Early History and Perspectives of Automated Deduction

Wolfgang Bibel

Darmstadt University of Technology
Also affiliated with the University of British Columbia
Bibel@gmx.net

Abstract. With this talk we want to pay tribute to the late Professor Gerd Veenker who deserves the historic credit of initiating the formation of the German AI community. We present a summary of his scientific contributions in the context of the early approaches to theorem proving and, against this background, we point out future perspectives of Automated Deduction.

Formal logic is still often looked upon as a kind of esoteric doctrine.

Evert W. Beth 1958

The fundamental scientific progress lies in the area of logic and the cognitive sciences.

Pierre Papon 2006

1 Introduction

Gerd Veenker is known in the German Artificial Intelligence (AI) community for his initiative and organisation of the first national AI meeting in Bonn in 1975 and the second one in Dortmund in the same year. This year we celebrate the thirtieth German AI conference and for this reason commemorate of him and of his work. Had he not died so prematurely we could as well have celebrated his seventieth birthday.

Veenker's scientific contributions are in the field of Automated Deduction (AD). In fact, he was the very first German scientist who contributed to this fruitful and still promising field. In the mid-sixties of the last century with his theoretical work and his working systems he was at the forefront of AD internationally. For instance, his system NEU of 1966 realized what only a decade later was reinvented and called UR-resolution (for Unit Resulting). Unfortunately, he was totally isolated in those days when Informatics did not yet exist in Germany, let alone an "esoteric doctrine" like computational logic [Bet58, p.50]. So his contributions have stayed totally unnoticed.

As a courageous pioneer he deserves to be commemorated. We therefore summarize in Section 4 of this paper some of his early contributions. Because these cannot be appreciated without some knowledge about the state of the art in

J. Hertzberg, M. Beetz, and R. Englert (Eds.): KI 2007, LNAI 4667, pp. 2–18, 2007.

AD at those days, we give in Section 2 an account of the first complete theorem proving procedures in first-order logic by Prawitz and Gilmore, with a brief mention of other early deductive systems by Dunham et al., Newell et al., and Davis as well as of McCarthy's seminal contribution of LISP.

Section 3 then describes the advances in AD made in the early sixties. These include unification, Skolem functions, Herbrand universe, clause form, unit resolution, and especially Robinson's resolution. We also point out the circumstances under which these achievements could be obtained and draw a lesson from these observations. This is contrasted with Veenker's situation and his work is analysed in comparison with those advances in the subsequent section as already mentioned. Again we draw a lesson from this comparison for the discipline of Informatics of our days in Germany (or in Europe for that matter).

The paper concludes with some perspectives for AD in the future. Although AD is extremely successful already and offers even more potential, a substantial advance of our systems' performance would require a much better support especially in terms of the quality of education and of the research environment. This is because the formidable challenge of an integration of the many different features in one single system as well as solving important remaining problems will hardly be achievable in the current splintered manner. Even further, there are deep remaining issues to be solved concerning the nature of the underlying logic. Under these considerations we are led to the proposal of the foundation of some European center of excellence for semantics, logic and computation in order to come a step closer to Leibniz' dream of a reasoning machine in the not so distant future.

2 How Automated Deduction Started

In 1957 Aridus Wedberg taught a first year logic course at the University of Stockholm. On one occasion he mentioned to the class the possibility of proving mathematical theorems in first-order logic on a machine. This remark raised the interest of one of the students in the class, namely Dag Prawitz, who decided to realize this idea in practice [SW83, p.200].

First he developed a general procedure for the predicate calculus which we illustrate with the valid formula $\forall x\, Px \rightarrow \exists y\, Py$, shortly F. In order to prove F, we assume it were false and infer a contradiction. That is, we start by assigning the truth value f to it and let the pair (F, f) be the first in a list of subformulas along with truth values. Given the semantics of implication, for F assumed to be false this means that $\forall x\, Px$ must be true, or t, and $\exists y\, Py$ must be false, yielding the next two pairs in the list. Taking the first list item not considered so far, this means that for any constant c in the universe under consideration Pc must be true. In order to mechanize this step for the general case, assume that all constants are enumerated as $c_1, c_2, \ldots$. Since no constant of this enumeration was used before in our example, we simply take its first one and, hence, add the pair (Pc_1, t) at the end of our list. The final item yet to be considered in the list, $(\exists y\, Py,\ f)$, implies that for any constant it must be false. The procedure in

general selects one already used before in this case, ie. here c_1, leading to the final list element (Pc_1, f). Now the list contains two occurrences of the literal Pc_1 with opposite truth values, indicating the expected inconsistency so that the proof now is complete in this case.

For those readers with some familiarity in AD it is clear that this procedure is generating a so-called *tableau* for the given formula except that today the truth values are coded by adding a negation sign in front of the subformula instead of falsehood. In generating the tableau Prawitz' procedure follows precise rules for each of the possible cases, characterized by the outermost logical symbol determining the form of the formula as well as by the associated truth value. Since, for instance, false conjunctions could lead to alternative subcases such a tableau in general consists of a tree with each of its branches being a list like the one in our example, especially in terms of closing the branch by some contradictory pair of literals. Also the selection of constants is a little more complicated than illustrated by our simple example.

Prawitz coded this procedure in a programming language which he designed himself for this special task and wrote a report in Swedish. His father, Håkan Prawitz, hand-translated the program into machine code in 1957. The result was worked over and tested by Neri Voghera, a software expert, for a number of examples in 1958. Thereby he used a computer named Facit EDB, built in Sweden. It featured a core memory of 2048 40-bits machine words and a drum with a capacity of 8192 words. In other words, the first experiments with a general theorem prover for first-order logic were performed in Stockholm in 1958. In 1959 the work was outlined in the discussion of the session on theorem proving at the First International Conference on Information Processing (IFIP) in Paris, the discussion being contained in the proceedings. In 1960 the full paper describing the work appeared in the Journal of the ACM [PPV60].

This short description of the very first work in first-order AD needs to be complemented by a number of comments. First, Prawitz' procedure did not fall from heaven but rooted in well-known work done in Mathematical Logic. Second, there were several other efforts undertaken in that period of time. Third, progress in the early years of AD depended a lot on the programming infrastructure available at the respective location. Let us discuss each of these three important issues in turn.

To begin with the first point, this is not the place to give an outline of the history of logic. There are excellent sources for this purpose such as [KK84]. Also the article [Dav83] summarizes this history with an emphasis on AD and the author's work in it. We want to point out the following highlights in this remarkable history.

Leibniz was the visionary for an instrument to increase the powers of reasoning [Dav83, pp.2ff,14]. Frege's Begriffsschrift [Fre79], with explicit reference to this vision, laid the grounds for all formal languages, logical or programming ones, as well as for logical calculi. Around the 1920's and early 30's the work of Skolem, Herbrand, Gödel, Gentzen, and Jaśkowski as well as the book by Hilbert and Ackermann [HA28] clarified the most important logical concepts and issues

such as completeness, decidability, Skolem functions, Herbrand's theorem, Herbrand universe which for historical correctness should actually be named "Skolem universe", Gentzen calculi, cut elimination, and so forth.

In the mid 1950's a proof procedure by W.V. Quine [vOQ55a] as well as four new and simplified completeness proofs for first-order logic by E.W. Beth [Bet55], K.J.J. Hintikka [Hin55], S. Kanger [Kan57], and K. Schütte [Sch56] were published independently which had an immediate impact on the way early theorem provers were designed. For instance, Prawitz followed closely Beth's formalism in developing his procedure discussed above. While Beth's and Hintikka's systems used proof by contradiction, Kanger and Schütte pursued an affirmative approach. The difference is completely irrelevant from a logical or deductive point of view. But Prawitz and others introduced the contradictory approach which led later researchers to follow this habit.

Let us now come to the second point concerning other early AD efforts. Here we may distinguish four different lines of research, namely first-order theorem proving, propositional methods, heuristic approaches, and decision procedures, which are discussed again in turn.

In 1958 Paul Gilmore, teaching at Penn State (Pennsylvania State University, State College PA), a place made famous by the great logician (and amateur ornithologist) Haskell B. Curry, read an advertisement in the New York Times for a mathematician interested in assisting in a project for proving theorems in Euclidean Geometry. He applied and eventually joined the Mathematics Department of IBM Research at the Lamb Estate in Croton-on-Hudson NY in July 1958 where he worked with Herbert Gelernter (of whom more below). Gilmore had a solid background in Mathematical Logic from a course in Mathematical Logic of S.W.P. Steen at Cambridge University, his studies in Amsterdam with E.W. Beth and A. Heyting, and his earlier collaboration as a postdoc with Abraham ("Abbie") Robinson at the University of Toronto in Canada. Since he had had no experience with or knowledge of electronic computers at the time, Gilmore decided to learn by implementing in assembly language on an IBM 704 Beth's method of semantic tableaux for first-order logic, although eventually the implemented method was "closer to the work of Hintikka".

The resulting program, described in [Gil60], took as input any negated first-order formula in prenex form with its matrix in disjunctive normal form (called standard form in the paper). In this relatively unimportant aspect it differed from Prawitz' procedure (applicable to arbitrary formulas) but otherwise used the same crude search technique for appropriate substitutions. Gilmore thought that his "work is the first working program for quantification theory". He learned of Prawitz' working program only, when he met him at the Paris conference in 1959, and acknowledged the fact in a footnote in [Gil60].

Due to the focus of the present paper we will treat the other three mentioned lines of AD research only in passing. Abbie Robinson had already pointed out in an influential talk at the important five weeks Summer Institute for Symbolic Logic at Cornell University in 1957[1] [Fef03] which was attended also by some

[1] In [Dav83, p.16] the year was stated incorrectly as 1954.

twenty people working in the computer industry including G.W. Collins, B. Dunham, R. Fridshal, H. Gelernter, J.H. North, who presented talks there and are mentioned elsewhere in the present paper, that through Herbrand's theorem first-order theorem proving could be reduced to the propositional level [Rob57]. So propositional theorem proving became interesting not only because of its relevance for the logic of computer hardware but also in our more general context. The first propositional method of a formal kind was used in [DFS60], again programmed for an IBM 704 and also presented at the Paris conference. It used rules such as case splitting and pure literal reduction to rewrite the given expression until the truth value t or f was obtained.

In a sense this work may be seen as a reaction to earlier work by Newell, Shaw and Simon [NSS56] who took a heuristic rather than systematic approach to propositional theorem proving, relying on axioms, forward and backward implicational chaining and modus ponens (cf. [Cor96] for an account of such heuristic approaches). Their program was run on the JOHNNIAC computer from Rand Corp. Russell. Similarly Herb Gelernter relied on a heuristic approach in his realization of a rather successful geometry-theorem proving machine [Gel59] which became operative in 1959. Geometry with its long axiomatic tradition since Euclid's *Elements* was a natural mathematical subject to start with. Through Gilmore's paper [Gil70] Gelernter's geometry theorem prover came to influence modern theorem proving by its use of models to test the consistency of an hypothesis.

The final line of early AD research mentioned above consisted in the implementation of known decision procedures. Martin Davis implemented such a procedure for Presburger arithmetic already in 1954, using the JOHNNIAC of the Princeton Institute for Advanced Study [Dav57]. In effect this then was the very first operative system in AD, although a very restricted one in scope as in achievement. George Collins from the IBM Labs implemented on an IBM 704 parts of Tarski's decision procedure for elementary algebra in order to deal with a variety of problems that could be expressed in that language, a work presented at the same occasion as Davis' [Fef03].

The third and final point in this review of the beginning years in AD refers to the available infrastructure. It constrained the possible success much more than anything else. Prawitz had the luck to find the support of his father and of Voghera who did the extremely time-consuming job of programming and testing. Similarly a well-equipped environment like IBM Research, eg. for Gilmore and Gelernter, turned out to be very helpful.

It was John McCarthy who, on the basis of this experience, put the design and implementation of LISP on the top of his priorities since the availability of such a high-level language could reduce the amount of implementational work dramatically. For instance in [McC59] McCarthy says: "[The Wang algorithm for propositional logic] took about two hours to write the program and it ran on the fourth try." Unfortunately, even more than a decade later computing centers such as those in Germany typically had no implementation of LISP yet available

so that, for a few exceptions, the waste of time in implementational efforts in AI continued for many more years all over the world.

3 Important Historical Advances in AD

In the previous section we have described the very first attempts of automating deduction in first-order logic. Needless to mention that the theorems, which could be proved with those systems, were only rather trivial ones. This initiated half a century of research into improvements of these first procedures. In this section we want to describe some of the main early contributions in this vein.

An immediately obvious drawback of the first procedures was their treatment of substituting constants in a stupid systematic manner. Prawitz was the first in proposing a unificational method instead [Pra60] which used metavariables and substituted constants *by need* rather than according to some fixed sequence. The unification was computed by way of a system of resulting equations to be solved under certain restrictions. These restrictions derived from the well-known variable conditions in Gentzen-type systems. It is worth pointing out that later unification algorithms used a rather similar way of computation. In other words Prawitz deserves the credit for having introduced unification into proof procedures.

None of the procedures mentioned so far allowed function symbols other than constants as it was known from standard logic textbooks that these could be replaced in a certain way by predicates. The first paper [DP60] introducing Skolem functions, hence function symbols, and the Herbrand universe was by Martin Davis, a former student of Alonzo Church, and by Hilary Putnam, a mathematician-turned philosopher. It also proposed the clause form arrangement of the initial data to be refuted, ie. proved by contradiction, which from there on has become a widely used standard. Unfortunately, this standard along with the weakness in Gilmore's procedure beared the ineradicable myth that the use of this standard has computational advantages over an analogue affirmative "clause" form, ie. a representation of the formula in disjunctive normal form to be proved rather than refuted, although the difference is of course totally irrelevant [Bib87]. A further contribution was unit resolution which the authors called rule for the elimination of one-literal clauses.[2] In its treatment of terms it remained ignorant of Prawitz' unificational ideas while in the propositional part it used the independently discovered rules from [DFS60] already described above, all without any implementation though.

In 1960 a postdoc, J. Alan Robinson, at the University of Pittsburgh sent applications to several institutions including the Applied Mathematics Division of the Argonne National Laboratory at Chicago IL and received an offer for a summer research position from it. He eventually decided to rather accept a tenure track teaching offer for logic and philosophy of science from Rice University, but go to Argonne for a summer research position with the task assignment

[2] In [Cor96, Sect.3] the author erroneously attributes the introduction of unit resolution to [WCR64] as did others before him.

by the Division's director, William Miller, to implement on Argonne's IBM 704 the method of the Davis-Putnam paper just mentioned. From a four years employment at duPont he already had a solid experience in assembly language programming. So when he arrived at Argonne in May 1961 the programming was already done, yet in lack of a computer untested. Getting it run, reprogrammed in Fortran, and debugged took considerable efforts which were supported by George Robinson, the head of the division's Programming Development Section. Alan wrote an Argonne Report [Rob61] which in a polished version appeared eventually in 1963 in the Journal of the ACM [Rob63].

In the summer months of 1962 and 1963 Alan Robinson returned to Argonne where George Robinson had already become so excited from the earlier experiments that he started the transition to become a theorem proving researcher. Miller assigned one of his mathematicians, Larry Wos, to join the team consisting of the two Robinsons who first gave him a crash course in logic using Quine's *Methods of Logic*. During these two summer projects, Alan Robinson rediscovered unification and the resolution rule (in 1962) and introduced these two terms into the literature for the first time with the seminal paper [Rob65] (although the publication was delayed by some rumored referee until January 1965).

As to unification we already pointed out that Prawitz' first paper did contain the basic idea behind unification (as did a work by N.A. Shanin – see [SW83, p.30]) and Robinson was strongly influenced by it. He himself says: "I was absolutely inspired by Prawitz".[3] However, already Herbrand's paper [Her30] contained a much more elegant version of it, which basically is the one Robinson published in [Rob65], expressed in recursive definitional rather than algorithmic terms though. Although Prawitz cited this Herbrand paper, he was not aware of this part of its contents (nor was Robinson). As to the resolution rule it was first discovered in [Bla37] already in 1937, coincidentally like Argonne also in Chicago, then rediscovered in [vOQ55b] as *consensus rule* and proposed for use in (propositional) theorem proving in [DN63], presented at Harvard University already in February 1962.

The beauty of Robinson's paper derives from his ability to rediscover these two powerful techniques and merge them with the solid platform for theorem proving which had been achieved by that time (including the purity and subsumption principles). Further, he did so in a mathematically clean and perfect way. This latter point is especially remarkable since Robinson by education was a philosopher with a Master's thesis on *Theories of Meaning Implicit in the British Empiricists Locke, Berkeley and Hume* and a PhD thesis on *Causality, Probability and Testimony*. He attributes the stimulation for his transformation towards a mathematically-oriented scientist especially to his teacher Arthur Pap,

[3] Personal communication (e-mail message of 7 March 2007). – Prawitz' influence can also be seen in the worked example of Davis paper [Dav63, Section 6] which of course references Prawitz' work. This paper does however give not yet any specific hint to a unification-like method à la Robinson beyond Prawitz' equational system as suggested in [Dav83, p.18].

like Putnam a mathematician-turned philosopher. Pap also advised him for his PhD studies to go to Princeton where Alonzo Church educated a whole generation of excellent logicians (including John McCarthy, Marvin Minsky, and Dana Scott), although it was actually Hilary Putnam who acted as supervisor to his dissertation.

Once resolution was available the group at Argonne under the direction of Larry Wos and George Robinson set down to improve its performance by reducing the search space of generated resolvents. Most importantly they introduced factoring, the unit preference, the set of support strategy and implemented resolution with these additional features on a Control Data 3600 [WCR64]. At this point Argonne had become the undisputed world champion in theorem proving. Later, in response to a suggestion by Alan Robinson, they started to concentrate on dealing with equality in a special way, introducing demodulation and paramodulation. Generally, the publication of J.A. Robinson's paper spawned a flood of publications in theorem proving, ninety alone in the years 1967–1970, a period which is well covered by the article [WH83].

This tremendous influence extends up to this day. For instance, the winner of the recent CASC competitions in theorem proving was the resolution-based system Vampire by Andrei Voronkov. This dominance is however not undisputed, a topic which we further pursue in the final section of this paper.

So what is the lesson to be drawn for fertilizing the grounds for future discoveries in our or other fields? The obvious first conclusion is that a top education is the most important prerequisite for excellence. Prawitz, Gilmore, Davis, Robinson, McCarthy and many others are proof to this rule as we described. Of similar importance is the research environment comprising a wise leader like William Miller at Argonne, the right combination of people like Prawitz (logician) and Voghera (software engineer) or Robinson (logician), George Robinson (software engineer) and Wos (Mathematician), and adequate facilities. Argonne had won this competition because it featured both prerequisites in the best possible combination. There was nothing like Argonne in Europe in those days. For instance the Gesellschaft für Mathematik und Datenverarbeitung (GMD) was founded in Bonn not before 1968.

4 Gerd Veenker (1936–1996)

Gerd Veenker was born 9.12.1936 in Lüneburg. His father was a tailor which is worth noting because not only Gerd but also his brother Wolfgang later became university professors. After his school education in Lüneburg until 1957 he studied Mathematics and Physics in Hamburg, München and Tübingen.

Around 1960 he became interested in computers. He and his friend Frieder Schwenkel developed a particular interest in non-numeric computation such as game playing and theorem proving. They studied for instance the respective parts in the proceedings of the first IFIP conference in Paris 1959 which has already been mentioned several times in the preceding two sections.

No guidance by any professor in Tübingen could be expected to further this interest. Possibly not even an appropriate logic course was offered which could have introduced him into the underlying subject. However, Karl Zeller (28.12.1924–20.7.2006), a Mathematics professor with a speciality in limit theory and with experiences from several visits at US universities, in 1960 got a chair (Lehrstuhl Mathematik der Hochleistungsrechenanlagen, ie. mathematics of high-performance computers) which was at the same time responsible for the university's computing center. It featured a Siemens 2002 also installed in 1960. Professor Zeller had a widely open mind and an unusually liberal attitude towards his students in terms of their subjects of interest. So when Veenker decided on his own to concentrate on theorem proving in his Diplomarbeit (Master's thesis) and dissertation he would let him go in this direction and formally play the role of the supervising professor. He was supportive in that he allowed his students to make suggestions for invited colloquium talks (eg. Hermes from Freiburg) as well as for the topic and the literature of seminars officially run under his name. Additionally helpful was the friendly cooperative atmosphere among the members of the small group of students which as "Hiwis" (research assistants) gathered around the computing center.

In 1963 Veenker completed his Diplomarbeit (master's thesis) entitled *Ein Entscheidungsverfahren für den Aussagenkalkül der Formalen Logik und seine Realisation in der Rechenmaschine* (A decision procedure for the propositional calculus of formal logic and its realisation on the computer). The list of its references demonstrates that in the meantime he had read most of the theorem proving literature available by 1962, in particular the papers discussed in the preceding two sections of the present paper. He gives a concise description of the related procedures of Hao Wang [Wan60b], Paul Gilmore [Gil60], Dunham et al. [DFS60], and Davis and Putnam [DP60]. In effect his procedure follows closely the one by Prawitz [PPV60], restricted to the ground level and allowing for the five logical operators $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$. It is programmed in the symbolic low-level programming language PROSA. For the first-order level he gives an outline of an envisaged but not yet implemented procedure.

For a master student this work is truly remarkable if one takes the lack of guidance and logical education as well as the limited computational infrastructure and ressources into account. For instance, he used a number of tricks to fit formulas with up to 50 logical operators into the machine's core memory consisting of 2048 machine words. In contrast to McCarthy's two hours mentioned in Section 2 it took Veenker probably hundreds of hours to get the program to the point of success. He proudly states that for one example his program takes 84 seconds, for which Gilmore's system could not find a proof after 21 minutes. The technical reason for this advantage is Gilmore's costly transformation to disjunctive normal form which Prawitz had already avoided. The work was published shortly after completion [Vee63].

He also published a paper on a program for chess endgames which allow mate in two or three moves. But his focus remained on theorem proving. At the end of 1966 he had a dissertation ready with which he passed the rigorosum at the

beginning of 1967. Again, although working in full isolation, he was fully upto-date with his references to the papers influencing the field during that thime which in those days in lack of anything like a world-wide web required enormous efforts indeed. Apparently he learned of Robinson's 1965 resolution paper after much of his thesis and especially of his program was already completed. This can for instance be seen from his way of treating unification (in his algorithm GLS which is short for German >GLeichSetzung<) which follows the equational style of Prawitz' first paper but now extended to cover general terms since he used Skolem functions. So he may well have – once again – reinvented unification (called Verschmelzung) for general terms. Generally, he follows Davis' representational style [Dav63] and uses Davis' term "linked" in form of the German Verkettung.

On this basis he enumerates all paths through the matrix given by the set of clauses. However, once the procedure has located a connection it eliminates all paths through this connection in one step, focusing next on the two paths obtained from the current one by replacing each of the connected literals by another literal of the same clause. One of these two resulting paths is handled next while the other one is put on a stack for later treatment. So in summary the procedure is advanced in terms of the enumeration of the paths but in comparison with the later connection procedures [Bib87] does not yet restrict the search for a subsequent connection to those with literals in the previously connected clause which in resolution is known as the linearity restriction. This is unfortunate because he already mentions this possibility but only as a preference strategy; apparently he had not seen that this preference can be done without restriction of generality. He does use also the unit preference and the set of support strategy.

In addition to the complete and sound proof procedure just described, Veenker gives an incomplete procedure, called NEU (for new), which today we know as the unit-resulting (UR) strategy combined with unit resolution. The literature thus incorrectly attributes the discovery of this strategy to the authors of [MOW76], where it was introduced under this name, while Veenker invented it already ten years earlier. In lack of a better machine he programmed both procedures still for the Siemens 2002 which in comparison with the CD 3600 at Argonne was at least two orders of magnitude inferior. Taking this into account his running times were well competitive with the state of the art in 1966, a remarkable achievement in view of the lack of what we pointed out at the end of the previous section as the prerequisite for excellence, namely a top education in the subject and a stimulating research environment.

His PhD work was published in [Vee67] but totally ignored by the community. One reason of course was the publication being in German. Another was his total isolation as the only theorem proving specialist in the entire German-speaking area, if not in all of Europe at that time with the sole exception of Prawitz in Stockholm and of Bernard Meltzer at Edinburgh. In addition Veenker was a rather modest and reserved person. When the present author met him in 1969, discussing theorem proving issues, he apparently failed to point to his published work, then as well as in all encounters in later years, which I therefore read

carefully for the very first time not before the preparation of the present paper. This meeting in 1969 gathered Informatics researchers with DFG-funded projects at the Chiemsee and most likely was the very first event were Informatics-based German AI researchers from different institutions got together for presentations and discussions. According to my recollections he did not give a talk (nor did I).

Presumably he had just applied to the DFG for project funds to attack his next goal in theorem proving which was a special treatment of equality in his (incomplete) procedure NEU. This was achieved in the Diplomarbeit of Geerd-Rüdiger Hoffmann and published 1971 in [HV71] in English. It uses a kind of theory unification for equality. In the same year the author presented his first theorem proving paper at the GI Jahreskonferenz in München which in the discussion was heavily attacked by Mr. Hoffmann who pointed out that my Gentzen-type approach had already been shown not to be workable and thus waste of efforts. One might infer from this opinion of his student that also Veenker at that time had given up hope to pursue the line initially taken by him and rather opt for resolution as the winning technique. Perhaps it is for this reason that he also gave no talk about his work at the Oberwolfach meeting on automated theorem proving in 1976 which he attended.

Without having undergone the procedure of Habilitation Veenker received an (associate) professorship for Informatics and Applied Mathematics at the University of Bonn in 1972 where for 34 years he represented AI in his teaching. He never was promoted to a full professorship. In this position he took the initiative for the first official German AI meeting mentioned in the Introduction. His PhD students are Rainer Fröning, Joachim Hertzberg, Eberhard Klein, Knut Möller, Peter Schmidt, Volker Steinhage, and Erich Vorwerk, as far as I could find out. Two of these (Hertzberg and Möller) are now professors. As a professor he was popular with students because of his friendly and warm-hearty personality and hence he supervised a great number of Diplomarbeiten (master's theses). Some of these laid the foundation of academic careers of prolific scientists like Gerd Brewka, Dieter Fox, and also Sebastian Thrun whose autonomous vehicle Stanley in 2005 spectacularly won the DARPA Grand Challenge. Unfortunately, since 1976, Veenker suffered from very serious health problems which apparently kept him from staying scientifically as productive as during the first decade of his career. He died 23.6.1996 at the age of 59 shortly after the deaths of his wife and of his brother in the same year.

Are there any lessons to be learnt from the history of a man who was the first German scientist in the area of AD? In any case it raises a number of questions. One such question was already asked in 1969 by the logician Richard Büchi: "Why did the German logicians not engage in establishing the new field of Informatics?" He posed this question, which is to be seen within the context of Germany during Hilbert's time being the world-leader in logic, at the occasion of the inauguration of the Informatics buildings at the Technical University of München (TUM) to the internationally known German logician Kurt Schütte, a student of Hilbert. Büchi, then at Penn State, was invited to this occasion for a presentation. Schütte could not provide any reasonable answer to his question.

Later, Friedrich L. Bauer, one of the founders of Informatics in Germany joined Büchi and Schütte and, although he had not heard their prior discussion, shocked the two (as well as me) by harshly stating from nowhere: "Logic by now has no more than a peripheral significance for Informatics."

In fact, Bauer's statement provides an explanation for Veenker's unfortunate situation. Establishing a new field like Informatics against the extremely rigid structures of the German academic world was not something to be achieved by decent and modest persons like Schütte who was exclusively devoted to his subject. It required clever and versatile power figures like Bauer who had all the required tricks at their disposal, even though they may have lacked the necessary education in the germane subjects. So Bauer was in fact right insofar as power influence was concerned.

Also it must be said that the German logics community did remain seated in its ivory tower. Besides organizing the International Logic Colloquium (held in Hannover) one of their major concerns in the sixties was the revision of the constitution of the German logic association (Deutsche Vereinigung für Mathematische Logik und Grundlagen der Wissenschaften, or DVMLG) which was bitterly debated for years. Bernays (Basel) was already too old to play a leading role, Specker (Zürich) as a Swiss kept himself at a distance, Büchi left to the US, Schütte did not even dare to respond to Bauer's statement, and so forth. Academically they kept themselves in high regard as an elite which allegedly had good reason to look down to the academically and logically uninteresting computational problems of Informatics. Those who transformed from Logic to Informatics, like the author, became sort of banned, in any case kept in low regard. So people like Veenker, and to some extent also Prawitz who perhaps for those reasons later retreated back into logic and philosophy, academically found themselves sitting between the chairs, in stark contrast to the analog situation in the US where Computer Science was open-minded enough to appreciate topics like theorem proving and respected logicians like Davis did not feel like making their fingers dirty by pondering over the computational issues of proof procedures.

Since these historic frictions are still virulent in various ways, the lesson then is that attempts should be made to become consciously aware of, and overcome, them. In particular this means that Informatics should acknowledge computational logic as one of their fundamental and promising subareas, reflected also in the official characterizations of the field where it is rarely mentioned at all.

5 Perspectives for AD

The author has outlined his credo for the field of AD only recently in the article [Bib06]. We will therefore not repeat these arguments here again except for a few additions especially with respect to the issues discussed in the previous sections.

Recall that at the end of Section 3 we reported of the success of resolution in the mid-sixties of the last century. But it turned out that resolution was no

panacea either in terms of efficiency of proof search. A lot of tricks have to be added and sometimes it is not clear why they work at all. The deeper reason for these problems lies in the fact that up to now the resolution rule has not been thoroughly understood. The present author thought he had achieved such a thorough understanding in his paper [BE97]. But Jörg Siekmann and Graham Wrightson pointed out that the result contradicts an example from [Eis91] so that there must still be some mistake in the obtained result which has not yet been discovered and corrected by anyone. Imagine that forty, in fact nearly 70 years after the discovery of resolution we are still struggling to understand it fully.

The situation is quite different for Gentzen-type theorem proving of the kind which was initiated by Prawitz as discussed in Section 2. With all the work which followed Andrews' matings method [And81] and Bibel's connection method [Bib83] which eliminated the original disadvantages pointed out in the literature discussed in Section 3 we know exactly what kind of improvements could still be made for a better performance of the systems. The difficulty lies in the enormous complexity of the task. In consequence many of the improvements which were worked out theoretically are not yet incorporated in one single system. The CASC-winning system SETHEO [LSBB92] featured many of them but by far not all. One particularly important example is the cut rule which has never been taken care of in any running system except for its very limited consideration in SETHEO. The author has stated a conjecture in [Bib06] which would open a way for its treatment. Another example concerns a refined treatment of variables as already discussed in [Bib87] which has just been worked out in more details in [AW07]. Like these two there are many more issues (heuristic guidance at the meta-level in special theories, learning of strategies, integration of models, etc.) let alone visions like Robinson's "science of proofs-as-explanations" [Rob00], which all are still waiting for being integrated into one single system along with all features scattered in various existing systems. Altogether this amounts to a formidable task.

One should even go a step further in broadening the perspective. Our field today features a great variety of different logics and logical calculi. In [Bib06, Sect.5] I already pointed out the importance of embedding the static logical space, under discussion so far in this paper, into the course of time in an appropriate way. We believe that transition logic [Bib04] achieves this aim in a more natural and effective way by focussing on local transitions rather than on a global transition from one world to another as in modal logics. But even within the static logical space something might be going wrong which could be rooted deeply in some historical decision made long ago. I mention the book [Brü96] which tries a restart of Aristotle's syllogisms in a modern and precise setting. It is just a very first little step in comparison to what modern logics offer. But it could be one of a more constructive nature, which, if followed by further ones, might possibly lead to a logic with better computational features than those which we know today. In fact it might be a good idea to start yet one step further back and abstract with modern AI technologies the logic underlying natural languages from large text corpora.

Achieving all these tremendously complex tasks according to the lesson from the end of Section 3 would require an excellent education of brilliant minds along with a research environment which I cannot spot anywhere in the world. Therefore I put forth the suggestion to *found sort of a Max-Planck Institute on the European level with such a broad basic research mission*. It would have to combine research excellence in a variety of related fields including semantics of natural language, logic and philosophy of logic, psychology of inferencing and proofs, cognitive science, knowledge representation and reasoning, and above all computation.

I want to conclude by pointing to the relevance of deduction in all kinds applications in virtually every area, independent of any of these future advances. This is because of the fundamental importance of reasoning in all human activities [Bib03]. Especially through the semantic web and through knowledge systems the importance of deduction will surely grow tremendously [Bib07]. So I completely share physicist Pierre Papon's conviction as expressed in his statement cited at the beginning of this paper [Pap06, p.10].

Acknowledgments. The text owes a lot to a number of people from whom the author got first-hand information about those early days. This includes a touching text about his personal developments by Alan Robinson and a delineation of his encounter with theorem proving by Paul Gilmore as well as helpful discussions with both. My picture about Gerd Veenker's early career derives from extensive material and support provided by Joachim Hertzberg, who was one of the initiators for this paper, and from lively discussions or exchange of letters with Margarete Zeller, Tübingen, Wilhelm Niethammer, Karlsruhe, Manfred Reimer, Dortmund, and Frieder and Trude Schwenkel, Hamburg/Winsen. Further information was received from Thomas Christaller, Martin Davis, Hans Langmaack and Sebastian Thrun. I am grateful to all of them. For any errors I take of course full responsibility.

References

[And81] Andrews, P.B.: Theorem proving via general matings. Journal of the ACM 28, 193–214 (1981)

[AW07] Antonsen, R., Waaler, A.: Liberalized variable splitting. J. Automated Reasoning (2007)

[BE97] Bibel, W., Eder, E.: Decomposition of tautologies into regular formulas and strong completeness of connection-graph resolution. Journal of the ACM 44(2), 320–344 (1997)

[Bet55] Beth, E.W.: Semantic entailment and formal derivability. Mededlingen der Koninklijke Nederlandse Akademie van Wetenschappen 18(13), 309–342 (1955)

[Bet58] Beth, E.W.: On machines which prove theorems. Simon Stevin Wis- en Naturkundig Tijdschrift 32, 49–60, Reprinted in [SW83, 76–90] (1958)

[Bib83] Bibel, W.: Matings in matrices. Comm. ACM 26, 844–852 (1983)

[Bib87] Bibel, W.: Automated Theorem Proving, 2nd edn. Vieweg Verlag, Braunschweig (1987)

[Bib03] Bibel, W.: Lehren vom Leben – Essays über Mensch und Gesellschaft. In: Sozialwissenschaft, Deutscher Universitäts-Verlag, Wiesbaden (2003)

[Bib04] Bibel, W.: Transition logic revisited, 2004 (Submitted)

[Bib06] Bibel, W.: Research perspectives for logic and deduction. In: Stock, O., Schaerf, M. (eds.) Reasoning, Action and Interaction in AI Theories and Systems. LNCS (LNAI), vol. 4155, pp. 25–43. Springer, Heidelberg (2006)

[Bib07] Bibel, W.: Wissenssysteme und Komplexitätsbewältigung. In: Leiber, T. (ed.) Denken und Handeln in einer komplexen Welt – Festschrift zum 60. Geburtstag von Professor Klaus Mainzer, Hirzel Verlag, Stuttgart (2007)

[Bla37] Blake, A.: Canonical Expressions in Boolean Algebra. PhD thesis, University of Chicago, Illinois (1937)

[Brü96] Brüning, W.: Grundlagen der Strengen Logik. Königshausen und Neumann, Würzburg (1996)

[Cor96] Cordeschi, R.: The role of heuristics in automated theorem proving – J.A. Robinson's resolution principle. Mathware & Soft Computing 3, 281–293 (1996)

[Dav57] Davis, M.: A computer program for Presburger's algorithm. In: Summaries of talks presented at the Summer Institute for Symbolic Logic, Princeton NJ, pp. 215–233, Institute for Defense Analysis (1957), Also contained in [SW83, 41–48]

[Dav63] Davis, M.: Eliminating the irrelevant from mechanical proofs. In: Proc. Symposium for Applied Mathematics XV, Providence, RI, pp. 15–30 (1963), Also contained in [SW83, 315–330]

[Dav83] Davis, M.: The Prehistory and Early History of Automated Deduction. In: Siekmann, J., Wrightson, G. (eds.) Automation of Reasoning 1 – Classical Papers on Computational Logic 1957–1966, pp. 1–28. Springer, Berlin (1983)

[DFS60] Dunham, B., Fridshal, R., Sward, G.L.: A non-heuristic program for proving elementary logical theorems. In: First International Conference on Information Processing, Paris, pp. 282–285. Unesco House (1960), Also contained in [SW83, 93–98]

[DN63] Dunham, B., North, J.H.: Theorem testing by computer. In: Proc. Sympos, Brooklyn NY, pp. 173–177. Polytechnic Press (1963) Also contained in [SW83, 271–275]

[DP60] Davis, M., Putnam, H.: A computing procedure for quantification theory. Journal of ACM 7, 201–215 (1960), Also contained in [SW83, 125–139]

[Eis91] Eisinger, N.: Completeness, Confluence, and Related Properties of Clause Graph Resolution. Pitman, London (1991)

[Fef03] Feferman, S.: Alfred tarski and a watershed meeting in logic: Cornell, 1957. In: Hintikka, J., et al. (eds.) Philosophy and Logic – In search of the Polish tradition. Synthese Library, vol. 323, pp. 151–162. Kluwer Acad. Publ., Dordrecht (2003), `http://math.stanford.edu/~feferman/papers/cornell.pdf`

[Fre79] Frege, G.: Begriffsschrift. Louis Nebert, Halle (1879)

[Gel59] Gelernter, H.: Realization of a geometry theorem-proving machine. In: Proc. First Intern. Conf. on Information Processing (IFIP), Paris, pp. 273–282. UNESCO House, (1959), Also contained in [SW83, 99–122]

[Gil60] Gilmore, P.C.: A proof method for quantification theory: Its justification and realization. IBM J. Research Develop. 4, 28–35 (1960)

[Gil70] Gilmore, P.C.: An examination of the geometry theory machine. Artificial Intelligence 1, 171–187 (1970)

[Gol71] Goldfarb, W.D. (ed.): J. J. Herbrand — Logical writings. Reidel, Dordrecht (1971)

[HA28] Hilbert, D., Ackermann, W.: Grundzüge der Theoretischen Logik. Springer, Heidelberg (1928)

[Her30] Herbrand, J.J.: Recherches sur la théorie de la démonstration. In: Travaux Soc. Sciences et Lettres Varsovie, Cl. 3 (Mathem., Phys.) (1930), Engl. transl. in [Gol71]

[Hin55] Hintikka, K.J.J.: Form and content in quantification theory. Acta Philosophica Fennica 8, 7–55 (1955)

[HV71] Hoffmann, G.-R., Veenker, G.: The unit-clause proof procedure with equality. Computing 7(1-2), 91–105 (1971)

[Kan57] Kanger, S.: Provability in Logic. PhD thesis, University of Stockholm (1957)

[KK84] Kneale, W., Kneale, M.: The Development of Logic. Clarendon Press, Oxford (1984)

[LSBB92] Letz, R., Schumann, J., Bayerl, S., Bibel, W.: SETHEO — A high-performance theorem prover for first-order logic. Journal of Automated Reasoning 8(2), 183–212 (1992)

[McC59] McCarthy, J.: The Wang algorithm for the propositional calculus programmed in LISP. In: McCarthy, J. (ed.) Symbol Manipulating Language Memo 14, Artificial Intelligence Project, MIT, Cambridge MA (1959), Quoted in [Wan60a, p.232]

[MOW76] McCharen, J., Overbeek, R., Wos, L.: Problems and experiments for and with automated theorem proving programs. IEEE Transactions on Computers C-25, 773–782 (1976)

[NSS56] Newell, A., Shaw, J.C., Simon, H.A.: The logic theory machine. IRE Trans. Information Theory IT-2, 61–79 (1956), Also contained in [SW83, 49-73]

[Pap06] Papon, P.: Die Wissenschaft, Zeichen der Zeit. FTE Info – Magazin über europäische Forschung (An interview) 50, 9–11 (2006)

[PPV60] Prawitz, D., Prawitz, H., Voghera, N.: A mechanical proof procedure and its realization in an electronic computer. J. ACM 7, 102–128 (1960)

[Pra60] Prawitz, D.: An improved proof procedure. Theoria 26, 102–139 (1960), Also contained in[SW83, 159–199]

[Rob57] Robinson, A.: Proving theorems (as done by man, logician, or machine). In: Summaries of Talks Presented at the Summer Institute for Symbolic Logic, Communic. Res. Div., Princeton, New Jersey, Institute for Defense Analysis (1957), Also contained in [SW83, 74–76]

[Rob61] Alan Robinson, J.: Gamma I: A general theorem proving program for the IBM 704. Technical Report ANL-6447, Argonne National Laboratory, Chicago IL (1961)

[Rob63] Alan Robinson, J.: Theorem proving on the computer. Journ. ACM 10(2), 163–174 (1963), Also contained in [SW83, 372–383]

[Rob65] Alan Robinson, J.: A machine-oriented logic based on the resolution principle. Journal of ACM 12, 23–41 (1965), Also contained in [SW83, 397–415]

[Rob00] Robinson, J.A.: PROOF=GUARANTEE+EXPLANATION. In: Hölldobler, S. (ed.) Intellectics and Computational Logic – Papers in Honor of Wolfgang Bibel. Applied Logic Series, vol. 19, pp. 277–294. Kluwer, Dordrecht (2000)

[Sch56] Schütte, K.: Ein System des verknüpfenden Schließens. Archiv f. Mathematische Logik und Grundlagen der Wissenschaften 2, 55–67 (1956)

[SW83] Siekmann, J., Wrightson, G. (eds.): Automation of Reasoning — Classical Papers on Computational Logic 1957-1966, vol. 1. Springer, Berlin (1983)

[Vee63] Veenker, G.: Ein Entscheidungsverfahren für den Aussagenkalkül und seine Realisation in einem Rechenautomaten. Grundl.stud. aus Kybernetik u. Geisteswiss 4, 127–136 (1963)

[Vee67] Veenker, G.: Beweisalgorithmen für die Prädikatenlogik. Computing 2(3), 263–283 (1967)

[vOQ55a] van Orman Quine, W.: A proof procedure for quantification theory. J. Symbolic Logic 20, 141–149 (1955)

[vOQ55b] van Orman Quine, W.: A way to simplify truth functions. American Mathematical Monthly 62, 627–631 (1955)

[Wan60a] Wang, H.: Proving theorems by pattern recognition, Part I. Comm. ACM 3, 220–234 (1960), Also contained in [SW83, 229–243]

[Wan60b] Wang, H.: Toward mechanical mathematics. IBM Journ. Res. Develop. 4, 2–22, Also contained in [SW83, 244–264] (1960)

[WCR64] Wos, L., Carson, D., Robinson, G.A.: The unit preference strategy in theorem proving. In: AFIPS Conf. Proc., Washington DC, vol. 26, pp. 615–621. Spartan Books (1964)

[WH83] Wos, L., Henschen, L.: Automated theorem proving 1965–1970. In: Siekmann, J., Wrightson, G. (eds.) Automated Reasoning 2 – Classical Papers on Computational Logic 1967–1970, vol. 2, pp. 1–24. Springer, Berlin (1983)