_1
5. Bibel, W., Otten, J.: From Schütte's formal systems to modern automated deduction. In: Kahle, R., Rathjen, M. (eds.) The Legacy of Kurt Schütte. Springer (to appear)
6. Ertel, W., Schumann, J.M., Suttner, C.B.: Learning heuristics for a theorem prover using back propagation. In: Retti, J., Leidlmair, K. (eds.) 5. Österreichische Artificial-Intelligence-Tagung: Igls/Tirol, 28.-31. März. pp. 87–95. Springer, London (1989)
7. Goller, C.: A Connectionist Approach for Learning Search-Control Heuristics for Automated Deduction Systems. Akademische Verlagsgesellschaft AKA, Berlin (1999)
8. Kaliszyk, C., Urban, J.: FEMaLeCoP: Fairly efficient machine learning connection prover. In: Davis, M., Fehnker, A., McIver, A., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2015). Lecture Notes in Computer Science, vol. 9450, pp. 88–96. Springer, Berlin, Heidelberg (2015)
9. Kaliszyk, C., Urban, J., Vyskočil, J.: Certified connection tableaux proofs for HOL Light and TPTP. In: CPP '15 Proceedings of the 2015 Conference on Certified Programs and Proofs. pp. 59–66. ACM (2015)
10. Łukasiewicz, J.: The shortest axiom of the implicational calculus of propositions. Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences 52, 25–33 (1948), https://www.jstor.org/stable/20488489
11. Otten, J.: *leanCoP* 2.0 and *ileanCoP* 1.2 – High performance lean theorem proving in classical and intuitionistic logic (system descriptions). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS, vol. 5195, pp. 283–291. Springer, Heidelberg (2008)
12. Otten, J.: Restricting backtracking in connection calculi. AI Communications 23, 159–182 (2010)
13. Otten, J.: A non-clausal connection calculus. In: Brünnler, K., Metcalfe, G. (eds.) TABLEAUX 2011. LNAI, vol. 6793, pp. 226–241. Springer, Heidelberg (2011)
14. Otten, J.: Implementing connection calculi for first-order modal logics. In: Ternovska, E., Korovin, K., Schulz, S. (eds.) 9th International Workshop on the Implementation of Logics (IWIL 2012). Merida, Venezuela (2012)
15. Otten, J.: *MleanCoP*: A connection prover for first-order modal logic. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) Automated Reasoning, 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 19-22, 2014, Proceedings. LNCS, vol. 8562, pp. 269–276. Springer, Heidelberg (2014), ISBN 978-3-319-08586-9
16. Otten, J.: *nanoCoP*: A non-clausal connection prover. In: Olivetti, N., Tiwari, A. (eds.) International Joint Conference on Automated Reasoning, IJCAR 2016. LNAI, vol. 9706, pp. 302–312. Springer, Heidelberg (2016)
17. Otten, J.: Non-clausal connection calculi for non-classical logics. In: Schmidt, R., Nalon, C. (eds.) 26th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods. pp. 209–227. LNAI 10501, Springer, Cham, Switzerland (2017)

18. Otten, J., Bibel, W.: *leanCoP*: Lean connection-based theorem proving. Journal of Symbolic Computation 36, 139–161 (2003)
19. Otten, J., Bibel, W.: Advances in connection-based automated theorem proving. In: Bowen, J., Hinchey, M., Olderog, E.R. (eds.) Provably Correct Systems, pp. 211–241. Springer, London (2016)
20. Urban, J., Vyskočil, J.: Theorem Proving in Large Formal Mathematics as an Emerging AI Field. In: Automated Reasoning and Mathematics – Essays in Memory of William W. McCune, Maria Paola Bonacina and Mark E. Stickel (eds.), Lecture Notes in Computer Science, vol. 7788, pp. 240–257. Springer (2013)
21. Urban, J., Vyskočil, J., Štěpánek, P.: MaLeCoP Machine learning connection prover. In: Brünnler, K., Metcalfe, G. (eds.) 20th International Conference, TABLEAUX 2011, Bern, Switzerland, July 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6793, pp. 263–277. Springer, Berlin (2011)