Editorial

This volume starts a new subline of Lecture Notes in Artificial Intelligence, called

AI-SYSTEMS

focusing on the most important systems and prototypical developments in artificial intelligence.

We have chosen an initiative of the year 2004 as the starting date to commemorate the 50th anniversary of the first time a mathematical theorem was proven by a computer system: Martin Davis' implementation of the Presburger Fragment of first-order logic proved the mind-stretching result that the sum of two even numbers is again even.

While the first volumes in this category will present – for historical reasons – today's internationally most relevant systems from the field of automated reasoning, we shall soon solicit detailed presentations of individual systems as well as systems from the other major subareas of AI.

October 2005

Jörg Siekmann
Alfred Hofmann

Foreword

Our compiler, Freek Wiedijk, whom everyone interested in machine-aided deduction will thank for this thought-provoking collection, set his correspondents the problem of proving the irrationality of the square root of 2. That is a nice, straight-forward question. Let's think about it geometrically – and intuitively.

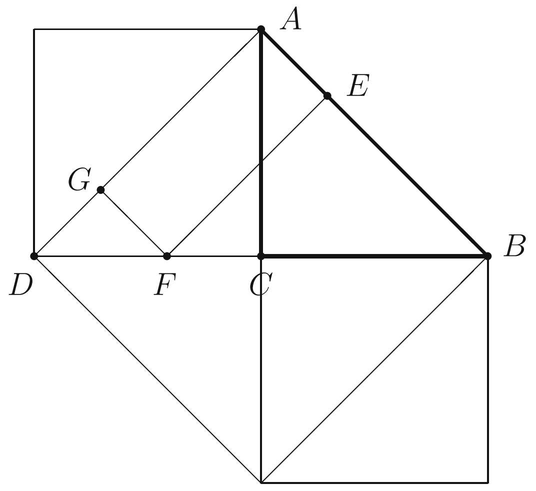
The original question involved comparing the side with the diagonal of a square. This reduces to looking at an isosceles right triangle. For such a triangle, the proof of the Pythagorean Theorem is obvious. As we can see from the figure, the squares on the legs are made up of *two copies* of the original triangle, while the square on the hypothenuse requires *four copies.* The question is whether a leg is commensurable with the hypothenuse.

Call the original triangle ABC, with the right angle at C. Let the hypothenuse $AB = p$, and let the legs $AC = BC = q$. As remarked, $p^2 = 2q^2$.

Reflect ABC around AC obtaining the congruent copy ADC. On AB position E so that $BE = q$. Thus $AE = p - q$. On CD position F so that $BF = p$. Thus $DF = 2q - p$. The triangle BFE is congruent to the original triangle ABC. EF is perpendicular to AB, the lines EF and AD are parallel.

Now, position G on AD so that $AG = EF = q$. Since $AEFG$ is a rectangle, we find $AG = q$. Thus, $DG = FG = AE = p - q$. So, the triangle DFG is an isosceles right triangle with a leg $= p - q$ and hypothenuse $= 2q - p$.

If there were commensurability of p and q, we could find an example with *integer lengths* of sides and with the perimeter $p + 2q$ a minimum. But we just constructed another example with a smaller perimeter p, where the sides are also obviously integers. Thus, assuming commensurability leads to a contradiction.



As one of the contributors remarks, this reduction of (p, q) to $(p - q, 2q - p)$ is very, very easy to accomplish with algebra – and the observation avoids the lemmas about even and odd numbers in finishing the required proof. But, what does this really mean? As I have often told students, “Algebra is smarter than you are!” By which I mean that the laws of algebra allow us to make many steps which combine information and hide tracks after simplifications, especially by cancellation. Results can be surprising, as we know from, say, the technique of generating functions.

In the case of the isosceles right triangle (from the diagonal of the square), an illumination about meaning can be obtained best from thinking about the Euclidean Algorithm. For a pair of commensurable magnitudes (a, b), the finding of “the greatest common measure” can be accomplished by setting up a sequence of pairs, starting with (a, b), and where the next pair is obtained from the preceding one by subtracting the smaller magnitude from the larger – and by replacing the larger by this difference. When, finally, equal pairs are found, this is the desired greatest common measure. (And, yes, I know this can be speeded up by use of the Division Algorithm.)

In our case we would have: (p, q), $(p-q, q)$, $(p-q, 2q-p)$, $\ldots$. If we do some calculation with ratios (as the ancient Greeks knew how to do), we remark that the Pythagorean Theorem gives us first $p/q = 2q/p$. (Look at the triangles to see this: all isosceles right triangles are similar!) From this follows $(p-q)/q = (2q-p)/p$. Now switch extremes to conclude that $p/q = (2q-p)/(p-q)$. This shows that the third term of our run of the Euclidean Algorithm gives a pair with the *same* ratio (when the larger is compared with the smaller) as for the initial pair. In any run of the Euclidean Algorithm, if a ratio ever repeats, then the algorithm *never finishes.* Why? Because the pattern of larger and smaller quantities is going to repeat and, thus, no equals will be found. Hence, the magnitudes of the original pair are *incommensurable.* Indeed, Exodus knew that $a/b = c/d$ could be *defined* by saying that the two runs of the algorithm starting with (a, b) *and* (c, d), respectively, have the same patterns of larger and smaller.

In later centuries it was recognized that the Euclidean Algorithm is directly connected with the (simple) continued fraction expansion. Moreover, as Lagrange showed, the infinite, eventually periodic, simple continued fractions give *exactly* the positive irrational roots of quadratic equations (with integer coefficients). Perhaps, then, it might have been a more interesting challenge to prove the Lagrange Theorem itself, but probably fewer groups would have responded.

Alas, I have never spent any extended time with the provers/checkers represented in this collection. I did invest many profitable hours in using the equational theorem prover, Waldmeister: it is small, yet very effective on many problems involving equational deductions. Unfortunately, some theorem provers based on first-order logic do not really incorporate all the techniques of equational provers, so with certain problems time and/or space may run out before finding a proof. It is imperative that implementers of these systems now take advantage of specialized algorithms if ever mathematicians are going to become interested in using a machine-based method.

We can also see clearly from the examples in this collection that the *notations* for input and output have to be made more human readable. Several systems do generate LaTeX output for the discovered proofs, but perhaps additional thought about formatting output might be valuable. The Theorema Project (system 12 in the present list) made readability of proofs a prime requirement, and their report shows their success. However, the objective Bruno Buchberger set originally for the project was to produce a tool for pedagogic use, not research. Thus, the power of their system does not yet reach what, say, the HOL-based systems surveyed in this report have. Also, the question of the *discovery* of a proof is different from *checking* a proffered proof. Hence, any features that make a system *interactive* – and many in this collection have such – do help in finding proofs through experimentation.

Over about a decade I developed undergraduate courses using *Mathematica*. One effort was directed at Discrete Mathematics, and my colleague, Klaus Sutner, at Carnegie Mellon has expanded that effort several fold with excellent success. Most of my own thought went into a course on Projective Geometry, basically an introduction to plane algebraic curves over the complex field. What I found via the use of computer algebra was that theorems can be *proved* by asking for simplifications and interaction between equations. Technically, I used not just commutative algebra but also an implementation of the algebra of partial differential operators acting on multivariate polynomials. The details are not important, as the point was that the user of *Mathematica* had to enter the right questions and control the choices of appropriate cases (say, after a factorization of a polynomial) in order to reach the desired conclusions. In other words, though there was automatic verification and generation of algebraic facts, there is not a *deductive* facility built into *Mathematica*. And I wish there were! Some very good progress has been made in the system, however, in simplifications of logical formulae involving the equations and inequalities over the real field. But welcome as this is, it is not *general-purpose* logical deduction.

Computer algebra systems have become very powerful and are used both for applications (say, in computer-aided design of complicated surfaces) and in research (say, in group theory, for example). But we have to note that though effective, *proofs* are not generated. The user of the system has to believe that the system is doing the simplifications correctly. Usually we are able to accept results on faith, and we are happy to see what is discovered, but, strictly speaking, a proof is lacking. For a wide-ranging discussion of such issues, the reader may consult "A Skeptic's Approach to Combining HOL and Maple" by John Harrison and Laurent Théry, which appeared in the Journal of Automated Reasoning, vol. 21 (1998), pp. 279–294. (This is also to be found on John Harrison's WWW page.)

So what we have here is a dilemma to be faced by implementors of proof systems. On the one hand, interaction and experimentation can be considerably speeded up by using automatic simplification of logical and algebraic expressions – and one can hope even by rules that the user specifies himself. Alternately, new methods for large-scale Boolean satisfaction algorithms might be employed. On the other hand, for verification (either by humans or by another part of the system), *checkable proofs* have to be generated and archived. Computers are so fast now that hundreds of pages of steps of simplifications can be recorded even for simple problems. Hence, we are faced with the questions, “What really is a proof?” and “How much detail is needed?” Several different answers are offered by the systems surveyed here. But, is there a canonical answer that will satisfy the test of time – and be relevant as new systems are put forward in the future? And don’t forget that probabilistic proof procedures (say, for checking whether a large number is prime) also involve the question of what constitutes a proof.

Large searches present another vexing block for understanding what a system has accomplished. The original attack by computer on the Four Color Conjecture is a case in point. As discussed in the introduction by Wiedijk, objections have now been eliminated by showing that the method for generating the necessary cases is correct, even though the total run of the program is not humanly surveyable. On the other hand, as noted, work by Hales to eliminate criticisms of his solution to Kepler’s Conjecture, though making progress, still continues. Of course, there will always be people who will say that such computer calculations, no matter how well designed – and with verified design principles – do not really give us proofs. They may even say, “How do you know that there was not some quantum-mechanical glitch that threw the computer off?” Running the program again with the same results will not be convincing either. But, what I think will silence the nay-sayers is the development of *whole suites* of general-purpose programs for solving new problems. Not wishing to criticize the work on Four Color Conjecture or on Kepler’s Conjecture, but it often seems that a big effort is put into solving one single problem, and that’s it. When proof assistants constitute a research tool that (suitably minded) mathematicians use daily for work, then there will be recognition and acceptance. This has already happened for computer-algebra systems and for chip-design verification systems. I remain optimistic that we will sooner and not later see real progress with solid mathematics proof systems.

But human imagination can always outstrip the capabilities of machines. To bring this point home in a very clear way, I think that the two delightful books by Roger B. Nelson, *Proofs Without Words: Exercises in Visual Thinking* (1993) and *Proofs Without Words II: More Exercises in Visual Thinking* (2000), published by The Mathematical Association of America, can give a deep fund of examples and questions about how proofs can be formalized. In the books there are, of course, many of the proofs of the Pythagorean Theorem, probably the most proved theorem in mathematics. Two I especially like involve facts about similar triangles: see proof VI on p. 8 of the first volume, and XI on p. 7 of the second. Proofs like these involve augmenting the original figure by what are often called “auxiliary lines”. I particularly hated this method of proof in geometry when I first saw it in school. The teacher would introduce these constructions in a way like a magician pulling a rabbit out of a hat. It did not seem fair to make a hard problem easy, because there was little made obvious about where these helpers came from. After a while, I learned to do this stuff myself, and then I liked it. But training machines to do this is another question.

A quite different method is given on p. 142 of the first book. The puzzle is taken from the article by Guy David and Carlos Tomei, “The problem of the calissons”, published in the American Mathematical Monthly, vol. 96 (1989), pp. 429–431. A calisson is a French candy in the shape of two equilateral triangles joined at an edge. The problem has to do with arrangements of these (as tiles) in a hexagonal box. Thinking of a triangular grid in the plane, a calisson is the appropriate “domino” for this grid. On the usual grid of squares, there are just two orientations of a rectangular domino: vertical or horizontal. The triangular grid allows three orientations, however. What David and Tomei remarked is that when the different orientations are colored in three colors, the fact about the balance of colors used becomes “obvious” – if the observer is used to optical illusions.

It is amusing that the late Edsger W. Dijkstra in his handwritten, privately circulated note, EWD 1055, of 5 July, 1989, strongly rejected this method of argument. He writes that they “give a very unsatisfactory treatment of the problem ... [and] come up with an elaborate non proof.” His note gives a rigorous proof, but I think it is one that would need some effort to automate. (Dijkstra’s notes can be downloaded over the Internet, by the way.)

N.G. de Bruijn has also written on this problem in a brief paper dating initially from May 1989, which he circulated privately after 1994. In his note he remarks:

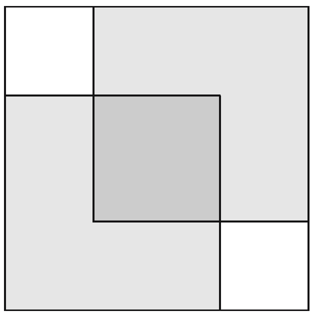
> The proof sketched [by David and Tomei] gives a very amusing intuitive argument, interpreting the box with calissons as a two-dimensional drawing of a collection of unit cubes in three dimensions. In the present note a more formal argument will be given, and a stronger result will be obtained. For any box, hexagonal or not, it will be shown that if it can be filled with calissons, then the number in each direction is uniquely determined by the box. These numbers can be found if we just know both the volume of the box and what we shall call the *weight sum* of the box. Moreover it will be shown that this weight sum can be expressed as a kind of discrete contour integral taken along the boundary of the box.

Indeed, Dijkstra proves the same result about each box determining the three numbers of orientations. But, it may be that de Bruijn adds something additional about how the shape of the box gives these numbers. Dijkstra’s proof seems more “combinatorial”, while de Bruijn’s is more “analytical”. But a closer reading might show they had equivalent ideas. Another question these authors may not have considered is the connections between the various tilings of a box. In the simple case of a hexagonal box, the counting result might be proved by “rewriting”. That is, the tiles in the different orientations might be driven to different corners of the box by replacing, one after the other, a small hexagon of three tiles by one of its rotations. And it might be that the space of tilings is “path-wise connected” – in the discrete sense that one could pass from one to the other by these elementary steps. For boxes of different shapes, it might be another story.

This puzzle is only one of many amusing tiling problems which show that even simple combinatorial questions often require special techniques to automate owing to the large number of possible configurations to be considered, as many authors have remarked. In many cases, the solutions do not depend on general theorems but require searches crafted solely for the particular problem. The problem of the calissons may be an example in between; if so, it might be more interesting to study than those requiring "brute force". And all such examples make us again ask: "What is a (good) proof?"

Note Added 22 May 2005

It was just brought to my attention that the late Stanley Tennenbaum told many people about a proof of the irrationality of root 2 he discovered in the 1960's. It is of course possible that the proof has been noted often before, especially as it is not so far from what is discussed above. However, it can be explained as a 'proof without words' involving no calculations beyond what is seen in the figure.



Suppose a square with integral sides is equal in area to the combination of two, smaller, congruent squares. Place the smaller squares inside the first square at two diagonally opposite corners. The two squares will have to overlap (Why?), making another square covered twice by them. But in the other corners there are two much smaller squares left uncovered. Inasmuch as the areas were supposed to add up, the two small squares must also add up to the central, overlapping square. (Why?) But the sides of these three smaller squares are obtained by subtraction, and hence must have integral values. Hence, there can be no minimal, integral configuration where the sum of two equal, integral squares adds up to another integral square.

Dana S. Scott
<dana.scott@cs.cmu.edu>
University Professor Emeritus
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Preface

This volume is not a collection of papers but a collection of small formalizations. These formalizations all formalize a proof of the same very small theorem: the irrationality of the square root of two. After each formalization there is a description of the system used for that formalization, again not in the form of a paper but in the form of answers to a standard 'questionnaire'.

The systems shown in this volume are most of the systems that one should consider if one is interested in the formalization of mathematics, as it is very lucidly described in the QED manifesto. The purpose of this volume is not to find out which system is 'best'. The aims of the systems are too diverse to be easily comparable in a linear fashion. Instead it tries to showcase all those systems, to make clear what formalizations in all those systems look like. The main point of the volume is that these systems can be very different.

I would like to thank all the people who wrote all these very interesting formalizations. Also I would like to thank Dana Scott for his willingness to write the Foreword for this volume. I would like to thank Henk Barendregt for the concept of this collection. Finally I would like to thank Jörg Siekmann for offering to have this volume published in the LNAI series, to commemorate the 50th anniversary of the first computer-checked proof.

October 2005 Freek Wiedijk