Proof Presentation

Jörg Siekmann

DFKI Saarbrücken, DE
siekmann@dfki.uni-sb.de

The talk is based on a forthcoming book is about the human-oriented presentation of a mathematical proof in naturallanguage, in a style as we may find it in a typical mathematical text book.

How can a proof be other than human-oriented? What we have in mind is a deduction systems, which is implemented on a computer, that proves–with some human interaction–a mathematical textbook as may be used in an undergraduate course. The proofs generated by these systems today are far from being human-oriented and can in general only be read by an expert in the respective field: proofs between several hundred (for a common mathematical theorem), for more than a thousand steps (for an unusually difficult theorem) and more than ten thousand deduction steps (in a program verification task) are not uncommon. [1] Although these proofs are provably correct,they are typically marred by many problems: to start with, that are usually written in a highly specialised logic such as the resolution calculus, in a matrix format, or even worse, they may be generated by a model checker. Moreover they record every logical step that may be necessary for the minute detail of some term transformation (such as, for example, the rearrangement of brackets) along side those arguments, a mathematician would call important steps or heureka–steps that capture the main idea of the proof. Only these would he be willing to communicate to his fellow mathematicians–provided they have a similar academic background and work in the same mathematical discipline. If not, i.e. if the proof was written say for an undergraduate textbook, the option of an important step may be viewed differently depending on the intended reader.

Now, even if we were able to isolate the ten important steps – out of those hundreds of machine generated proof steps – there would still be the startling

[1] The Argonne deduction system OTTER, the old MKRP-System or more recently the SPASS system from Saarbrücken, SETHEO in München or Voronkov's System VAMPIRE can generate and represent search spaces of more than a billion clauses and the proof as thus found may be up to several hundred or even more than a thousand steps long. The interactive VSE program verification system of the DFKI at Saarbücken has generated proofs of more than ten thousand steps for proving one program assertion, when it verified the control software of a telecommunication network consisting of about 10,000 lines of source code. This required proofs for many thousand assertions (and some of these proofs were then a little less than 10,000 steps long). The early verification of a complex hardware component such as the microprocessor FM8501 that was verified with the Boyer–Moore System required about 150 pages of formulas of specifications and lemmata in the Boyer– Moore Logic and was carried out in about thirteen man-months. The complexity of today's verification task surpasses all of this by far.

Dagstuhl Seminar Proceedings 05431
Deduction and Applications
http://drops.dagstuhl.de/opus/volltexte/2006/561

problem that they are usually written in the 'wrong' order. A human reader might say: 'they do not have a logical structure'; which is to say that of course they follow a logical pattern (as they are correctly generated by a machine), but, given the convention of the respective field and the way the trained mathematician in this field is used to communicate, they are somewhat strange and ill structured.

And finally, there is the problem that proofs are purely formal and recorded in a predicate logic that is very far from the usual presentation that relies on a mixture of natural language arguments interspersed with some formalism. The proof is far from such a presentation in the sense that even if all predicate, function and constant symbols were replaced by their natural language counterpart and even if all the logical formalisms were replaced by its natural language equivalent (such as $'\wedge'$ and ')' by implies and so forth) the resulting proof in natural language would still be a far cry from what we consider natural, even it it were ingeniously augmented by the usual mathematical phraseology such as 'thus follows', 'em as can easily be seen' or 'now we have the following cases' followed by 'quod erat demonstrandum'.

Is there a typical mathematical textbook with a universally accepted way of proving theorems? Style, notation and level of abstraction in the presentation of a mathematical argument have changed considerably over the centuries. They are still changing today.

Moreover every mathematical discipline and even every subarea within this discipline has its own jargon and its battery of proof techniques and general methods, that make it distinct and immediately recognisable by an experienced mathematician of this field.

Apart from these considerations there is a more principle point to the problem of what constitutes a typical proof. Until the last century a mathematical proof was essentially a convincing argument a mathematician would use to persuade his colleagues in believing his theorem. While there was considerable controversy on the form of such an argument, dating back at least to Euclid's Elements, it essential means of presentation was that of natural language prose – albeit highly stylised and augmented with some formal notation. Now what constitutes a 'convincing argument' is open to debate and mathematical truth was – and in spite of all logicism still is – established through a social process of conjectures and refutations carried out anew by each generation.

This point of view was challenged and appeared to be irrevocably superseded by the turn of the century, when the foundational studies in mathematical logic and the slow but sure recognition of the importance of Frege's Begriffsschrift marked the beginning of logicism.

Hilbert's program with its success in the twenties and thirties seemed to render the whole of mathematics into a formal and essentially mechanical enterprise: a proof is a formal object – itself subject to mathematical rigour and analysis – that proceeds from the hypothesis (the axioms) through a sequence of well understood and simple formal operations based on general accepted inference rules, to its conclusion, the actual theorem. Although there is considerable

variation within this point of view, which is the subject of proof theory (some of its standard material is covered in the first part of this book) the essence of this holds universally true for all logical calculi: axioms and theorems are purely syntactical objects and the intermediate steps are based on formal rules of inference that could in principle be carried out mechanically on a suitable machine.

Thus the touching seventeenth century prophecy of Gottfried Wilhelm Leibniz "that, when the new language is perfected, men of good will desiring to settle a controversy on any subject whatsoever will take their pens in their hands and say calculemus – let us calculate, and carry out the argument in a purely formal manner within the calculus ratiocanator"– this vision had in principle become true for the mathematical disciplines by the middle of our century and logicism began to spread into other disciplines as well, as hallmarked inter alia by the Vienna Circle.

As Martin Davis notes in his Prehistory and Early History of Automated Deduction, this view was not entirely unchallenged: "Henry Poincare realised perfectly well that if the claim of the logicians were to be taken seriously, the possibility of mechanising human reason would be very real. But the very absurdity of such a possibility which threatened everything creative and beautiful in mathematical thought, showed in fact the logicians claims need not be taken seriously". Poincare expressed his argument by reductio ad absurdum picturesquely as follows: "Thus it will be readily understood that in order to demonstrate a theorem, it is not necessary or even useful to know what it means. We might replace geometry by the reasoning piano imagined by Stanley Jevons; or, if we prefer, we might imagine a machine where we should put in axioms at one end and take out theorems at the other, like the legendary machine in Chicago where pigs go in alive and come out transformed into hams and sausages. It is no more necessary for the mathematician than it is for these machines to know what he is doing." Be this as it may modern textbooks in many mathematical fields now reflect the logical point of view: set theory, model theory, some books on algebra and many textbooks on logic itself are typical examples of this purely formal approach in the sense that their proofs are carried out entirely by syntactical operations on formulae – often with very little natural language explanation in between. From Whitehead and Russell's Principia Mathematica to the Bourbaki group of mathematicians , there have been many attempts to reconstruct the whole of mathematics on the basis of a few basic principles, from which follows all that is known in mathematics by a few syntactical operations, i.e. logical inferences.

The first mechanical calculating devices for the four numerical operations, addition, multiplication and their inverses, constructed by W. Schicard, B. Pascal and G. W. Leibniz in 1623, 1642 and 1671 respectively relied on two fundamental developments: the arithmetical calculus had developed from an art (for example calculating the payrolls for the Roman legionnaires) to a purely mechanical application of rules in the algorithms for the four basic arithmetic operations. The craftsmanship and the art of engineering of the time had developed in France and Germany to a point, where mechanical clocks based on springs, cogwheels

and so on became a reality,that could now also be used to build a mechanical calculator for numbers.

Similarly two basic developments had taken place for the art of formal reasoning by the mid of our century: the development of logical calculi that only required mechanically applicable syntactic operations and the advent of the first electronic computer (such as Konrad Zuse's Z1 and his Plankalkül in 1936) to carry them out. The first computer–generated proof of a mathematical assertion was found fully automatically in 1954. It became the hallmark of a new area of formal reasoning by proving the mind-boggling theorem that the sum of two even numbers is again even.

The computer program that found the proof implemented a decidable fragment of the first–order calculus known as 'Pressburger's arithmetic' and was developed by Martin Davis at the American Institute of Advanced Studies on a 'JOHNIAL' machine. At the same time A. Newell, J.C. Shaw and H. Simon developed a computer program based on entirely different principles that proved dozens theorems from Principia Mathematica. This program implemented some general heuristics for proving theorems as could be observed in empirical psychological studies with students.

These two programs, first presented at the Dartmouth Conference in 1956, nowadays considered as the cradle of artificial intelligence, spawned a debate that has dominated automated reasoning ever since: should a strong system for automated theorem proving rely on the advance in computer hardware and search for a proof within an appropriate logical calculus or else should it try to emulate human behaviour as observed in a given mathematical task? This controversy,that became more general and fundamental later on in the whole field of artificial intelligence is still unresolved – in spite of the success of the chess laying program Deep Blue that beat the world champion Kasparov in 1997 and appeared to swing the pendulum to the former point of view. Marvin Minsky suggested in 1961: "It seems clear that a program to solve real mathematical problems will have to combine the mathematical sophistication of H. Wang [2] with the heuristic sophistication of Newell, Shaw and Simon" – and exactly how this is to be achieved has been a controversial cornerstone in the field of automated deduction ever since.

Automated deduction systems, while still inferior in comparison to some human capabilities, have developed considerably in the four decades that followed the Dartmouth Conference. They are now routinely used for many formal reasoning tasks in mathematics, computer science and artificial intelligence, and have been instrumental in proving open mathematical problems as well as the correctness of complex software and hardware components of a computer. While this book is not on automated deduction systems as such, we assume that there

[2] Hoa Wang worked in the logic-oriented tradition of Martin Davies and received the "Milestone Award for Automated Theorem Proving" in 1983 from the American Mathematical Society for his outstanding contributions to the field. In 1958 he developed a program at IBM and later at Bell Labs that proved 350 theorem of Principia Mathematica. It was by far the strongest system at the time.

is a computer based proof development environment that may either operate interactively with a mathematician (as the tactic based deduction systems do) or fully automated (as most resolution style theorem provers do) or else in a mixed mode as in our preferred system ? MEGA, that combines proof of planning with the best of these two worlds and probably currently comes closest to Marvin Minsky's view of 1961. In either case, the output of any of these systems is a formal proof in some logical calculus, usually ill structured and lengthy. This formal proof is to be translated into a human oriented style of presentation.

In spite of all logicism, why is the majority of mathematical papers and textbooks still written in natural language at the highly personal, albeit socially influenced informal level that is typical for a particular mathematical area? While there is no single and immediate answer, the point we want to emphasise in this book is this: the important discovery that a proof is a formal object, which is itself subject to mathematical analysis sometimes tends to overemphasise formality – and not necessarily needs to de–emphasise its role as a means of communication between fellow mathematicians and human beings, where clarity, pleasing aesthetics, brilliance of the argument and personal style have an important role to play. As the Bourbaki group of French mathematicians note in the first volume on the foundation of modern mathematics: "Elements of Mathematics: Theory of Sets" : "If formalised mathematics were as simple as the game of chess, then once our chosen formalised language had been described there would remain only the task of writingou tour proofs in this language, just as the author of a chess manual writes down in his notation the games he proposes to teach, accompanied by commentaries as necessary.But the matter is far from being as simple as that, and no great experience is necessary to perceive that such a project is absolutely unrealisable: the tiniest proof at the beginning of the theory of sets would already require several hundreds of signs for its complete formalisation... Hence formalised mathematics cannot in practice be written down in full, and therefore we must have confidence analogous to that accorded by a calculator or an engineer to a formula or a numerical table without any awareness of the existence of Peano's axioms, and which ultimately is based on the knowledge that it has never been contradicted by facts. We shall therefore very quickly abandon formalised mathematics, but not before we have carefully traced the path which leads back to it". While this is no longer entirely true given the advances in proof theory and more expressively logics tailored to specific mathematical areas, the essence of the above quotation is still valid: a proof when communicated to a human fellow mathematician is more of a schema or recipe that could in principle be expanded into a formal object but in practice never is. And the working mathematician hardly cares.

The point of view we would like to advance in this book is that there can be a welldefined computational relationship between the formal and the informal. Although we may only see the beginning of it today, completely formalised mathematics can now be written down in full by a machine and translated back to a surface representation that is easily understandable and possibly pleasing for a human mathematician. John Alan Robinson, author of the seminal paper:

A Machine–oriented Logic Based on the Resolution Principle , posed a challenge to the research community in his invited talk at the Conference on Automated Deduction, which was held in 1990 at Argonne National Labs, the birthplace of the resolution principle. A delighted audience that celebrated the quarter–century anniversary of its invention was captured by a brilliant and humorous talk that centered around the topic of the lengths of proofs, their ways of communication and falsification and after many examples given, culminated in the case of the proof of the so–called Enormous Theorem of finite group theory. This is the classification theorem for finite simple groups, which states that such groups fall into only finitely many natural classes, whose description is part of the statement of the theorem. The original proof is published in over 500 different papers by several hundred authors over a period of about thirty years. The late professor Gorenstein published a three–volume summary of the proof and then, a little later, he published a twenty-page article in the Scientific American, in which he condensed the proof even further. Hence the challenge: is there a general procedure for condensing or summarising long proofs into shorter ones? If so, then 'iterating' this procedure on a given proof would, as Alan whimsically conjectured, produce a series of shorter and shorter proofs converging to the one word proof: 'OBVIOUS'.

Could all this be done by a machine or is it here, where the stipulated watershed between human and artificial intelligence can be located? The PROVERB systems that is presented in the second and third part of this book, achieves a compression/abstraction ratio somewhere between one and two orders of magnitude (depending on the particular case at hand and measured in the number of proof steps of the original machine generated proof at the calculus level). The condensed and abstracted version of the proof is then fed into a natural language generator that plans and restructures again and finally produces a mathematical proof in the form and content as we may find it in a typical mathematical textbook.The theorem proving system ? MEGA, of which PROVERB is a part, was used inter alia to prove a large part of a textbook on automata theory that was widely used at an undergraduate level in the German computer science curriculum.

The mechanically generated proofs were then fed into the PROVERB system.

Apart from interesting variations and differences most proofs, even those that were generated entirely without human intervention, were often in shape, proof technique and natural language diction very similar to those mathematical arguments, the author Peter Deussen had used when he wrote his book 30 years earlier with paper and pencil – except that he wrote his book in German, whereas the PROVERB system generates English prose in its current version. The resulting documents, and since 1995, when we first obtained these results, a many more mathematical proofs have been transformed and translated into natural language, are an interesting source of inspiration for further research in artificial intelligence techniques for mathematics and natural language generation. It shows that the two aspects of a proof, namely as a means of communication and as a means of assurance, beautifully captured in Alan Robinson's equation

Proof = Certification + Explanation
have a computational relationship that is subject to serious scholarship just as traditional proof theory was earlier on in the previous century.

The views expressed in this book are formed by the intellectual traditions of mathematical logic (and mathematics), artificial intelligence, natural language generation and automated reasoning and finally by cognitive psychology. To make the book self contained as far as possible so that the reader unfamiliar with one or other of these traditions may be able to follow the argument, we give some of the proof-theoretical background and an introduction to natural language generation the first part and concentrate in the second part on the subject of human oriented but computer generated proof presentation. The development of the proof transformation and final translation programs are based on a sequence of diploma and PhD theses that are built upon each other starting in 1976 with Peter Kursawe, and Christoph Lingenfelder followed by iaosong Huang in 1995 and Armin Fiedler in 1998 and by Helmuth Hosagelis work on the linguistic aspects. While most of the technical material and system description is based on Xiasong's thesis and more than a decade of scientific labour, the writing and final preparation for this book represents joint work of all the authors, including some additional material by Erica Melis and Andreas Maier. The individual contributions of the authors and other contributors are marked in the appropriate places within the text.

Jörg Siekmann

London, 2005

Forthcoming: Jörg Siekmann, Armin Fiedler, Dov Gabbay, Helmuth Horacek: "Proof Presentation" forthcoming, Elsevier,2006