

Research Perspectives for Logic and Deduction

Wolfgang Bibel*
Darmstadt University of Technology

Abstract

The article is meant to be kind of the author's manifesto for the role of logic and deduction within Intellectics. Based on a brief analysis of this role the paper presents a number of proposals for future scientific research along the various dimensions in the space of logical explorations. These dimensions include the range of possible applications including modelling intelligent behavior, the grounding of logic in some semantic context, the choice of an appropriate logic from the great variety of alternatives, then the choice of an appropriate formal system for representing the chosen logic, and finally the issue of developing the most efficient search strategies. Among the proposals is a conjecture concerning the treatment of cuts in proof search.

Often a key advance is a matter of applying a small change to a single formula.

Ray Kurzweil [Kur05, p.5]

1 Introduction

Luigia Aiello has made numerous important scientific contributions in many areas of Artificial Intelligence. But it is fair to say that her core interest has always been in a logical approach to Artificial Intelligence (AI) throughout her career. For instance, as early as 1980 her paper [Aie80] appeared in the section on theorem proving at the very first AAAI Conference, noteworthy one out of merely two papers presented by European authors at this legendary conference in the US. Many more papers in a similar vein by herself and her numerous students preceded and followed this particular one.

It is for this reason that I chose to honor her at the occasion of her sixtieth birthday with a perspective contribution to this particular area. I would like to express through it my highest respect for her achievements and my deepest gratitude for the professional and personal friendship and the fruitful cooperation which has lasted for more than a quarter of a century.

*Also affiliated with the University of British Columbia. Author's e-address: Bibel@gmx.net

The elder AI generation still has vivid recollections of the hot debates of the seventies in the last century within the community concerning the role of logic and deduction in AI. Notwithstanding the GOFAI (“good oldfashioned AI”) debate triggered by Rodney Brooks, the central role of logic within many areas of AI and Computer Science (CS) today is undisputed – perhaps even too much so. Like in any scientific discipline it is from time to time worthwhile to review the direction of research from a high-level point of view, thereby abstracting from the day-to-day focus on specific research problems, and rather take the entire picture of the discipline into consideration. This is what the current article aims to do. In other words we want to discuss the various dimensions of logic and deduction and their role within AI. Neither the analysis nor the role description can in any way be comprehensive in such a short article; they rather reflect the bias of the author concerning his judgment of particularly important issues. The result of these considerations is a number of concrete proposals for future research which are deemed particularly promising. In short, the text may be regarded as the author’s manifesto for an area in which he has worked for nearly four decades.

For completeness the article contains a short summary of the goals of AI – or rather of Intellectics. As we all know those seventies also brought about a schism of our discipline in the way of a separation of the field into the more systems-oriented AI and the Cognitive Science (CogSci) focussing more on the study of natural intelligence and its basis. I am deeply convinced that these two directions have to go together in a synergetic way in order to achieve their mutual and intertwined goals. This deep conviction is the reason for my stubborn adherence to a common name, *Intellectics* [Bib92], proposed in 1980 for the discipline spanning both subareas; in short, Intellectics = AI & CogSci.

As we said the article analyses the nature and role of logic in achieving these general and longterm goals. It begins with a brief view at those goals of Intellectics, thereby pointing out two major subgoals, viz. solving the integration problem and contributing to the solutions of the complex problems with which our societies are currently confronted, whereby logic could play an important role. In addition to the standard applications of logic we then outline as a challenging research line its role in modelling intelligent behavior in a conjunctive way and in compiling from such a descriptive model applicational systems. This development would include a logical modelling language which is not suffering from the limitations experienced with languages like UML.

While logic currently is used exclusively without any kind of grounding its constants, such an association with semantic information could be rather beneficial in terms of efficiency, and is therefore proposed as another challenge. A further section deals with the choice of an appropriate logic in dependence of the intended application and the required features including change, vagueness and uncertainty and proposes research on some measure for a more rational distinction among a variety of logics.

Once we have settled in for a chosen logic for a certain application there is still a wide variety of formal systems to chose from for expressing the logic and support the inferential mechanisms. We once again remind of the important research strategy aiming at a formalism which is as compressed as possible. While remarkable results have been achieved in this line of research such as the *leanCoP* system, the approach as a whole is

not exhausted at all. Several important longstanding questions have still not been solved and incorporated into actual systems and further ones are raised for future work in this respect. Among these is the challenge of integrating cuts which lead to shorter proofs. We conjecture that a way to do so is by the use of factoring and engaging a nonclausal form calculus (or a subsequent linear transformation to clausal form).

Finally we discuss that part in proof search on which most efforts were spent in the last decades, which is the development of efficient search strategies. We argue that such strategies would need to be context dependent and that really efficient and specific ones might become too complex to develop by hand. For this reason we propose an automatic design of search strategies based on experimental data, a methodology successfully applied already to solving hard combinatorial problems. We also encourage the community to reconsider the integration of examples into the search for proofs and propose to do so in a preprocessing manner.

2 Main Intellectuals Goals

Intellectics aims at a profound understanding of the working of human intelligence in brains (CogSci part) and at mechanizing human-level intelligence (AI part). The fiftieth anniversary of the Dartmouth Conference in 1956 has given rise to numerous reflections on what the field has achieved in the first half century in pursuing these goals and what should be done now in order to progress further. The issue of the *AI magazine* (vol. 26,4) celebrating the 25th anniversary contains numerous statements of this kind. They suggest new challenge problems and research strategies.

But perhaps one should once again take one step further back and ask whether and why we should continue to pursue these grand goals. As far as the CogSci part is concerned the justification is straightforward. Curiosity is inherent in human's nature and we are simply curious what mechanisms make us intelligent. A deeper insight into these mechanisms could have numerous beneficial implications including cues how to educate humans more effectively, how to communicate more smoothly among ourselves, how to improve our problem solving capabilities, and so forth.

The justification of the AI part is not as obvious. Why should the more than six billions of humans on earth strive for a new breed of intelligent agents? Well, first of all Intellecticians are convinced that the CogSci part of the goals cannot be achieved without actually realizing human intelligence in an artificial way. In other words, AI is a prerequisite for CogSci in this sense so that AI inherits its justification already from CogSci. Reversely, AI also needs CogSci insights for inspirations and new ideas how to proceed. In fact one approach to AI consists in *reverse engineering* of the brain [Kur05, Ch.4] which can only be achieved on the basis of CogSci input (not least the one from the neurosciences). This mutual dependency and the common goals are good reasons for regarding AI and CogSci as a single discipline. Apart from the basic justification of AI just mentioned, the short history of AI research has produced plenty of evidence that it generates techniques which have become extremely useful in numerous applications, even so many that any short list of examples would leave us with too a distorting picture.

I myself have always regarded the ultimate AI goal as a rather distant one providing the Intellectics community with a socially uniting umbrella, but too distant to influence our daily work in concrete terms. In this vein I continue to think that our next subgoals should rather be guided by responsibilities to the society at large, although in such a way that they are deemed compatible with, and their achievements steps towards, AI's grand goal. In this sense I see two such subgoals as of paramount importance, one basic the other applicational. The first, basic subgoal consists in solving the urgent issue of integration, the second in attacking fundamental problems in our societies whose solution could be achieved by AI technologies. Both subgoals will be expanded further in the following.

Intellectics in general and AI in particular today is rather fragmented. The vision community within AI has little or no interaction with the knowledge representation community, to give one out of many possible examples. As a result we have vision systems such as those built into the autonomous vehicle Stanley which triumphally won the DARPA contest in 2005; but Stanley "knows" literally nothing at all about the world it sees. How could we integrate into such vision systems knowledge systems without reprogramming everything? How could we then extend the resulting system in the same vein by integrating speech and natural language understanding systems, planning systems and a variety of systems with further functions including those beyond AI in a way so that the final integrated system features a truly intelligent behavior? These questions refer to what we call the *integration problem*. We believe that logic holds the key for solving this problem as discussed in the subsequent section.

Humanity faces dramatically complex problems to be solved in a relatively short time, foremost the problems caused by the world climate change due to the man-made increase of greenhouse gas like carbon dioxide and methane in the atmosphere the consequences of which can be traced in many global phenomena like the glacier retreats, the warming of oceans leading to a dramatic reduction of life in it and to numerous other frightening consequences, the disappearance of virgin forrests, the spreading of deserts, to mention just a few [Lov06]. Despite the exponential growth of technological advances we have to make sure that enough time is left for humanity to be able to harvest the fruits of these advances. Namely, the intrusion of these advances into the social mechanisms seems to take place at a much slower pace, as some of those (like politics, law, social struggles etc.) have not changed much since ancient Greek and Roman times. Hence, it is currently undecided where this brinkmanship of meddling into the global mechanisms of nature will lead us. Therefore scientists have the responsibility to contribute to the solutions of these fundamental problems rather than pursuing prestigious goals for the goals sake.

AI technology can contribute substantially in this endeavor in many ways as has been described in great detail in my recent book [Bib03]. I see for instance a key role for knowledge and problem solving systems in a more rational approach towards solving societal and global problems such as the one just mentioned (viz. the world climate). Problems of this complexity cannot be coped with by the locally oriented problem solving attitude of humans but only by a truly global consideration of all aspects involved of the kind as realized through objectively accumulated knowledge bases and general problem solving mechanisms. In [Bib05a] this potential has been outlined for the complex area of law. Progress in such domains which are fundamental for the prosperity of societies

– and there are many more than just the legal domain – would have an even far greater impact also for the standing of AI as a discipline than, say, a program which beats the worldmaster in chess, notwithstanding the fact that this is indeed a truly impressive achievement. Another vital domain of application of this kind of AI technology is science itself as has recently been pointed out in several foresighting reports [Bib05b, Emm06] followed by the 23 March 2006 special issue of *Nature*. Again we believe that logic is substantial for these kinds of contributions.

The emphasis on these two selected subgoals is not meant to diminish all the fascinating work currently going on in all other subareas of Intellectics. Rather we want to point out that these particular ones deserve at least the same level of attention. We do sense an imbalance in this respect which to some extent may be due to the schism between AI and CogSci.

3 Why Logic?

Logic is often paraphrased as the language of thought. Because thinking is a crucial component of intelligence, logic on this account will most likely play an important role in an artificially intelligent agent at some level of abstraction. Aspects of this role can already be observed in knowledge systems, in automated theorem proving (ATP), logic programming, problem solving, and so forth.

Given these successful applications of logic we feel that no further justification for the relevance of logic is needed. Nevertheless there is a fundamental criticism of a logical approach to achieve artificial intelligence. According to this argument our brains function in a rather different way. For instance, catching a ball does not involve solving differential equations but a direct transformation of the observed movements of the ball into an appropriate movement of the player's arms and legs. Similarly, it is supposed that reasoning as well is realized in the brain by analogue direct transformations rather than by logical deductions. While this may well be the case it is still important to understand the underlying mechanism in terms of the higher level of abstraction of logic, as it is important to understand a ball's movements in terms of differential equations. How we eventually will realize such behavior in artificial systems is quite an independent question.

Besides this role of logic as the language of thought there might be another similarly prominent role for logic in Intellectics. As we said in the preceding section understanding intelligent, cognitive behavior and making machines exhibit such behavior is the goal of Intellectics. In order to characterize this additional role of logic we begin with mentioning that there are at least three different viewpoints from which this goal can be approached. The first is the viewpoint of observing and analyzing intelligent behavior in existing creatures, foremost in humans. The second takes the perspective of future artificially intelligent systems or robots and their potential architectural design without much ado about how to realize it with present technology. And the third focusses on concrete steps towards realizing selected intelligent functions with present technology which eventually might be part of a future intelligent agent.

Each of the viewpoints has its merits and each is necessary for a future overall success. Depending on which of these viewpoints one takes rather differing standpoints and preferences may be chosen. In the past these differences were, as already indicated in the Introduction, the cause for schism and hot debates in the community. For instance, CogSci – rooted deeply in the first and analytic viewpoint – separated from AI in the late seventies (see [Fei05, pp.33f] for some background information) mainly because the AI community to a large extent became obsessed with quick commercial successes based more or less exclusively on the third viewpoint which is synthetic and bottom-up. How about the second viewpoint?

It seems unlikely to me that the current bottom-up and patchwork-like approach in AI will ever lead to a truly intelligent agent. Ultimately this goal will not be reached but in a top-down fashion starting from the insights gained by CogSci, Neuroscience and by introspection (cf. [Kur05, p.168] for a similar argument). For that purpose these insights need to be accumulated in a computational model reflecting the many facets of human intelligence. The generation of such a model is a Herculean task given the complexity of intelligent behavior.

The only way I could think of mastering it would be a *conjunctive* one in the logical sense. That is, if we have two independently generated parts M_1, M_2 of the model then these can be combined by a simple conjunctive (or additive) operation like logical conjunction $M_1 \wedge M_2$. The reason for this requirement lies in our human way of insight. We are bound to understand just small fragments of the entire workings of intelligence at any given time. So in addition to forming each single fragment in some representation there must be some operation which makes a coherent mosaic out of the myriads of represented fragments. The operation must be simple in order to cope with the sheer amount of pieces.

In other words I am pleading here for the accumulation of a coherent, formalized and implemented model of intelligence in the sense of the second viewpoint. This task would involve many scientists, even generations of scientists over a long period of time. Could and should we afford such a grand endeavor? Yes, we could because the project would necessarily employ an economic “anytime” procedure of the following kind.

Such a model would have to be in some way *descriptive* in order to comply with the requirement of conjunctiveness, whereby descriptive (or declarative) is meant in a rather broad sense possibly even including natural language descriptions, pictures and scenarios, the simulation of dynamic processes, etc. At any given time the model accumulated upto a given point could be synthesized to a working agent featuring all the aspects accumulated in the model. This means that the model could at the same time serve a variety of many practical purposes in the sense of the third viewpoint. Namely, for each particular application one selects the desired parts in the model existing upto that point in time and synthesizes from there the applicational system. In other words the second and the third viewpoint could from there on be pursued in a synergetic manner.

This vision assumes two major prerequisites in order to be realizable. The first is the existence of a conjunctive formalism which is descriptive in the sense just indicated. Since logic is both descriptive and conjunctive it is exactly here where we see a central role for logic, possibly an extended logic with many more features than currently familiar. The

second prerequisite is a mechanism which synthesizes working systems out of such a formal model. This is less illusionary than one might think at first sight. Just think of current practice in systems engineering which often generates a model in some language such as UML (ie. universal modelling language) and extracts from it the systems code, to some degree in an automated way.

The analogy with UML demonstrates that we are proposing here a rather realistic and fruitful research project with two major research lines. One is the development of a language like UML but without UML's severe limitations which at present are painfully felt in many applications. Most likely such a language would be more logic-like. The other line consists of a further automation of the synthesis of systems code out of a formalized model. Both goals are of utmost importance for current software practice. It is these two goals which drive the Mercury project [BGM⁺06] (see also below Section 6) for exactly the reasons I have given here. So the grand endeavor of accumulating a model of intelligence could in fact be pursued in parallel with and on the basis of research on very practical tasks. In addition, the methodology promoted here for evolving a model of intelligence would of course be useful for any area which strives for understanding complex structures, eg. those present in social systems.

As we just touched upon current software practice, it is interesting to take a retrospective view and note how little progress the software community has achieved in the last thirty years in terms of a truly user-oriented software approach of the kind which the present author described more than thirty years ago (see eg. [Bib80] and the references given therein). In essence that approach (originally termed *predicative programming*) shares the methodology with what is described here for developing a model of intelligence. So in a nutshell I have here just reiterated for the development of intelligent agents what I proposed three decades ago as a better way of producing software. In this context it is encouraging that the recent years have indeed seen remarkable steps into this direction also within the software community. Namely, computation independent models (CIM), model-driven architectures, post requirement specification traceability, and several further terms of this kind are now the catch-words of the day circumscribing an approach to software generation of the kind envisioned with predicative programming.

4 On the Issue of Grounding

Logic employs a formal language without any grounding. A constant symbol such as *a* or *table* has no semantics at all. A logical statement like *On(glass, table)* may be subject to many different interpretations, not the one intended by the choice of the names of the constants. Human knowledge seems to operate in a completely different way (and the specific discipline of semiotics studies the meanings of human symbols) – or does it? Well, we do not really understand the working of the brain wrt. its knowledge processing, so at the time being we cannot be sure.

Nevertheless anyone can experience by introspection that many seemingly logical conclusions in everyday life are drawn by inspection of a mental model rather than by deductive inference [JLB91]. For instance if someone tells me that a book lies on the table with

a glass standing on the book, I “see” that in this imagined scenario the glass is above the table without regress to formal rules concerning the transitivity of the “on”-relation and its connection with the “above”-relation.

If, on the basis of such experience, we take for granted that the brain realizes logical conclusions partially by way of inspecting mental models the question naturally arises whether an analogue mechanism might not be similarly useful in AI systems. It is therefore our proposal to investigate this possibility in future research. Here are a few ideas how this might be achieved.

In the brain the word, say, *table* is associated with the sensory information deriving from a number of concrete tables experienced out there in the world. Already the issue of this association raises a number of questions. Does the mental model of a table refer to the sensory data of a particular, selected table? Or does the brain use some mechanism of abstraction to generate some mental model of table with the common characteristics of all experienced tables? Anyway, an artificial agent could similarly associate with a name such as *table*, used in its logic, corresponding sensory data and classify it as a table, ie. as a particular unary predicate. Similarly with any other constant, function or predicate. There might be many ways to exploit such an association in the reasoning processes carried out within a logical system. Such an association would be a first step towards involving a true semantics in machines and would have great relevance for many applications where knowledge plays a key role. We will come back to one of these applications in Section 7 where we discuss examples supporting the inferential process which in AI was the first and so far only attempt to integrate this kind of information.

5 Which logic?

The history of logic has produced a great variety of logics. One of the reasons for this variety is the choice one has between axiomatization and logical structure. In other words, some more advanced features of natural expression can be characterized in an axiomatic way or, alternatively, be built into the logic itself.

An example is how actions and change are formalized. Logic has traditionally been used to model static logical reasoning while changes in the modelled world have been considered as quite different a matter. Modelling applications like planning and computation however have forced us to integrate change into logical frameworks. How this best is done is still a matter of debate. The situation calculus was one of the very first attempts of this kind which uses first-order logic to characterize the features needed to model actions and change in an axiomatic way. In a series of papers eventually resulting in [Bib04] the present author has developed a transition logic as an alternative. The idea is to regard changes or transitions as first class citizens (in the form of transitional rules) within the formal framework and otherwise keep the logical part more or less like in first-order logic. A related approach focussing on an integration of concurrency is described in [Kah05]. Modal logic, linear logic as well as the more recent computability logic [Jap05] in contrast build changes into the logical connectives inside the logic.

There is an even larger area open for research in this topic of modelling changes. Namely, the bread and butter of actual modelling and simulation systems (eg. for modelling the climate, ocean currents, physiology, etc.) are differential equations. But in addition we would urgently need knowledge of a logical kind to be integrated into such systems. However the author is not aware of any practical formalism which could provide the theoretical basis for such an integration.

Another example of modelling features in logic is vagueness and uncertainty where the same kinds of alternatives have been developed. Namely, there is the basically first-order treatment extended by probabilistic features as in [Poo03] and, alternatively, there are logics which express vagueness and uncertainty by means of the logical operators like fuzzy logic, nonmonotonic logic and so forth.

Upto this day there is no systematic study on which a rational choice of an appropriate logic in dependence of the intended application and the required features could be based. Research rather proceeds in trying out many possibilities at the same time and rather independently. This unsystematic strategy is one reason for a waste of resources in the community which should be taken notice of.

This is not to say that we are in lack of any comparative arguments concerning the various logics. An important one is the complexity of the proof search. For instance in description logics we have a detailed classification in this respect which of course is extremely helpful. However, there seems to be no way around involving logics which – assuming $P \neq NP$ – are computationally non-polynomial. Among them there is still a great variety of logics waiting for a distinction through some measure. Apart from the ones mentioned to handle changes and uncertainty there are many more where such a distinction would be rather helpful.

So here we have come across another proposal for future research in our domain namely to work out comparative arguments or measures distinguishing different logics in view of intended applications. While I refrain from stating any preferences in this respect which would be based merely on personal prejudice, my experience still tells me that classical first-order logic might in some form or another continue to play a major role in future formal logical reasoning as would higher-order logic whose potential seems still underestimated.

6 Formal Systems

There are numerous formal systems (or calculi) which encode in one way or another a given logic. Let us take first-order logic as our paradigm example because of its wide dissemination although any other logic could as well have been selected to make the point.

Chapter 4 in [Bib93, pp. 97ff] demonstrates that such formal systems may differ in the degree of their compression. Thereby we consider a formal system S_1 to be more compressed than another S_2 if any proof in S_1 is shorter (in terms of the number of symbols required) than the corresponding one in S_2 . For example, Gentzen's formal system of natural deduction NK is less compressed in this sense than the tableau calculus.

Occasionally compression may result in a substantial change in the complexity. For instance, the elimination of the cut rule leads to a less compressed calculus and to possibly exponentially longer proofs. In most cases however, compression has less dramatic effects. For instance the connection calculus is more compressed than the tableau calculus although the proof lengths differ only by a polynomial factor. This however does not mean that such a compression is worthless. On the contrary, our experience shows that the performance may increase dramatically as we demonstrated through a comparison of leanTAP with leanCoP in [OB03]. In fact, the intuitionistic version of leanCoP, called ileanCoP, is now by a wide margin the fastest theorem prover in existence for intuitionistic first-order logic [Ott05].

```

(1) prove(Mat,PathLim) :-
(2)   append(MatA,[FV:ClA|MatB],Mat), \+ member(-(_):_,ClA),
(3)   append(MatA,MatB,Mat1),
(4)   prove(!:[[]],[FV:[-(!):(-[])|ClA]|Mat1),[],PathLim,[PreSet,FreeV]),
(5)   check_addco(FreeV), prefix_unify(PreSet).
(6) prove(Mat,PathLim) :-
(7)   \+ ground(Mat), PathLim1 is PathLim+1, prove(Mat,PathLim1).

(8) prove([],_,-,_,[[]],[[]]).
(9) prove([Lit:Pre|ClA],Mat,Path,PathLim,[PreSet,FreeV]) :-
(10)  (-NegLit=Lit;-Lit=NegLit) ->
(11)  ( member(NegL:PreN,Path), unify_with_occurs_check(NegL,NegLit),
(12)  \+ \+ prefix_unify([Pre=PreN]), PreSet1=[], FreeV3=[]
(13)  ;
(14)  append(MatA,[ClA1|MatB],Mat), copy_term(Cla1,FV:ClA2),
(15)  append(ClaA,[NegL:PreN|ClA],ClA2),
(16)  unify_with_occurs_check(NegL,NegLit),
(17)  \+ \+ prefix_unify([Pre=PreN]),
(18)  append(ClaA,ClaB,Cla3),
(19)  ( ClA1==FV:ClA2 ->
(20)  append(MatB,MatA,Mat1)
(21)  ;
(22)  length(Path,K), K<PathLim,
(23)  append(MatB,[ClA1|MatA],Mat1)
(24)  ),
(25)  prove(Cla3,Mat1,[Lit:Pre|Path],PathLim,[PreSet1,FreeV1]),
(26)  append(FreeV1,FV,FreeV3)
(27)  ),
(28)  prove(Cla,Mat,Path,PathLim,[PreSet2,FreeV2]),
(29)  append([Pre=PreN|PreSet1],PreSet2,PreSet),
(30)  append(FreeV2,FreeV3,FreeV).

```

Figure 1: Main part of the ileanCoP source code

In the preceding discussion we measured compression in terms of lengths of proofs. It is interesting to have a look also at the length of the program underlying the theorem prover as a measure. leanCoP needs 333 bytes (!) for representing the program. The smallest version of ileanCoP derives from leanCoP by adding 191 bytes so that altogether it needs exactly 524 bytes without the approximately 30 lines required for prefix unification. Figure 1 shows the three clauses of the long version of the source code in a way such that both, leanCoP (everything except the underlined symbols) and ileanCoP can be seen. Note thereby that not a single symbol had to be changed in leanCoP but only additional information had to be included in the form of new terms and literals. The way how this extension can be achieved comes close to the kind of predicative programming the author

had in mind 30 years ago (and already mentioned further above). In contrast provers like Otter need hundreds of thousand times longer code. Modifying such a monster is simply impossible for any person (except for the system's author within a period of a few years after completion).

The way compression is achieved by the connection method (CM) for various logics has been described many times (eg. see [Bib93] and the references given therein) so that we can – and for reasons of space must – restrict ourselves to stating the most important of its essential features. The CM analyzes the structure of a given formula F without changing F whatsoever which has a particularly beneficial effect on the length of proofs. It focusses on establishing a spanning set of connections which characterizes the formula's validity. Thereby the procedure is connection- and goal-oriented, and unification is employed preferably relying on a particular partial-ordering on the set of terms rather than on Skolemization [Bib87, Sect. IV.8]. In the case of non-classical logics the unificational part is extended, eg. with prefix unification in the case of intuitionistic and modal logics. Complementary to the main top-down procedure bottom-up preprocessing steps may reduce the proof problem substantially. A CM proof in some connection calculus represents many different proofs in say a Gentzen-type formal system, ie. it identifies them by disregarding and abstracting from irrelevant differences. It is obvious that this dispensing with irrelevant burden has a beneficial effect on the efficiency of the resulting systems. Sofar the CM's features.

The research program underlying the CM approach is not exhausted at all. For instance, the transformation to normal form, even if done wisely, still introduces a lot of redundancy which distracts the proof search. A `leanCoP` for nonclausal-form formulas along the lines of [Bib87, p.150ff] is therefore highly desired but requires a mind with the unique talents like that of Jens Otten nurtured by an appropriate research climate. Further compressions like those described in Chapter 4 in [Bib93, pp. 97ff] (such as equality handling etc.) need then to be integrated into such a system. And last not least a further boost for the performance would come from a compilation of the various `leanCoPs` into a low-level programming language preferably by some automatic mechanism. The way how this can be achieved is shown by the remarkable Mercury project whose goal is to combine the virtues of declarative programming with features from current software practice, especially providing for (separate) compilation of declarative code among many other attractive features [BGM⁺06].

6.1 A conjecture concerning the cut

We have just pointed out the importance of compression and its role in the CM. In recent years there have been complementary attempts towards the compression of logical calculi. A prominent one is Guglielmi's calculus of structures [BG04]. It overcomes the restriction in Gentzen's calculi that inference rules can only be applied to surface (or main) connectives. Rather it allows inference rules to be applicable at any time to any logical connective anywhere inside the formula, a technique termed *deep inference*. Due to this enhanced flexibility the cut formula can be restricted to atoms with predicate symbols exclusively from the conclusion of the cut inference. This leads to a finitely generating

system even if the cut rule is included. Recall that the inclusion of the cut rule leads to a potentially exponential compression as already mentioned at the beginning of this section to see the relevance of this achievement. In Section 4.2 of [Ede00] a similar result is presented using meta-variables on arbitrary cut-formulas which insofar is to be considered as a less compressed approach. No clue is given in either approach when to apply the cut rule during the search for a proof as it could be applied at any point. A strategy in this regard is then a major challenge for future research in this area.

A first step into such a direction has been made in [LMG94] with the so-called folding up technique. It derives lemmata in a bottom-up manner during the main top-down procedure which is derived from an amalgamation of the connection and the tableau calculus underlying the high-performance proof system Setheo [LSBB92]. The paper shows that folding up can be viewed as a controlled integration of the cut rule.

Folding up builds into the connection tableau calculus a technique whose effect can alternatively be achieved with the form of factoring reduction which has been termed FACTOR in [Bib93, pp. 56]. In order to illustrate this relationship let us consider the matrix $\{\{p, t\}, \{\neg p, q, s\}, \{\neg t, p, s\}, \{\neg q, r\}, \{\neg s, r\}, \{\neg r, \neg p\}\}$ which is the prime example for illustrating folding up in [LMG94]. FACTOR applied to this matrix twice leads to the non-clausal form (NCF) matrix $\{\{p, \{\{t\}, \{\neg t, s\}\}\}, \{\neg p, q, s\}, \{\{\{\neg q\}, \{\neg s\}\}, r\}, \{\neg r, \neg p\}\}$ by factoring the p in the first and third clause and the r in the fourth and fifth clause of the matrix.¹ A connection calculus like the extension procedure for NCF formulas or matrices in [Bib87, p.150ff] would then behave exactly like folding up and establish the partial proof of the matrix as in [LMG94, Figure 8] with 5 extension and 1 reduction steps, ie. altogether 6 connections. In other words the effect of folding up and hence of the corresponding use of cut could equally be achieved with this kind of factoring.

Note that this matrix is in fact not complementary, but could be made so eg. by adding the clause $\neg s$ to it. Also note that FACTOR could be applied yet another time to the resulting NCF matrix by factoring additionally $\neg p$ which yields the matrix $\{\{p, \{\{t\}, \{\neg t, s\}\}\}, \{\neg p, \{\{q, s\}, \{\neg r\}\}\}, \{\{\{\neg q\}, \{\neg s\}\}, r\}\}$. Now the extension procedure would require only 5 extension steps (or connections) which illustrates that even in the special case of factoring just literals this kind of factoring is more powerful than folding-up. As a final remark we mention that FACTOR could alternatively have been applied to the original matrix by factoring the s in the second and third clause and the r in the fourth and fifth clause of the matrix resulting in $\{\{p, t\}, \{\{\{\neg p, q\}, \{\neg t, p\}\}, s\}, \{\{\{\neg q\}, \{\neg s\}\}, r\}, \{\neg r, \neg p\}\}$. A different (partial) extension proof would then be found.

This result established by Letz concerning the relationship of folding up and the cut along with the rather obvious fact just illustrated that in general folding up can always be

¹For readers unfamiliar with this kind of matrix notation for formulas we mention that (in the positive interpretation) such a (clause-form) matrix, ie. a set of clauses, can be read as a disjunction of its clauses (ie. sets of literals) which may in turn be interpreted as conjunctions of literals. In an NCF matrix the elements of the clauses may in turn be matrices, ie. disjunctions of clauses, rather than just literals, and so forth until any nesting depth. In the negative interpretation (commonly used eg. in the resolution literature) the role of disjunction and conjunction are interchanged, ie. a matrix is interpreted as a conjunction (rather than disjunction) of clauses, and so forth. The formal details may be found in a standard textbook like [Bib93].

viewed as applying FACTOR a finite number of times and then use a connection calculus for the resulting NCF matrix suggests an even more general conjecture. Namely, we conjecture that the general cut can linearly be simulated by FACTOR applied to arbitrary submatrices (ie. not only atomic ones) a finite number of times and then a connection calculus applied to the resulting NCF matrix. I pose this conjecture as a research challenge.

There is more evidence than the one just mentioned supporting the conjecture. It is known that the pigeonhole formulas are hard for resolution requiring exponential proof lengths [Hak85]. In contrast they can be established with polynomial proofs both in a Gentzen system with cut (or a Frege system) and in a connection calculus using among others FACTOR as a preprocessing rule [Bib90].² The general reason for this advantage could be the following.

The cut enables a compression of a proof in a Gentzen system.³ Resolution, in contrast, although superficially of the form of a cut, does not feature the full power of the cut rule since otherwise there would be polynomial resolution proofs for the pigeonhole formulas. It is unknown how much of the power of cut is inherent in resolution. It is conjectured that there is some of it and that this part makes resolution occasionally more efficient than CM-type proof systems for clausal logic (without FACTOR).

Since there is a close relationship between the formula to be proved and its Gentzen-type proof, an elimination of the redundancy in the formula by compression also decreases the potential for further compression of the proof by the cut rule. FACTOR enables the elimination of redundancy without loss of information, ie. the formula's "entropy" (in analogy with Shannon's information theoretical concept) increases by its application. When it reaches its maximal value, there is no room anymore left for compression of the corresponding proof by way of the cut rule.⁴ That basically is another way of putting the conjecture above. Once this potential is exhausted by future CM-systems they should

²Other reduction rules used are PURE, UNIT and Prawitz' matrix reduction. Renaming, a rather strong rule, is not required for proving the formulas, but is just used in the paper (on the metalevel) to be able to apply the induction hypothesis.

³As pointed out by Alessio Guglielmi (private communication) the cut may also play the role of enabling case analysis. For example, the cut $A \rightarrow B, \neg A \rightarrow B \vdash B$ features a way of trying to prove B in the two mutually exclusive cases in which the hypotheses A and $\neg A$ are assumed. This aspect of case analysis has been studied extensively in deduction either explicitly (like for instance in Plaisted's work) or implicitly (like in the connection calculus).

⁴In letters dated 4 August 1980 to both, Georg Kreisel and Dag Prawitz, the author already pointed out the conjecture that the shortening effect of cuts is mainly due to the redundancy (or "bureaucracy") in derivations and would disappear when the author's derivational skeletons [Bib87, p.190] were used instead. This conjecture raised the curiosity and interest of Kreisel which he expressed in his response letter.

In a nutshell this early conjecture can be phrased as follows. Let P_1, P_2 denote the premises of a cut with conclusion C . Let s_1, s_2 denote the skeletons of cutfree derivations of P_1, P_2 , resp., and s the skeleton of the derivation of C obtained from the derivations of P_1, P_2 by the well-known process of cut-elimination. Then the length of s is a polynomial function of that of s_1, s_2 , ie. not an exponential function as in the case of Gentzen-type derivations (which carry all that redundancy).

The conjecture stated in the present paper is even stronger than that earlier one and, in the present terms, states that this function is even a linear one for some C' obtained from C by applications of the (first-order version of the) FACTOR operation to C , and even if Gentzen-type derivations would be taken instead of skeletons.

uniformly outperform standard resolution systems since then the only remaining advantage of resolution will also be incorporated into them, and otherwise their connection and goal-oriented behavior endows them with a clear advantage over standard resolution systems.

It is generally believed that cut formulas have to be invented creatively without much clue given by the conclusion formula which seems to speak against our conjecture. However, Guglielmi's result just mentioned might be seen as an indication that all propositional information about the cut formula might indeed be contained in the conclusion while its first-order features, ie. the terms, could anyway be determined by unificational mechanisms in the usual way. Some "creativity" would of course still be needed as is already illustrated by our example above where different sequences of applying FACTOR have led to different proofs, so that the mechanism would have to explore the finite space of different such sequences for the most suitable one (in the sense of the maximal entropy value). A parallel approach to this exploration might eventually be taken into account (similarly as in reality where more than one mathematicians are trying to solve mathematical problems).

If this conjecture could be proved correct then an algorithmic realization of this idea would still be complicated, especially when first-order unificational mechanisms have to be integrated (into FACTOR etc.), thus posing a further research challenge in this case. Even if the conjecture would turn out not to be generally valid FACTOR would still remain an attractive reduction operation which has been neglected in current systems. Note thereby that, even if a connection calculus for NCF would not be available or undesirable, one could still apply FACTOR, then apply to the result a linear or quadratic transformation to clause form [Bib93], and finally apply any proof method to the result, which occasionally would be a more compressed formula than the original one.

Guglielmi's calculus has motivated the question for the *essence* of proofs after eliminating all bureaucracy caused by individual formal systems. The answer given in [LS05] naturally is closely related to that given by the CM, namely that this essence is basically given by the connection structure underlying a proof. The paper in addition clarifies the effect of the cut rule in terms of a composition operation on such structures (without addressing the questions underlying our conjecture). This result might be helpful as well in the context of incorporating the cut into proof search one way or another. But it does not yet attack the first-order features as has been done within the CM with its skeleton concept pointed out in footnote 3.

Another step towards compressing Gentzen's sequent calculus has recently been taken in [Jap05] which introduces the so-called cirquent calculus. "Roughly speaking, the difference between the two is that, while in Gentzen-style proof trees sibling (or cousin, etc.) sequents are disjoint and independent sequences of formulas, in cirquent calculus they are permitted to share elements." In other words, a proof is no more a tree of sequents but becomes a compressed tree-like structure whereby different branches share joint parts. Such a calculus can be sensible to resources and in fact it has evolved in the context of attempts to develop a computational logic of the linear-logic kind. Whether or not computation and planning will be modelled in such a purely logical way in the future or rather

in the transitional way described in the previous section is independent of the interesting compressional idea behind the cirquent calculus.

7 Search strategies

Whatever formal system we might have chosen proofs in it cannot be found without search. Any such proof search has two quite different parts. One consists of a mechanism which is not really search in the sense of the word. Rather it combines a number of operations which are needed to test one single alternative for success or failure. In contrast to this the other part consists of choosing true alternatives where the wrong choice might well lead into a blind alley or at least into a superfluous detour (in a confluent system). For instance if we think of a tableau system in propositional logic for simplicity, it is the alternative branching points which give rise to search in the true sense while the remaining steps are rather straightforward. Let us refer to the sequential and the choice part to distinguish the two in the following.

A lot of efforts in a variety of directions have been invested into dealing with the choice part. One direction has been to deal with the alternatives in a parallel way. Since always only a limited number of different processors are available parallelization has the potential to provide some improvement but not a cure to the underlying exponential explosion. Since the single processor machines became so much more efficient they outraced the advantage of multiprocessor machines in this application and will do so for some time to come. An exception might be pursuing a finite (and small) number of alternatives in parallel like those in the application of FACTOR described in the previous section.

A second direction of research tried to take advantage of the information gathered in one alternative to be used also in another one in order at least not to waste redundant efforts completely. Intelligent backtracking was a popular technique in this direction. To some extent the same effect can be achieved with compression as discussed above.

A third direction tried to enhance the chances for selecting the right alternative. Many attempts have been made in this regard, including some rather naïve ones as seen from hindsight. Just think of the many so-called refinements of resolution some of which were mere adhoc attempts based on evidence of a few selected examples. Others were indeed based on solid theoretical arguments. For instance, Setheo [LSBB92] featured a preference in its selection strategy based on basic probabilistic arguments. However there could be many more preferences of this sort but they are difficult to develop and integrate into the overall strategy. It seems therefore that the ideal preference measure might be too complex to develop by hand.

Faced with all these difficulties some people nurtured the hope that human ingenuity might interact with systems in cases of difficult choices. I continue to regard this as a vage hope. Human ingenuity fails to blossom in the complex technical contexts and at states of our systems when these would need advice most. Human advice should therefore be integrated into the way the problem to be solved is stated upfront rather than investing

in interactive systems and proof planning approaches which require an interaction on a rather deep technical level.

For all these experiences the most promising perspectives are in an automatic design of the search strategy based on experimental data. The technique for this direction has been formally developed in [Bir05] in the context of metaheuristics. What is needed then is the integration of this technique into the framework of theorem proving. Basically this amounts to learning the search strategy from the data of successful proof search. It is to be expected that the resulting strategy depends upon the theory within which the proofs are to be searched for. That is the strategy in a purely logical framework without special axioms will presumably be different from one in, say, linear algebra.

I also believe that this way analogical reasoning will eventually be made possible in a practical sense. That is if the learned strategy used data from successful proofs, it will succeed in finding analogical proofs by way of the learned strategy; in other words, the analogy is coded into the strategy rather than in some logical form. This fits well with the observation that strategies used by humans typically are fuzzy. For instance, in chess many such fuzzy rules can be learned from textbooks. Similarly, in law such rules or strategic principles are common place known under the term *topoi*. Although sometimes seemingly contradictory, they are extremely helpful in human problem solving in chess, law, mathematics and many other disciplines. Capturing them in precise rules seems nearly impossible while computationally learning such strategic principles appears a promising perspective.

In human theorem proving examples play a prominent role. For instance, in [Rob00] the attention is focussing on mathematical proofs including psychological phenomena like gazing at some structures or immediacy in recognizing truth. It is therefore surprising that in current systems such kind of feature is hardly ever present. It is well-known how examples can guide the proof search and avoid blind alleys [Bib93, pp. 143ff] but the technique has not found its way into applications. The reference just given mentions as a possible reason for this fact the problem of how to generate examples or counterexamples automatically for a given theory (and gives references to respective approaches). Possibly it has been overlooked that the technique could be used in a preprocessing manner rather than by interrupting the proof search at certain choice points and query the available examples for guidance. While such an interrupt does not fit into the fast processing of modern proof systems, a preprocessing of this kind could indeed easily and elegantly be integrated and this way approximate the human way of mathematical proofs more closely.

Namely, as explained in the given reference examples require the open subgoals during a proof search to be satisfiable for the interpretations given by the examples. The respective information could be collected prior to the proof search for a number of examples and for each potential literal in the clauses of the theory and stored along with the literals by way of an appropriate data structure. During the actual proof search this information could then easily be checked and the choice made appropriately. I consider the ignorance of this possibility to be a major oversight on the side of the ATP community and its realization a project of high priority.

8 Conclusions

This paper has explored the most important dimensions of the space of logical research. In each of these dimensions we have pointed out opportunities for future research which are deemed of great relevance for the success of the logical approach towards the grand goals of Intellectics. A list of these research proposals has been given at the end of the Introduction.

The author shares the confidence with Luigia Aiello that the succeeding generation of researchers will be picking up these challenges and pursue their solutions with the same enthusiasm as we did during the hey-days of our careers to the benefit of humankind. I combine this confidence with the hope that society will appreciate this work more than it did so far and provide the logic talents with a research environment appropriate for their work (eg. with a research institute of the kind of a Max-Planck-Institute) which, like in Mathematics, requires an extreme amount of concentration, certainly more so than in “softer” disciplines.

Acknowledgments. I greatly appreciate discussions on the topics of the paper with Kai Brännler, Uwe Egly, Alessio Guglielmi, Reinhold Letz and last not least Jens Otten who also provided the leanCoP figure. I also thank two anonymous referees for their comments. All these interactions have led to substantial improvements of the text. For any remaining errors I am of course fully responsible.

References

- [Aie80] Luigia Aiello. Automatic generation of semantic attachments in FOL. In Robert Balzer, editor, *Proceedings of AAAI-80*, Menlo Park CA, 1980. AAAI.
- [BG04] Kai Brännler and Alessio Guglielmi. A first order system with finite choice of premises. In Vincent Hendricks, Fabian Neuhaus, Stig Andur Pedersen, Uwe Scheffler, and Heinrich Wansing, editors, *First-Order Logic Revisited*, Logische Philosophie, pages 59–74, Berlin, 2004. Logos Verlag.
- [BGM⁺06] Ralph Becket, Maria Garcia de la Banda, Kim Marriott, Zoltan Somogyi, Peter J. Stuckey, and Mark Wallace. Adding constraint solving to mercury. In *Proceedings of the Eighth International Symposium on Practical Aspects of Declarative languages*, Charleston, South Carolina, January 2006. Springer Verlag.
- [Bib80] Wolfgang Bibel. Syntax-directed, semantics-supported program synthesis. *Artificial Intelligence*, 14:243–261, 1980.
- [Bib87] Wolfgang Bibel. *Automated Theorem Proving*. Vieweg Verlag, Braunschweig, second edition, 1987.

- [Bib90] W. Bibel. Short proofs of the pigeonhole formulas based on the connection method. *Journal of Automated Reasoning*, 1990.
- [Bib92] Wolfgang Bibel. Intellectics. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 705–706. John Wiley, New York, 1992.
- [Bib93] Wolfgang Bibel. *Deduction: Automated Logic*. Academic Press, London, 1993.
- [Bib03] Wolfgang Bibel. *Lehren vom Leben – Essays über Mensch und Gesellschaft*. Sozialwissenschaft. Deutscher Universitäts-Verlag, Wiesbaden, 2003.
- [Bib04] Wolfgang Bibel. Transition logic revisited. 2004. Submitted.
- [Bib05a] L. Wolfgang Bibel. AI and the conquest of complexity in law. *Artificial Intelligence and Law Journal*, 12:159–180, 2005.
- [Bib05b] Wolfgang Bibel. Information technology. Technical report, European Commission, 2005. ftp://ftp.cordis.lu/pub/foresight/docs/kte_informationtech.pdf.
- [Bir05] Mauro Birattari. *The Problem of Tuning Metaheuristics as seen from a machine learning perspective*, volume 292 of *DISKI*. Akademische Verlagsgesellschaft Aka, Berlin, 2005.
- [Ede00] Elmar Eder. The cut rule in theorem proving. In Steffen Hölldobler, editor, *Intellectics and Computational Logic*, volume 19 of *Applied Logic Series*, pages 101–123. Kluwer Academic Publishers, Dordrecht, 2000.
- [Emm06] Stephen Emmott. Towards 2020 science. Technical report, Microsoft Research Cambridge, 2006. <http://research.microsoft.com/towards2020science/>.
- [Fei05] Edward A. Feigenbaum. Stories of AAI – Before the beginning and after – A love letter. *AI Magazine*, 26(4):30–35, 2005.
- [Hak85] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.
- [Jap05] Giorgi Japaridze. Introduction to cirquent calculus and abstract resource semantics. *Journal of Logic and Computation*, 2005. To appear, see <http://arxiv.org/abs/math.LO/0506553>.
- [JLB91] Philip Nicholas Johnson-Laird and Ruth M. J. Byrne. *Deduction*. Lawrence Erlbaum Associates, Hove and London (UK), 1991.
- [Kah05] Ozan Kahramanoğulları. Towards planning as concurrency. In M.H. Hamza, editor, *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, AIA 2005, February 14-16*, pages 387–394, Innsbruck, Austria, 2005. Acta Press.
- [Kur05] Ray Kurzweil. *The Singularity Is Near – When Humans Transcend Biology*. Viking, Penguin Group, New York NY, 2005.

- [LMG94] Reinhold Letz, K. Mayr, and C. Goller. Controlled integration of the cut rule into connection tableaux calculi. *Journal of Automated Reasoning*, 13:297–337, 1994.
- [Lov06] James Lovelock. *The Revenge of Gaia: Why the Earth Is Fighting Back – and How We Can Still Save Humanity*. Allen Lane, 2006.
- [LS05] François Lamarche and Lutz Straßburger. Naming proofs in classical propositional logic. In Paweł Urzyczyn, editor, *Typed Lambda Calculi and Applications, TLCA 2005*, volume 3461 of *Lecture Notes in Computer Science*, pages 246–261, Berlin, 2005. Springer Verlag.
- [LSBB92] Reinhold Letz, Johannes Schumann, Stephan Bayerl, and Wolfgang Bibel. SETHEO — A high-performance theorem prover for first-order logic. *Journal of Automated Reasoning*, 8(2):183–212, 1992.
- [OB03] Jens Otten and Wolfgang Bibel. leanCoP: Lean connection-based theorem proving. *Journal of Symbolic Computation*, 36:139–161, 2003.
- [Ott05] Jens 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*, pages 245–261, Berlin, 2005. Springer.
- [Poo03] David Poole. First-order probabilistic inference. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 985–991, 2003.
- [Rob00] J. Alan Robinson. Proof = guarantee + explanation. In Steffen Hölldobler, editor, *Intellectics and Computational Logic*, volume 19 of *Applied Logic Series*, pages 277–294. Kluwer Academic Publishers, Dordrecht, 2000.