Comparison of Proof Methods

Wolfgang Bibel[1]

[1]*Technical University of Darmstadt, Germany*

Abstract

The paper analyses and compares on a high level of abstraction significant features of the leading proof methods in Automated Theorem Proving. It aims at the identification of promising areas of research in order to further advance the field. The evaluation yields relevant evidence for promising such a substantial advance by solving open research questions in the Connection Method. The most striking feature thereby is the method's unique formula-orientedness which recently has led to a notable success by the system SGCD based on it.

Keywords

Automated Reasoning, Automated Theorem Proving, Connection Method, Resolution, Tableaux Method

1. Introduction

The discipline of Automated Reasoning (AR), or Automated Theorem Proving (ATP), derives its central importance for the science of Artificial Intelligence (AI) from the following three fundamental assumptions. A1: Knowledge of any kind can be represented in a language with a precisely definable semantics. A2: Logical languages provide a close-enough approximation for such a knowledge representation language. A3: Logical reasoning on the basis of such logical languages provides the key mechanism for crucial tasks involved in the various applications of knowledge and its handling including consistency-checking of knowledge bases as well as their extensions by reasoning.

Mathematics due to its wide-ranging success in the sciences and their applications provides convincing evidence for the validity of these fundamental assumptions. For this reason, ATP systems have primarily been tested paradigmatically on mathematical knowledge although its mechanisms apply to any kind of knowledge and are not at all restricted to the mathematical one.

There is a variety of logical languages known for representing knowledge. First-order logic (fol) may be seen as the language exhibiting core features shared by most others. Hence, fol along with its inferential structure has been studied so extensively. In this paper we will follow this attitude and restrict the discussion to fol. This allows us to assume that the reader is familiar with this widely known formal language.

Along with the underlying language any logic features some mechanism which to some generality allows to identify true resp. valid statements among all possible ones. In Logic these

CEUR Workshop Proceedings (CEUR-WS.org)

are called formal systems while in ATP we refer to them as proof calculi. Numerous such calculi have been developed in ATP. Many of them are based on the resolution rule, others are variants of tableaux, further ones derive from the Connection Method (CM), and so forth.

The main topic of this paper is a high-level analysis of their differences, virtues, advantages, disadvantages, and the resulting perspectives for our field. Thereby we will focus on the three clusters of calculi just mentioned. Let us refer to these clusters of calculi as RM, TM, and CM.

The subsequent section will present the intended analysis of these three methods. The section thereafter will point out several aspects of the CM which still are waiting for being explored and realized in competitive proof systems. The most important aspect thereby is the separation of the structural work in proof search from the formula to be proved. This separation allows a global guidance of the search from all proof-specific syntactic features of the formula as demonstrated by the system SGCD in a first step into such a direction. We outline the resulting research perspective in a first subsection. Apart from pointing to further hints in the literature towards advancing the CM, we then outline the special handling of equality within the CM in a second subsection, an important gap still waiting to be closed in a rather straightforward way simulating the analog techniques successfully applied in the RM. Conclusions complete the paper.

2. A comparative analysis of proof methods

The concept of *proof* has been developed, used, and refined in Mathematics for more than two thousand years. Establishing the validity of a statement, here assumed to be a formula F in fol, is the goal of a proof. Generally, F is of the form $A_1 \wedge \ldots \wedge A_n \to T$, $n \geq 0$. The A_i's are the axioms and T is the theorem to be established by the proof. "Axioms" here and in the following is meant as an abbreviation for "axioms, theorems, lemmata, or assumptions".

Typically, mathematical proofs are presented in a semi-formal natural language text, ie. in a linear format, and involves some sort of logical/mathematical proof steps which have evolved by tradition over the centuries. Gerhard Gentzen has precisely formalized this semi-formal style in his seminal paper [1]. Among others it features his well-known Kalkül des natürlichen Schließens, NK, or calculus of natural deduction, which is modelling the traditional proof steps in a formal way. Gentzen's formal system LK [1, p.192f], a technical modification of NK, features 19 rules ("Schlußfiguren") including the so-called cut rule. Gentzen proved in his Hauptsatz that this rule is admissible, ie. that it can be dispensed with without affecting the amount of provable formulas.

This *cut elimination* leads to longer proofs, in fact exponentially longer proofs in the worst case [2, Sect. 5.2]. This increase of the proof lengths is therefore an important, even crucial issue for proof search. Imagine a competition of two systems one searching for the short proof of a theorem achievable with cut, the other for the proof of the same theorem, but without cut and for the exponentially longer proof. It is obvious that the latter system starts at an unassailable disadvantage.

This observation leads to the first fundamental distinction of proof calculi: calculi with cut or some equivalent feature vs. calculi without, shortly C_c vs. C_{-c}. Thereby, the calculi with property C_c have an exponential advantage over those in C_{-c} for some set of theorems. LK

belongs to C_c unless the cut-rule is deleted in which case it belongs to C_{-c}.

Proofs in Gentzen's systems are trees with axioms at the leaves and the theorem at the root. In a more compressed form they might also be represented as dags, ie. directed acyclic graphs. Several proof calculi developed thereafter took Gentzen as sort of a reference point. In principle, all complete and consistent proof systems for fol related with LK are equally suitable for realizing the automatic search for proofs. There is, however, the crucial criterion just mentioned putting a subset among these proof systems at a serious disadvantage; the subset consists of those inheriting the just mentioned exponential expansion of proof lengths.

Since TM calculi are derived from LK without cut by way of inverting its rules, C_{-c} holds for TM and thus suffers from that serious disadvantage. It also holds for most CM calculi realized in implemented systems like leanCoP. In contrast, RM calculi involve the cut since resolution amounts to sort of a cut: resolution on ground clauses is a version of the cut rule restricted to atomic formulas. Each resolution step involves the generation of a lemma which can be used multiple times in the search for a proof, a feature closely related with cut. At our high level of discussion we may thus say that RM calculi belong to C_c, without going into the nontrivial details of this statement (see eg. [3, 4]). C_c holding for RM has been considered as a significant advantage in comparison with TM.[1]

In [5, Sect. 2.10] a CM calculus has been presented which involves connection structures in addition to simple connections. For this calculus it has been proved that it can linearly simulate the resolution calculus, and vice versa. In other words, both calculi are computationally equally powerful. For the CM as a family of calculi this means that its assumed major weakness in comparison with RM of lacking cuts does in fact *not* hold. In short, for the CM we rather have C_c. The first ATP system with such connection structure features is CCS [6]. First promising results achieved with it in its interplay with the system SGCD [7] have been reported in [8].

In human mathematical proofs the proof steps typically consist in relatively small changes of the representational state. This takes into account the limited computational capabilities of humans in carrying out such a step in a precise way and supports the minimization of errors. To a large extent this feature, noted as C_{-s}, is modelled in some way or another by most logical calculi. They modify in each step the actual state, represented by some derived formula, in some relatively small syntactic way. This includes the characteristic that the deductive structure is combined with some formula structure, at any rate not separating these two different structural aspects which is expressed by the "$-s$".

The unique exception is CM whose calculi during the course of a proof search leave the formula itself completely untouched. This is because they strictly separate the proof structural information from the formula, ie. for CM we have C_s, a significant distinction. While all other calculi involve the superfluous ballast of formulas associated with each state in proof search, the CM has been freed completely from this burden which thus is totally unnecessary from a computational point of view and is meaningful only with regard to our human understanding. This separation of the proof structure from the formula at the same time allows to focus the proof search on all structural features inherent in the formula. Thereby proof structure includes all structural information possibly leading to proofs, not only the finally found proof, of course. This includes, eg., alternative spanning sets of connections, variable multiplicities, term

---

[1]The table towards the end of the present section might be helpful for the reader in overlooking this discussion.

transformations involved with connections, the discrepancy between goal and axiom terms to be bridged, etc. A *global* view may be taken on all this, as we termed it in [9]. In other words, the search can be done in a *formula-oriented* manner as we denoted it in [9]. We consider these advantages of equal relevance as those going along with the cut. They have been demonstrated in a first step by the already mentioned system CCS/SGCD.

LK is of a *generative* type, as it is called in the formal languages literature. Any proof in it starts with axioms and ends with the given theorem. For the search for a proof of a theorem the opposite direction, namely starting with the theorem, is much more promising for rather obvious reasons which are not repeated here. For enabling this switch of direction an *analytic* type of calculus is asked for which starts from the theorem in a *goal-oriented* manner, shortly C_g, and ends up with axioms. Tableaux calculi belong to this family. They have been derived from LK roughly by inverting its deductive rules along with certain simplifications and modifications in detail. Since inverting the cut rule would involve a guess for the cut formula without any guidance whatsoever, TM calculi have dispensed with that rule altogether, ie. TM calculi as such are members of C_{-c} as already noted above.

RM calculi start out with the proof problem as a whole, ie. including axioms and theorem; more precisely, they do so in the simplified format as a set of clauses. RM therefore features the flexibility to operate in both directions in a mixed manner, shortly C_m, which may include C_g.[2]

As presented in [10, Sect. 13.2] in formal details, the CM is a method abstracted from TM (and LK) so that all these are closely related. Hence, for the CM calculi published until recently in the literature we have C_g, ie. it shares this property with TM. As noted above CM and TM differ in that we have C_s for CM, but C_{-s} for TM.[3] For a computational comparison all this amounts to a clear advantage of the CM over TM. Since otherwise both methods are so closely related, in the important computational context we may thus set aside TM in our comparative analysis as being represented by the computationally superior CM. In what follows we will therefore reduce our discussion to CM vs. RM.

For CCS/SGCD mentioned above, and thus also for CM, we also have C_m which gives the flexibility of combining goal-oriented search with lemmata derived from the axioms. In fact, C_m has turned out to become the crucial improvement for CM calculi in order to achieve the decisive property C_c for them. For that reason it has been included into the list of the three properties discussed here. Altogether, the comparison of CM with RM so far has thus resulted in an advantage for CM because of the favorable C_s while otherwise both methods share the same relevant computational features C_c and C_m.

Unless *P=co-NP* we have to assume that the complexity of any proof calculus is (at least) exponential. So, there will always be theorems for which any calculus will fail to find a proof for them within reasonable time limits. But it could well be the case that one calculus always fares better than another one. That this is definitely not the case for resolution vs. the CM has been shown in [11]. Namely, Haken has shown in [12] that resolution exhibits an exponential

---

[2]With resolution a goal-oriented search can be achieved with an appropriate set-of-support and an axiom-oriented search for instance with hyperresolution. Since RM calculi typically operate in an exhaustive manner, the distinction is of little relevance in this context.

[3]Disjunction (in the sense of the originally given formula) is handled differently in CM and TM; whether or not this bears an effect on the complexity of the two methods is unknown to the present author; given the way of different handling in both, it could only be one favorable for CM.

behavior for the so-called pigeonhole formulas while the present author has presented in [11] short CM proofs for exactly these formulas; recently a similar result was achieved for RM [13].[4] In other words, CM and RM so far are at least on an equal footing, except for the mentioned advantage of C_s for the CM.

The ATP community has established a standard for comparing proof systems by way of the CASC competition (along with TPTP).[5] The best system in any problem category is the one which, roughly, within a given time-limit proves most theorems. Sean Holden puts this on his homepage such: "*The automated theorem-proving (ATP) world likes to see things going faster and/or more things being proved. If your prover is implemented in Prolog it starts at a disadvantage.*"[6] This manner of comparison suits the inherent human attitude towards spectacular races and cheering winners. But obviously it involves too crude a measure of long-term comparison for revealing the true virtues or weaknesses of the underlying proof methods, ie. their potential performance in the future.

The winning method under this crude measure for many years has been RM which this way has achieved a monopoly in our field, in accordance with the well-known social principle: "*The winner takes all.*" Since computers have dramatically increased in speed and power along these years, so has the number of proved theorems. Yet, there are still many theorems proved by humans which have withstood any attempts to prove them by leading systems. Why that? It is not unlikely that it is this particular manner of comparison within CASC which is favoring in some way a possibly suboptimal research direction and through that might be suppressing support for promising alternatives. The analysis in this paper is meant to alert the community to this possibility – once again – in order to open its interest in competing directions (as I have vainly done at numerous occasions within the last decades).

The contradiction between our analysis favoring CM and the CASC favoring RM (in some way) has simple explanations. Building CM systems in earnest has started in the 1980s engaging since then altogether a mere few dozens of researchers. This is in sharp contrast with RM systems whose serious research and development started about two decades earlier engaging over the decades altogether thousands of researchers. For these sheer investmental reasons,

---

[4][13] fails to refer to [11].

[5]CASC was initiated in the 1990s by members of the present author's former Munich research group along with Geoff Sutcliffe during his collaborative visit of it. This footnote remark in the preceding sentence, contained already in the original version of the paper, was meant to indicate the present author's favorable attitude towards Geoff's enormous and extremely valuable contributions done within this enterprise for many decades. In response to one of the referees's "*objection/criticism*" concerning its subsequent critical assessment of CASC in the paper's main text I extend this remark with the following additional explanations.

First of all, the present author has supported TPTP/CASC in various ways, not least in my role as initiator, coordinator, and speaker of the DFG Focus Programme on Deduction in Germany (1992–1998) which included several substantial contributions to TPTP/CASC. But I stay with my assessment expressed in the main text that the present format of CASC implicitly favors the leading and popular proof methods in a too indiscriminate manner. I fully agree with the referee in that "*TPTP/CASC is agnostic wrt the proof methods used in the provers.*" But a comprehensive assessment in addition has to take into consideration the well-known human attitudes in reaction to the results of any such competition. Under such a more comprehensive view TPTP/CASC in its present format is focussing merely on quantitative measures without any consideration of further relevant information whatsoever and, of course, it is supporting the winners in an indirect way, even though the winning methods might not be the most promising ones in view of their future performance. Admittedly, a finer tuned way of competition is not easily achievable.

[6]https://www.cl.cam.ac.uk/ sbh11/, access 8.7.2023.

	$C_{\pm c}$	$C_{\pm s}$	$C_{m/g}$	*cmp*	*orf*	*unf*	*thr*
RM	C_c	C_{-s}	C_m				
TM	C_{-c}	C_{-s}	C_g		√	√	√
CM	C_c	C_s	C_m	√	√	√	√

Legend: RM resolution; TM tableaux; CM connection method; $C_{\pm c}$ with/without cut; C_s formula and proof separated; C_{-s} formula and proof not separated; C_m mixed manner operation; C_g goal-oriented operation; *cmp* compact proof structure; *orf* formula in original form; *unf* uniformity over different logics; *thr* ease of transformation to a human-readable text.

Table 1
Summary of Comparison

the RM technology this way has advanced far ahead of the CM one, of course. Under these circumstances it is even remarkable how relatively successful the CM has become none the less. Let us mention some of these achievements.

CM allows the proof search for theorems in their original form. This makes the transformations to normal form including Skolemization superfluous which offers a lot of advantages. Also, the CM generalizes to logics other than fol in a rather straightforward way. The separate proof structures are quite similar in all these variants. In consequence, a system like leanCoP [14] has required only marginal changes to mutate to a non-clausal form prover [15], to an intuitionistic prover [16], to a modal logic prover [17], and so forth. In [9] we introduced the term *uniformity* for denoting this advantageous property of CM. Last not least, due to its close relationship with LK – shared with TM – a discovered CM proof can be transformed in a straightforward way into a human-readable text, a further important advantage. The discussion of this section is summarized in Table 1.[7]

3. Research tasks in the pipeline

Depending on the taken perspective the state of the art in ATP may be regarded as extremely impressive or relatively disappointing. The fact that its systems are able to prove tens of thousands of theorems, many of them within a few seconds is impressive indeed. The fact, however, that there are still theorems with proofs found by humans within a reasonable amount of time, but still untractable for our systems is certainly disappointing. It demonstrates that the underlying methodology is still lacking of possibly fundamental features. In the present section I will try to point to features that I feel would advance the field if explored and exploited. After the result of the comparison in the previous section it is needless to mention that I am expecting progress especially by advancing the CM.

---

[7]The inclusion of such a table was proposed by one of the referees. While it certainly supports the view over the discussion of the various properties for the reader, it has to be interpreted with the following proviso. Each of the properties carries with it a longer story. The simplification to a binary interpretation suggested by table is not the whole story. Consider property *orf*, for instance. There is some work on generalizing resolution to work also for non-clausal form input; it resulted in concluding that resolution under several aspects suits much better to clausal input. The lack of √ in the table has to be interpreted in this more elaborate manner and similarly for the other entries.

$$Pi(\overbrace{i(i(ixy,z),i(izx,iux))}^{\overset{4}{\overset{5}{\frown}}} \wedge (\underbrace{Pv \wedge Pivw}_{\underset{3}{\underset{2}{\smile}}} \overset{1}{\to} Pw) \to Pi(iab,i(ibc,iac))$$

Figure 1: Łukasiewicz with its five unifiable connections.

3.1. Exploiting structural features

The characteristic of CM to separate the deductive structure from the formula, ie. C_s, opens the door for studying and exploiting this structure in a global way wrt. the given problem. This has been demonstrated in a series of recent papers [18, 8, 19]. In order to simplify the underlying research task to a reasonable state of tractability for the beginning, these papers have restricted the study to the class of fol formulas known as *condensed detachment problems* (CD). A well-known theorem of this class is the following one due to Łukasiewicz. It is shown in Figure 1 along with its five unifiable connections c_1, c_2, c_3, c_4, c_5. In fact, there is one more connection, c_0, connecting the axiom with the goal which, however, is not unifiable for which reason it is not shown explicitly. All CD formulas have the same structure $A \wedge DET \to G$ illustrated by the Figure whereby A is the axiom, DET the detachment rule (or modus ponens), and G the goal. While CD is a rather restricted class, it already features most important aspects of general fol.

Any proof of such a CD formula consists of a number of instances of each of these five connections along with a global substitution of the instances of the variables $x, y, \ldots, w$. This global view on the structure of proofs is unique to the CM and expressed as C_s in the previous section. In consequence, this opens the door for novel and promising ways of proof search.

One such approach consists in enumerating these proof structures by the following list S:

$$\{c_0\}, \{c_1^{11}, c_4^{11}, c_5^{21}\}, \{c_1^{11}, c_2^{12}, c_4^{12}, c_5^{22}, c_5^{31}\}, \ldots$$

Recall that a connection is a pair of occurrences of literals in instances of formula parts or clauses.[8] The upper indices attached to the connections denote the pair of instances of the connected left and right clauses. For instance, c_2^{12} is the connection from the major premise of the first instance of DET to its conclusion in the second instance, and similarly for all others.

This enumeration is restricted to contain only spanning sets of connections. This guarantees that the identification of a unifiable element in the list proves the theorem.[9] In the case of CD formulas this restriction to spanning sets is easily guaranteed by associating with each incoming connection containing the conclusion of an instance of DET two outgoing connections from the two premises of this instance. In [18, 8, 19] the sets of S have been represented as so-called *D-terms* built from terms of the form $D(d_1, d_2)$. In correspondence with the members of S the D corresponds to the incoming connection and its two arguments to the outgoing ones (or, more precisely, to the corresponding occurrences along with their multiplicities in the formula).

---

[8]The reader not familiar with details of the CM is referred to the literature such as eg. [20].

[9]In an email of 26.Aug.2023 Peter Andrews has reminded the author of the possibility to involve Quantum Computing (QC) which might allow to test many list elements at the same time for unifiability. In his words: "*The problem of searching for complete sets of connections may be radically different when quantum computers are used.*" This possibility would speed up the proof process in a sensational manner.

So, we have two equivalent notations for exactly the same structure. However, the notation used in S is completely general for any kind of formulas, not only CD ones.

On the basis of this kind of proof structures SGCD mentioned in the previous section has achieved an impressive performance [8]. Using D-terms it enumerates such proof structures together with the unification of associated formulas. It also generates lemmata for supporting proof search in an axiom-oriented way, thus demonstrating the characteristic C_m. SGCD is the first proof system among all currently available provers which achieved a fully automatic proof for the TPTP problem LCL073-1, known as Meredith Single Axiom problem. Altogether it ranks among the very best systems within the CD domain. This excellent performance of SGCD demonstrates the importance of enhancing CM systems the way discussed here.

The steps taken into this direction by the research discussed here is, however, just a beginning. In the remaining of the subsection I shall outline several future research options. First of all, the sequence S may be enhanced in the following way. In its present form it lists what may be represented as full binary trees, for human understanding a more intuitive representation of the same thing. The different members of S may, however, share common substructures. If these are represented such that any such common substructure occurs only once in any element of S then its elements represent dags rather than trees. This way we may reduce the size of those proof structures considerably and speed up proof search accordingly. Let us assume from here on that S represents such a list of dags rather than trees.

Along with each such dag there is a substitution of its variables by unification of associated formulas, including the goal and the axioms. If the unification fails, like in the trivial case of $\{c_0\}$, the set may be discarded from the list without restricting generality. Such failure may be discovered in a strategic manner thus reducing the amount of unificational work involved. Also, the terms in A and G may give guidance towards limiting the size of the members of S, and thus reducing S. By closer inspection into the dynamics of the terms determined by the connection structure, involved strategies may be worked out resulting in preferences among the members of S, or such preferences might be learned by experiments. By noting such preferences the amount of search might be reduced considerably.

More generally, along with the given formula's connection structure CS like the five connections in Fig. 1 there is a transformation T of terms for each of the connections which yields the unification of the connected literals. The information I given by CS along with T is necessary to establish a proof, and at the same time it is also sufficient. Thus, the isolated information contained in I contains all information needed for finding a proof in a maximally condensed form. In other words, the property C_s from the previous section may be taken as to include the minimality of the separated information I. Exploiting the various aspects contained in I is the main task for finding proofs. Note that proof search in this generality is a fairly unexplored and unexploited domain of ATP research, as it has been invisible through the eyes of RM.

All these considerations and many further ones have the potential of improving CM systems designed along these lines in a substantial way well beyond the current state of the art in ATP. In our opinion it is a good research strategy to carry out all these improvements first for limited structures like that of CD formulas. The crucial test of course comes with the generalization of the gained insights for the limited cases eventually to arbitrary fol formulas, an enormous research task and challenge. Since a list like S or the characterizing information like I may be generated for any formula, as already mentioned above, there is no fundamental barrier for this

generalization.

All this shows that the development of CM is by far not completed at all. The ATP community is rather faced with substantial work ahead waiting to be undertaken. This amounts to promising work with a potential for advancing the field in a way which was not recognized in other methods due to their particular features, especially to their C_{-s} — in contrast to the C_s for CM.

3.2. Equality handling and others

In Mathematics equality is surely the most ubiquitous notion at all. It is worth a special handling not only for Mathematicians, but also for ATP systems. While there has been extensive work in the context of RM for handling equality for more than half a century, no analog work exists for the underdog CM. CM systems like leanCoP [14] handle equality in an axiomatic manner like any other relation. In consequence, CM systems are doing poorly in CASC competitions for any problem featuring equality, hence for many problems indeed.

In principle, transferring the existing techniques for handling equality from RM including superposition to CM in a sense might be a straightforward task. Yet, it requires top expertise and familiarity in both subfields of ATP, a rare constellation indeed, if existing at all anywhere in the world. With the necessary support it could be acquired none the less. In the hope that this will happen in the not-too-distant future I outline here a first high-level version of how equality could be incorporated more efficiently in CM.

In the books [20, V.3] as well as [21, 4.5] first steps towards such an incoporation have already been outlined. There the treatment is based on the three axioms of equality, namely reflexivity as well as substitutivity for functions and predicates. Typical treatments in the context of resolution rather use the three congruence axioms along with functional substitutivity [22]. While logically the two sets of axioms are equivalent, it is worthwhile to think about which of these suites the environment more adequately.

In the two just referenced books the concept of a connection is extended by so-called *eq-connections*. Eg. see the examples 3.2, 3.3, and 3.4 in [20, V.3] illustrating the concept. They allow the direct substitution of terms by equal terms in the given formula or matrix similarly to paramodulation in RM. They perfectly fit into the structural representation of proofs in CM and fulfil the preferred property C_s. The corollary 3.6 in that reference provides the theoretical basis for this approach.[10]

Without a clever strategic guidance for substituting equals by equals, this initial step is of little help however. What is needed is some ordering relation which imposes a direction to these possible replacements. Within RM the decisive technique to realize such a directed replacement without restricting generality has been presented in [23], optimized and polished in subsequent publications by the same and other authors (eg. [22]). In principle, the transfer of these techniques into the context of eq-connections within the CM seems to be a straightforward task. It would complement the comfortable way of dealing with equality by way of eq-connections with an ordering relation for restricting the prolific nature of substituting equals by equals. But,

---

[10]The present author is not aware of any work continuing this way of handling equality. Whatever the reason for this ignorance in the last four decades, it clearly demonstrates the lack of interest in the CM within the ATP community.

as already stated, carrying out such a transfer never the less requires high competence in two different demanding areas.

Unfortunately, there is little incentive for getting the performance of one system in view of a feature like equality to the level of a competing one by incorporating the latter's techniques. In fact, this unattractiveness is most likely the reason why this transfer has not been done up to now (but see [24] for a recent encouraging step into this direction). In the context of the perspectives outlined in the previous subsection the situation eventually might change, since in combination the respective efforts open up the perspective of CM systems outperforming RM systems.

The present author has outlined many further potential improvements in detail for the CM in the last decades. Here I may thus restrict myself to just mentioning the existence of these details in those publications. For instance, see [9, 25] and the references to earlier work given therein.

Recall the reference in Sect. 2 to Sean Holden's opinion concerning the programmming languages used in implementations. The use of Prolog for developing the CM further is fully appropriate in my opinion. But once the CM has reached a state competitive with that of its competitors the time will have come that a system will have to be implemented in some low-level programming language to speed-up the low-level efficiency in its performance matching that of leading RM systems.

Once such a stable system with high performance will be available, also the time has come for building a comfortable interface for the interaction with users, especially with mathematicians. Such an interface would present the discovered proofs to the users in a natural mathematical style of text. From that time on, mathematicians, Math-teachers, engineers etc. might finally become interested in a daily use of such a system within their work.

4. Conclusions

The paper has taken a high-level comparative look at the three major proof methods in ATP, namely resolution (RM), tableaux (TM), and the Connection Method (CM). Generally, the competition among different research directions is fruitful if it is done on fair grounds. We have argued that the community in ATP seems to be misguided by superficial advantages favoring RM. If one takes a closer look at qualitative characteristics of these competing methods, as done in this text, the arguments all rather favor the CM as the most promising of the three competitors.

To a certain degree, however, the CM is sort of still in its infancy, lacking important aspects to be worked out in much further detail. As described in Sect. 3.1 the most important aspect is the realization of an involved proof search along the information provided by the connection structure extractable from the formula to be proved. Further aspects like the special handling of equality are described in Sect. 3.2. For being ready to the substantial investments needed for advancing the CM in these directions, the community has first to become convinced that such an investment would advance ATP as such. The arguments presented in the text have tried to nourish exactly this kind of conviction.

Acknowledgments

The initiative to the Workshop AReCCa 2023 is due to Jens Otten which in turn triggered the writing of this text. I am grateful to him for this promotion of the CM as well as for decades of fruitful collaboration. Thanks are due to Christoph Wernhard and two anonymous referees for a number of helpful suggestions towards improving the paper.

References

[1] G. Gentzen, Untersuchungen über das logische Schließen, Mathematische Zeitschrift 39 (1935) 176–210 and 405–431. Engl. transl. in [26].

[2] A. S. Troelstra, H. Schwichtenberg, Basic Proof Theory, Cambridge Tracts in Theoretical Computer Science, 2nd ed., Cambridge University Press, Cambridge, 2000.

[3] N. Arai, T. Pitassi, A. Urquhart, The complexity of analytic tableaux, Journal of Symbolic Logic 71 (2006) 777–790.

[4] A. Urquhart, The complexity of propositional proofs, Bulletin of Symbolic Logic 1 (1995) 425–467.

[5] W. Bibel, E. Eder, Methods and calculi for deduction, in: D. M. Gabbay, C. J. Hogger, J. A. Robinson (Eds.), Handbook of Logic in Artificial Intelligence and Logic Programming, volume 1, Oxford University Press, Oxford, 1993, pp. 67–182.

[6] C. Wernhard, Generating compressed combinatory proof structures – an approach to automated first-order theorem proving, in: B. Konev, C. Schon, A. Steen (Eds.), PAAR 2022. CEUR Workshop Proceedings, volume 3201, 2022. https://arxiv.org/abs/2209.12592.

[7] C. Wernhard, CD tools – Condensed detachment and structure generating theorem proving (system description), CoRR abs/2207.08453, 2023. https://doi.org/10.48550/ARXIV.2207.08453.

[8] M. Rawson, C. Wernhard, Z. Zombori, W. Bibel, Lemmas: Generation, selection, application, in: R. Ramanayake, J. Urban (Eds.), The 32nd International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2023), Prague, Proceedings, Prague, 2023. Https://arxiv.org/abs/2303.05854.

[9] W. Bibel, J. Otten, Experiments with connection method provers, in: 4th Conference on Artificial Intelligence and Theorem Proving, AITP 2019, April 7-12, 2019, Obergurgl, Austria, 2019. http://aitp-conference.org/2019/slides/WB.pdf.

[10] W. Bibel, J. Otten, From Schütte's formal systems to modern automated deduction, in: R. Kahle, M. Rathjen (Eds.), The Legacy of Kurt Schütte, Springer, Cham, 2020, pp. 215–249.

[11] W. Bibel, Short proofs of the pigeonhole formulas based on the connection method, Journal of Automated Reasoning (1990).

[12] A. Haken, The intractability of resolution, Theor. Comput. Sci. 39 (1985) 297–308.

[13] M. J. H. Heule, B. Kiesl, A. Biere, Short proofs without new variables, in: L. de Moura (Ed.), CADE 26, LNCS (LNAI), Springer, 2017.

[14] J. Otten, W. Bibel, leanCoP: Lean connection-based theorem proving, Journal of Symbolic Computation 36 (2003) 139–161.

[15] J. Otten, nanoCoP: A non-clausal connection prover, in: N. Olivetti, A. Tiwari (Eds.),

International Joint Conference on Automated Reasoning, IJCAR 2016, volume 9706 of *LNAI*, Springer, Heidelberg, 2016, pp. 302–312.

[16] J. Otten, Clausal connection-based theorem proving in intuitionistic First-Order Logic, in: TABLEAUX 2005, International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, volume 3702 of *Lecture Notes in Computer Science*, Springer, Berlin, 2005, pp. 245–261.

[17] J. Otten, Implementing connection calculi for first-order modal logics, in: E. Ternovska, K. Korovin, S. Schulz (Eds.), 9th International Workshop on the Implementation of Logics (IWIL 2012), Merida, Venezuela, 2012.

[18] C. Wernhard, W. Bibel, Learning from łukasiewicz and Meredith: Investigations into proof structures, in: A. Platzer, G. Sutcliffe (Eds.), Proceedings of the 28th International Conference on Automated Deduction (CADE-28), volume 12699 of *LNAI*, Springer, 2021, pp. 58–75.

[19] C. Wernhard, W. Bibel, Investigations into proof structures, JAR (under review 2023).

[20] W. Bibel, Automated Theorem Proving, second ed., Vieweg Verlag, Braunschweig, 1987. First ed. 1982.

[21] W. Bibel, Deduction: Automated Logic, Academic Press, London, 1993.

[22] L. Bachmair, H. Ganzinger, Equational Reasoning in Saturation-Based Theorem Proving, volume Automated Deduction – A Basis for Applications. W. Bibel and P. H. Schmitt (eds.), Volume I of *Applied Logic Series*, Kluwer, Dordrecht, 1998, pp. 353–398.

[23] L. Bachmair, H. Ganzinger, C. Lynch, W. Snyder, Basic paramodulation, Information and Computation 121 (1995) 172–192.

[24] G. Prusak, C. Kaliszyk, Lazy paramodulation in practice, in: B. Konev, C. Schon, A. Steen (Eds.), PAAR 2022, volume 3201 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.

[25] W. Bibel, A vision for Automated Deduction rooted in the connection method, in: R. Schmidt, C. Nalon (Eds.), The 26th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2017), volume 10501 of *LNAI*, Springer, 2017, pp. 3–21. DOI: 10.1007/978-3-319-66902-1_1.

[26] M. E. Szabo (Ed.), The collected papers of Gerhard Gentzen, North-Holland, Amsterdam, 1969.