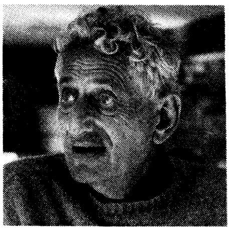
What Do I Know?
A Study of Mathematical Self-Awareness

Philip J. Davis



Philip J. Davis received his Ph.D. from Harvard under Ralph Boas. He has taught at Harvard, Maryland, the University of Utah, and Brown. He was Chief, Numerical Analysis Section, National Bureau of Standards for five years. He was a Guggenheim Fellow in 1956–57. His extensive work in numerical analysis and applied mathematics includes the books Interpolation and Approximation (1963), Mathematics of Matrices (1964), Numerical Integration (with P. Rabinowitz, 1967), Circulant Matrices (1979). The Mathematical Experience, written jointly with Reuben Hersh of the University of New Mexico, won an American Book Award for the year 1983.

Professor Davis received the 1960 Award in Mathematics of the Washington Academy of Sciences, the MAA Chauvenet Prize in 1963, and the Lester R. Ford Award of the MAA in 1982.

> Tout homme crée sans le savoir
> Comme il respire
> Mais l'artiste se sent créer
>
> Every man creates unwittingly
> As he breathes
> But the artist feels himself create.
>
> Paul Valéry
> Inscription for the Palais de Chaillot

Introduction. I am confronted with a mathematical problem P. I think about P and perhaps I work on it a bit with pencil and paper. What then is the state of my knowledge about P and its solution?

At the simplest level, I might say "I have solved P" or "I know how to solve P; it will take me a few minutes to push through the details." But I might say "I don't see how to solve P. I can understand it, it makes sense to me, but it has me stumped. Perhaps, if I think about it for a while, I'll be able to crack it." If I am a student, I may turn to my teacher or to a fellow student: "Help me, I'm stuck." If I am a professional, I may consult a fellow professional or turn to a professional reference book.

But there are many additional states of knowledge that have been described and studied. Thus, it might be interesting and useful to describe some of the wider possibilities that exist. It is the purpose of this article, then, to arrange several dozen current states of mathematical knowledge in an informal taxonomy and to comment on them.

Out of typographical necessity, the paradigms that have been set up to describe knowledge states have been presented in linear order. I do not think that these paradigms can be ordered linearly or even partially, for there is no comprehensive and totally adequate structure for the representation of knowledge in general, and if one limits oneself to mathematical knowledge, the situation is hardly any better. Therefore, the states that are listed should not be interpreted as either ordered, independent, or complete. Many more states could have been delineated and commented on.

To be known, to be knowable; to be proved, to be provable; to be computed, to be computable; to be decided, to be decidable; to be true, to be false; to be verified or verifiable; to intuit; to have evidence for; to doubt; to know that one knows; to know that one has proved; to know that one can't prove—these are some of the epistemological environments (or clouds) out of which the states of mathematical knowledge are formed. In many instances, compounding is significant: to know that Fermat knew. A third compounding may set us adrift in a fog of attenuated meaning: to know that one knows that one knows.

The whole of mathematics can be examined and presented from the point of view of an emerging epistemological structure. This is not the traditional way, which is to locate time in the immediate past or to suppress it entirely. States of current knowledge—as opposed to the knowledge itself—are part of the mental set that lies behind emerging material. The mathematician is aware of them. They have a shadow existence. They certainly affect his work, but they are not often described. Formalized mathematical exposition does not admit them.

States of Knowledge

0. The problem is: what is the mathematical problem?

Where do problems come from? Before any of us enter the mathematical arena, well-formulated mathematical problems are already on the scene (or lurking behind the arras) waiting to be solved. What typically happens in a prolonged investigation is that problem formulation and problem solution go hand in hand, each eliciting the other as the investigation progresses.

The formulation of a problem brings matters to a head. It is a platform from which further development proceeds. An unsolved problem calls for a solution. It represents a state of tension that must be resolved.

1. *P* is a problem that makes mathematical sense.

To arrive at this state already requires considerable knowledge and experience. Let P be the problem: Is the integer 68492304 a square number? I assert that P makes sense to me. If P is the problem "Is the integer 68492304 a soft number?" I assert that P makes no sense to me, because I do not know what the word 'soft' means in a numerical context.

Let P be the problem: Does an application of the Hopf construction to $SO(m) \times S^{m-1} \to S^{m-1}$ result in a stable mapping?

I do not know whether this problem makes sense because, although I recognize some of the symbols, I do not recognize the mathematical subfield with any precision. To find out whether it makes sense might involve me in a considerable literature search or the tapping of knowledgeable individuals. Presumably I got the problem from some source, and referral to this place would provide me with a lead.

But suppose I consider the problem P: Find the resolvent of $xy \to \downarrow \circ \int$, which I obtained by putting down mathematical-like symbols and phrases in a random fashion. It may already make sense to someone—though I doubt it—or it may make sense in the future. A theory of soft numbers or of the resolvents described above is easily imagined.

But consider problem P: What is the probability $p(n)$ that the integer n is a prime? Does this problem make sense? On the one hand, the primality of n is not a matter of probability; n is either a prime or it is not. Yet there is a heuristic context in which one might assert $p(n) = 1/\log n$ and on this basis argue fruitfully about a variety of problems about the distribution of prime numbers.

It would appear, then, that to arrive at the knowledge that a problem makes sense is a considerable achievement, and surely when it comes to formulations that are intended as mathematical models of the real world, that knowledge may be a supreme achievement.

One of the principal problems of linguistics is this: describe the mechanisms by which a person is able to recognize as meaningful sentences which he has never heard before. There are numerous theories here, but none is adequate. A similar problem may be formulated for mathematical discourse: describe the mechanisms which govern the formation and function of meaningful mathematical sentences. It is clear that within exceedingly narrow subfields such mechanisms exist and can be described. After all, I doubt very much if the reader has ever seen this sentence: find $8360294 \times (191)^2 \div .0280114$. Yet the reader perceives it immediately as meaningful. It would not be difficult to formalize by recursive definition the idea of meaningful sentences in numerical arithmetic (Gödelian construction). This kind of thing may be useful but is misleading. When it comes to the totality of mathematics, the only description of what is meaningful is to be found in the whole mathematical culture itself, written and unwritten, past and present. This is the only way in which mathematics can preserve its open-ended quality.

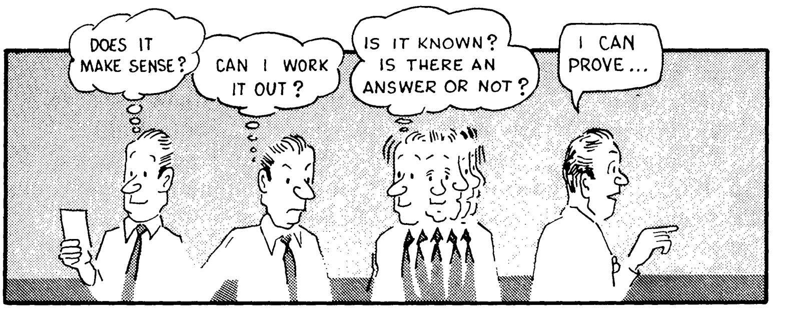
2. I do not know how to do *P*, but perhaps I can work it out for myself.

In problems that students are given to drill on, the answer or the method for finding the answer is usually quite close to what is in the book or to what the teacher has put on the blackboard.

Occasionally a problem is assigned of greater difficulty or in which the element of originality is greater. Even with such problems there is a psychologically beneficial presumption—namely, that *the answer is known*: in the back of the book, in the author's mind; some place.

A researcher, on the other hand, may not know whether the problem makes sense, whether a sensible question has an answer, or how the answer is obtainable.

There are no rules for working out all problems. Such rules would indeed constitute a royal road to mathematics. Finding such roads—insofar as they can be found—is currently a psychomathematical topic of considerable interest. Descriptions of practical heuristic strategies have been given by Pólya, Schoenfeld, and many others. It is a topic in which artificial intelligence and decision theory overlap [G1], [N1], [P1], [R4], [S1].



3. I do not know the answer to *P*, but perhaps it is known.

In what sense is it known and how can I find out?

Perhaps my friend knows the answer and will tell me. In the Renaissance, mathematical knowledge might have been a personal secret. Today it might be company-confidential or restricted in the sense of national security. Perhaps I can find the answer in a book, in a table, in an index or an encyclopedia. Perhaps I should look in the Mathematical Reviews. Perhaps there is a key word in context (KWIC) listing. Perhaps I can query a computerized information system. They are occasionally useful.

The creation of mathematics has always gone hand-in-glove with the creation of systems for information dissemination. We must have material which establishes, references, and cross indexes current knowledge. In view of the fact that the amount of mathematical material is very large and is increasing rapidly (some authorities have estimated 200,000 new theorems per annum) we are led to the very important question of how best to store, arrange, index, categorize, cross reference, represent, display, mathematical information. How does one make information available upon search? This is the key problem of *information technology*.

One does the best that one can, and the best is often mediocre. In a general context, the systematic representation of knowledge poses enormous difficulties. As one of the basic tasks of Artificial Intelligence, it has led to such schemata as scene analysis and frame analysis. The influential article in this field is by Marvin Minsky [M2], but Minsky's knuckles have been rapped by Herbert Dreyfus in [D6].

In the mathematical field, with scope limited and with a self-proclaimed reputation for precision, one might think the problem is less severe. But no. Concepts are born, concepts get buried, get their names changed or their contexts altered; and these are only a small part of the difficulties [H1], [R2], [R3].

Readers might like to try their hand at finding a reference to the following simple problem and its solution.

> *P*: Given four distinct points in the plane, under what circumstances can I pass an ellipse through them? (Note the vagueness and open endedness here: 'under what circumstances.')

Is it in Apollonius, who wrote a major treatise on conics in 225 B.C? Is it in Pappus (A.D. 300), who summed up large parts of classical mathematics? Is it in Kepler, who fiddled around marvelously with ellipses, fitting data by them? Is it in Salmon, who wrote the definitive mid-Victorian treatise on conics? Is it given as a problem in some elementary text on analytic or projective geometry? I have looked and I don't know.

Another issue here is: should I try to find out what is known about P prior to attempting the problem myself? Although one should be as well informed as possible, the desired information may be hard or costly to come by. What is the cost-effective policy?

With regard to buried concepts and altered contexts, consider this example. In one of the first issues of the American Journal of Mathematics (1879), Arthur Cayley, world-renowned British mathematician, works out the number of asyzygetic covariants of degorder (θ, μ) for the binary seventhic. I do not know what these words mean. I have a vague feeling of what kind of mathematics this is likely to be, and I would suspect that whatever it means, it would be said differently today. How can an information system be devised that will make Cayley's result comprehensible to me quickly and cheaply?

4. I do not know whether *P* does or does not have an answer.

Example. P: Is there a 2×2 matrix X with real elements such that

$$X^3 + X^2 + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} X = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}?$$

I do not know the answer. Perhaps the theory of matrix equations can tell me [G2]. My doubts that P may not have an answer are based on some experience with matrix equations. Thus, I can prove that there is no 2×2 matrix X with complex elements such that $X^2 = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$. This leads me to wonder about the equation posed above.

The beginner thinks that all problems that make mathematical sense must have answers (otherwise, why should they have been posed as problems?). The sophisticate knows this may not be the case. When it comes to problems that arise in applied science, the answer status is very complex.

5. I can prove that *P* has an answer.

Notice that one can distinguish between knowing that P has an answer and knowing what the answer to P is.

Examples.

P: What is the first digit (from the left) of 12×13? P has an answer. The answer is 1.

P: What is the $10^{100^{100}}$th digit of pi? Although P has an answer, I don't know what the answer is. Mankind may never know what the answer is.

6. I can prove that *P* has no answer.

Sometimes one says paradoxically: the answer is that there is no answer.

Examples.

P: Find an integer x such that $x^2 = 2$.

P: Find a ruler and compass construction of a square whose area is that of a given circle.

P: Find an integer solution to the system

$$2x + y = 2$$
$$4x + y = 5.$$

P: Find integers x, y, z, not all zero, such that $x^3 + y^3 = z^3$ (Fermat).

7. While I can prove that *P* has no answer along the lines specified, I can show that by enlarging the context of the problem, *P* has an answer in the enlarged context.

In the enlarged context, *P* may be said to have a generalized answer. The process of context enlargement is one of the major principles of growth in mathematics.

Examples.

P: Solve $x + 1 = 0$. *P* has no answer among the positive integers, but it has an answer among all integers.

P: Solve $x^2 = 2$. *P* has no answer among the rationals (Pythagoras) but it does among the quadratic irrationals.

P: Sum $\sum_{n=1}^{\infty} 1/n!$. *P* has no answer among the algebraic numbers (Hermite), but it does among the reals.

P: Solve $x^2 = -1$. *P* has no answer among the reals, but it does among the complex numbers (Cardan).

P: Find a function δ such that

$$\delta(x) = 0 \quad \text{for} \quad x \neq 0, \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x) = 1.$$

P has no answer among standard functions, but it does among "generalized" functions (P. Dirac, L. Schwartz, others).

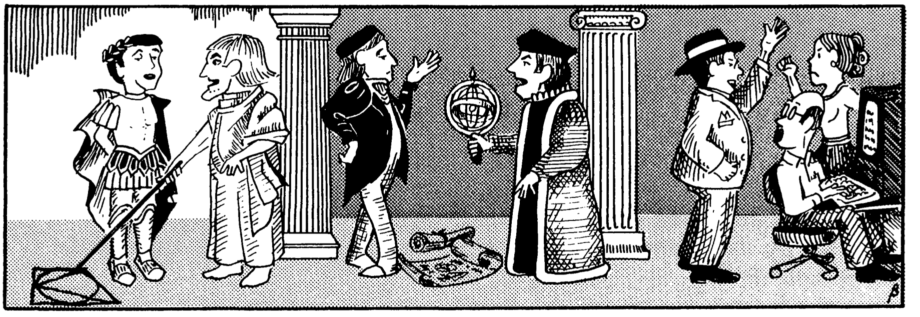
Example. The well known "sixteen puzzle," of sliding the squares into proper numerical order, demonstrably has no solution. But a generalized solution exists. Lift the squares into the 3rd dimension and you will be able to arrange the numbers in order.

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

Moral: If you are cornered, go through the ceiling.

Example. The series $1! - 2! + 3! - 4! + \cdots$ is divergent. Nevertheless, Euler gave it meaning and utility and later mathematicians gave it a more rigorous foundation within the theory of summability [B1].

What we have here is the success of the *Theatre of the Absurd*. It should be noted that in each of the cases given, the necessity for enlarging the context was perceived *prior to the formalization of the standard context*. In a certain sense, the standard context and its extension emerge simultaneously, and later workers assign the standard context a psychologically anterior status. This may be a misreading of mathematical history.



We may often proceed—history bears this out—in a formal way without knowing how rigorously to enlarge the context or without perceiving the necessity for doing so. And yet, we may still arrive at an interesting theory, contradictory in parts and consistent in other parts, which has useful applications. We may then speak of the success of the absurd and cite it as an instance of what J. J. Schwartz has called "amusing naiveté," particularly on the part of physicists. The success of the absurd raises the question that when the context is eventually enlarged, what role does the subsequent deabsurdification play? For example, the interpretation of $\sqrt{-1}$ and all other complex numbers as pairs of real numbers endows them with a concrete, demystified, ontological status; what role does this representation really play, considering that the theory of functions of a complex variable was well developed prior to the promulgation of this interpretation, and that most contemporary texts on the subject, having mentioned the demystification proceed to ignore it? It is clear, then, that context enlargement can equally well be interpreted as naiveté on the part of the mathematical profession.

The frequent and dramatic successes of the theatre of the mathematical absurd raise further questions: Given a problem P which demonstrably has no answer, is it possible to enlarge the context so that P then has an answer? If this is possible, is such an extension useful beyond the formal triumph?

To get a feeling for what is involved here, consider P: Find a 2×2 matrix M with complex entries such that $M^2 = \left(\begin{smallmatrix}0 & 1\\ 0 & 0\end{smallmatrix}\right)$. I can prove that there is no such matrix M. Yet I know two different ways of enlarging the context so that the matrix equation can be "solved." Presumably there are other ways. Why should I promote such an enlargement? For its own sake? For the sake of connections to other parts of mathematics?

Or, consider P: Find a regular polyhedron of 32 vertices. Now I can prove (Euclid, Euler) that there is no solution to this problem. Suppose, in the absence of outside motivation, I wish to enlarge the context so that what is an impossibility becomes a possibility. More latitude is required; something must give. Should one expand the notion of regularity, or the notion of what constitutes a polyhedron? Should one change the dimensionality $d = 3$? Or should one simply not worry about how rigorously to make the extension; simply make the assumption that there is a regular polyhedron of 32 vertices and plunge blithely ahead and see what—if anything—comes of it.

8. I can prove that P has an answer, but I don't know what the answer is.

Presumably grave difficulties stand in the way.

Example. P: Find a complex number z such that $z^8 + 7z^7 + 18z^4 + 283z + 18 = 0$. By the Fundamental Theorem of Algebra, I can prove that P has an answer. (In fact, it may have as many as eight different answers.) But there are numerical difficulties that stand in the way of obtaining the answer.

This leads us straight to the general question: What "is" an answer, really? A more productive formulation is: What kind of an answer is acceptable?

Just as a problem may develop organically together with its solution, notions of what is an answer will grow organically, hand-in-hand with notions of what is desirable, feasible, or available.

Let us suppose that P has been stated in a way that it is solved when we have exhibited a mathematical entity or structure with certain properties. For example, "Find a number such that . . . " The problem P will be considered solved when I have exhibited, produced, displayed, a positive integer and then have demonstrated that that integer indeed satisfies the stated requirements. Now, in order to exhibit, produce, display an integer, I must display it in some specific representation. Thus, as a first display,

$$\mathrm{D}_1: \quad n \text{ is } 16804392.$$

This is to be interpreted as an integer in decimal form. But there are many representational schemes available. I might say, for example,

$$\mathrm{D}_2: \quad p \text{ is } 380127 \text{ in base } 9,$$

or

$$\mathrm{D}_3: \quad q \text{ is } 2 \times 2 \times 651, \quad \text{or} \quad r \text{ is } 10^{100} + 9^{99}.$$

These are in the form of small programs and with some effort I can display such a number in the form D_1. I might also say, as in the process of Gödelization,

$$\mathrm{D}_4: \quad s \text{ is } 3^{2^2} \cdot 5^{3^{2^2}} \cdot 7^{2^4} \cdot 11^{2^5} \cdot 13^{2^{5^7}}.$$

Note that it is not possible to exhibit the number s in the form D_1. It is too large. Gradually, one is led to pass from the notion of representation of numbers in terms of a base or something similar to the possibility of a more general type of specification. It is usual to talk about *descriptions*. We can contemplate such things as

D_5: n is the greatest integer $\leqslant \sum_{p=2}^{\infty} 1/p!$, where p runs over the primes.

D_6: n is the $10^{100} + 9^{99}$th digit in the decimal expansion of π.

D_7: n is $f(50)$, where $f(x) = x - 10$ if $x > 100$ and $f(x) = f(f(x + 11))$ if $x \leqslant 100$ (the "91 function").

But one has to watch how descriptions are stated:

D_8: n is the number such that $n = n - 1$. (But then $n^2 = n^2 - 2n + 1$, and hence $n = \frac{1}{2}$ (!).)

Allowing real numbers momentarily in our discussion, consider the following:

D_9: Ω is the halting probability of a given universal computer whose program is generated randomly.

The number Ω is Chaitan's number, and the consequences of the digits of Ω being displayed have amusing neo-cabalistic consequences for mathematics. For example, knowledge of its digits permits the solution of all finitely refutable mathematical conjectures [B2].

If descriptions are phrased in common language or in metalanguage, there is the distinct possibility of paradox [H2].

D_{10}: n is the smallest uninteresting integer. (The smallest uninteresting integer is surely interesting.)

D_{11}: n is the integer whose description is "the smallest integer whose description requires more than ten words." But here (Berry's Paradox) is a description of n that requires exactly ten words.

It is clear, then, that the notion of what is or isn't a description of a number is not a matter that is completely formalizable, and this situation diffuses back to the impossibility of formalizing what kind of thing can be considered the displayed answer to a problem.

9. *P* has an answer. Here it is (in some representation). I can check that it is an answer. I can prove that in some sense it is the only answer.

Example. *P*: Find real numbers x and y such that

$$\begin{cases} x + 2y = 3 \\ 2y + 3y = 5. \end{cases} \qquad (*)$$

The answer is $x = 1, y = 1$. By back substitution, I check that it is an answer. By elimination, I can prove that it is the only answer. Shall we say that P is completely solved? Perhaps. Suppose that the above problem is reformulated to read P': Find necessary and sufficient conditions in order that (*) holds. Then $x = 1, y = 1$ constitutes such a set of conditions. But so also does

$$\begin{cases} 2x + 5y = 7 \\ 5x + 8y = 13. \end{cases}$$

Is this useful? Maybe. In this way, one can contemplate an infinite class of mathematical statements that are answers to P'. Is there a best or ultimate answer?

10. Problem *P* is under discussion. *A* claims to have found the answer and displays it. *B* disputes *A*'s claim.

This is not infrequent, but hopefully for mathematics, it is a transitory state. (In some historical instances, it is a state that has lasted several hundred years.) B may convince A, and then, back to the drawing board for A. Or maybe not. Perhaps the mathematical community sides with B against A. Will this settle things? Not necessarily. People make mistakes. Judgements change. Contexts may be enlarged. What was once heterodox can become orthodox.

Despite claims to the contrary, the process of verification that an answer is an answer has not and cannot be totally formalized.

Example. The comedy of errors, lasting more than a hundred years, centered on the Euler–Poincaré formula $V - E + F = 2$, where V is the number of vertices, E is the number of edges, and F is the number of faces of a polyhedron. See [D4], [L1].

Example. At the 3rd International Congress of Mathematics in Heidelberg at the turn of the century, J. König delivered a paper showing that Cantor's conclusion—that the real numbers can be well ordered—was erroneous. An error in König's proof was pointed out to him and he retracted his paper.

Example. Cantor proved that the real numbers are nondenumerable by using his so-called diagonal process. Wittgenstein [W2] says that all Cantor has achieved is the computation of a real number that is distinct from a given sequence of real numbers. He says, further, that to proceed from there to Cantor's conclusion is meaningless verbiage, because there is no precise idea of the ordering of all the real numbers. Many people including the physicist Percy Bridgman [B5] have asserted that Cantorian set theory is meaningless.

11. I do not know what the answer to *P* is. I have an algorithm for finding the answer. If the algorithm exists, it will exist with the answer.

I am interpreting the word 'algorithm' in the theoretical sense, as opposed to a real-world program to be run on a real-world machine. A good portion of elementary mathematics is algorithmic in character, and many students equate a knowledge of mathematics with a knowledge of specific algorithms for the working out of certain types of problems.

Example. P: Is the number $n = 168050300000071$ a prime? Apply the algorithm: divide n by $2, 3, \ldots,$ and see whether there is a remainder. If there is always a remainder, n is prime. This algorithm will exist in a finite number of steps. But now consider the following complications. Suppose P is any problem whatsoever. Apply any algorithm which puts the computer into a loop. Then this is an algorithm which, if it exits, exits with the answer to P.

Or suppose P: Are there 1000 consecutive 7's in the decimal expansion of π? Apply the algorithm: compute successively better and better approximations to π and examine them for consecutive 7's. Exit if you find 1000 of them. I suspect that this algorithm will exit after a finite number of steps, but I can't prove it one way or another.

12. *P* has an answer. I do not know what it is. Here is an algorithm for getting the answer. Here also is a program I have written for the computer *C*. I can prove that *C* will output with the answer to *P* in less than *M* minutes.

Here indeed, is a felicitous state of affairs, particularly if P is important and M is small. Such "proofs" would involve a mixture of mathematical reasoning and physical assumptions or experience. Such a state is rare (if one requires that the mathematical part be strict). But given a class of similar problems and a program on C for them, it is possible to build up considerable experience on C which will allow the conclusion that one will generally get the answer to P in $\leqslant M$ minutes.

Example. Machine C has nanosecond addition for single precision numbers.

13. *P* has an answer. I do not know what it is. Here is an algorithm for getting the answer and a program for computer *C*. I can prove that the program exits in a finite number of steps, but this will not take place for 10^{10} years.

Example. Evaluation of an nth order determinant by the direct definition in terms of elements involves $n!$ additions. This is an enormous number when n is, say, of the order of 20. In such a state, back to the drawing board.

But what about the next possibility?

14. *P* has an answer which is obtainable on a computer in a finite number of steps. I can prove that no algorithm exists which will get me the answer in fewer than *N* steps.

Here, as in the last paradigm, we are dealing with questions of *computer complexity*. Upper bounds to the number of steps required are often easily obtained; lower bounds are more difficult, but there are such [A2], [S3].

Example. It has been proved that to sort n items into a linear order requires at least $\log n!$ (approximately equal to $n \log n$) comparisons.

On the basis of the operation count, one can go over to an estimate for a minimum time for any specific computer. Statements about what future machines might or might not deliver are based upon our current knowledge of physical principles.

In this connection also, notice that a person, working backward, might easily come up with a problem to which *he knows the answer*, but which the computer might take an inordinate time to answer.

Example. Multiply two large primes and obtain the product. Present the product to the computer and ask for the prime factors. This problem is of importance in recent cryptographic schemes.

15. A mathematical statement *P* makes sense. I can prove that I cannot prove *P* on the basis only of certain axioms. I can also prove that I cannot disprove *P* on the basis of those same axioms.

Under these circumstances, one says that *P* is *independent* of the assumed axioms. As the Vermonter says, "You cannot get there from here."

Example. I can prove that one can neither prove nor disprove Euclid's Fifth Axiom on the basis of the other axioms of Euclidean Geometry (Non-Euclidean Geometry).

Example. I can prove that one cannot prove nor disprove the axiom of choice on the basis of the eight axioms (ZF) of set theory given by Zermelo–Frankel.

Example. I can prove that one can neither prove nor disprove the continuum hypothesis on the basis of the axioms of set theory known as ZFC (Zermelo–Frankel together with the axiom of choice). This is the result of Gödel–Cohen.

These results are important insofar as they emphasize the fact that we cannot know beforehand what kind of thing is provable on the basis of some other kind of thing. Take, for example, any unsolved problem of mathematics—Fermat's last theorem, if you will. Its current status is that it has neither been proved nor disproved. But what is more, it is not *a priori* the case that it is either provable or disprovable on the basis of the current axioms of arithmetic. Given any axiom system, there exist statements P meaningful in the system that are true but which cannot be proved in the system.

As the Vermonter might say: "I know I'm some place. But I can't say how I got here." This is an informal statement of Gödel's Incompleteness Theorem, and delimits what can be done along axiomatic lines.

16. Problem P has an answer. I have a method (or an algorithm) for obtaining the answer and it has yielded A. The only way I have of checking that A is the answer is to work the method over again.

This situation seems to suggest that insofar as I might make the same error over and over again, a putative answer A is established only with a certain degree of probability.

Example. Find the sum of the integers 168304, 29, 840007, 77829, 105302, 314159, 7, 89, 890 and 38507. Method: working with pencil and paper, I place these integers in a column and I add, proceeding from bottom to top, as I was instructed in grade school. I may check my answer by doing this several times until I feel satisfied that I have obtained the right answer. Or I may adopt a slightly different algorithm by way of a check (for example, I may add from top to bottom). But what if I have consistently made an error with my elementary sums (for example, $6 + 8 = 12$)?

If my problem involves a mathematical proof, I may go over the mathematical reasoning step by step. I may do this once, twice, three times. I may then offer it for the verification of the mathematical community. The community may verify it. It may find ways other than mine for finding the answer. If another way is found yielding the same answer, then my sense of security is increased. I, or the community, may have insight as to why my answer is a reasonable one. This, too, increases my feelings of security.

If I am a hardware engineer, there are numerous devices and strategies that I may employ to increase the *computer system reliability*. I may put in parity checks, or error-correcting codes. I may build in two independent circuits that do the same job and then check one against the other. This is using *redundant hardware*. If I am a computer programmer, I may use *redundant software* to the same end.

17. P has an answer. I have an algorithm for finding the answer. I have written a computer program to implement the algorithm. I do not know whether my program is a correct realization of the algorithm.

This is a constant worry. Nonetheless, taking heart as I usually do, I run the program. Then I may be able to check whether the output is an answer.

This concern leads to the problem of *program verification* and to the problem of what constitutes such verification.

A programmer is confronted with a problem. In his mind's eye, he is able to see through to a computer algorithm and to a computer program. In some instances he puts this down on paper as a flow chart or as a set of schemata. More than likely, he neglects a good part of this formalization and simply works along in the programming language, improving and testing parts of it as he goes along. At the end, he has a "finished" product. A Q.E.F. ("which was to have been done") can now be attached to it as a seal of approval. But no programmer (or team of programmers) is sufficiently sanguine that he will want to attach this high-flown phrase to his work. He knows that all programs are works in progress; that inevitably he will have to correct bugs, to make modifications and improvements, even though his work has been released as a commercial product.

At any rate, here, at last, is a program. This program can be proved or disproved in the logical sense. This theoretical possibility has been known for some time, but the technology of program verification is still at an early stage of development, and very few programs of any substance have been "proved" in the sense of logic. Speaking pragmatically, there are numerous reasons why this is not possible at the level of 100% security. In order to prove that the program you have written makes the machine do precisely what you want it to do, you have to have a formalization of two things: (1) what the machine does under all circumstances; (2) what output you want for a given input. Now there are no *complete* formalizations of what a prototype machine does; the circuitry may be full of what designers call "don't cares" in which the circuitry itself is its only formalization. And there are no certifications that the copy of the prototype that is sitting on your desk does precisely what the formalization (assuming it exists) says it does.

Secondly, you think you know, in a precise mathematical sense, just what output you want for a given input. But this precision may be illusory.

Example. Given two positive integers m and n, I would like to produce their greatest common divisor. What could be clearer: input: m, n ($m > 0$, $n > 0$) output: g.c.d. (m, n). But the exigencies of the machine force a reformulation. First, a minor point: how are the numbers specified—in binary, in octal, in decimal? Okay, say decimal. In what shape are the numbers—fixed point, or floating point, or something else? Hmmm? Will floating point cause problems? Let's say, then, fixed point. Review the formal input statement: for *all* m, n. Now we clearly cannot provide for the input of *all* integers; we have to restrict the size. To what size: $m, n < 10^{15}$? Well, this may depend on how I have formulated my g.c.d. algorithm and how arithmetic is handled by my machine. So I had better have the formalization of the machine in front of me.

Though the techniques of program verification are interesting theoretically, and may lead to the development of significant new mathematics, their role in real world programming is moot.

In a practical context, this problem is discussed under the name of *software quality management* [C1].

18. *P* has an answer. I have an algorithm for finding it. I have written a computer program for my algorithm and I think I've proved that the program is correct.

This is a rare and felicitous state of affairs! But the worm of doubt begins to gnaw at the vitals of those who are pessimistically inclined. I do not know whether the proof is correct. The proof is full of hundreds of rinky-dink and niggling details—far more than the program itself. Thus, I begin to wonder whether some of the dirty work of program verification might itself be carried out by a machine. And indeed, within a limited scope, it can (theoretically).

Finally, I have no guarantee that when the button is pushed, the machine will function as it is "supposed" to function, assuming that I know how it is "supposed" to function. The laws of nature might have been abrogated momentarily just to annoy me.

At this stage, I begin to wonder whether I shouldn't throw in the towel.

19. *P* has an answer. I think the algorithm *A* that I have devised will give me the answer, but I am not sure.

This is not an infrequent state in numerical analysis. (Note that I am talking about an algorithm and not the program.) However, one writes a program for *A* and runs it. There may then be some way of telling whether the answer provided by A is *not unreasonable*. Under certain circumstances, we may adopt the policy that a not unreasonable answer is an acceptable answer. Presumably the algorithm A is not just a random affair plucked down from the heavens, but is based upon insight and experience.

Most programs in numerical analysis do not output an estimate of the possible error in the computed answer. If they do, such an estimate has the character of a rule of thumb. To appreciate some of the difficulties involved in obtaining an error estimate which is both realistic and has the character of a mathematical theorem, see [S2] where the numerical solution of ordinary differential equations is treated.

Here is a variation of considerable significance.

20. *P* is a problem derived from a mathematical model of a physical situation. The laws of physics together with experience in the field suggest that *P* has a solution. However, I have not been able to prove—in the mathematical sense—that it does. *A* is an algorithm I have derived for obtaining the answer (assuming it has an answer), etc. etc.

The contrast between knowing merely that there is an answer and that a specific algorithm will provide an answer is profound. It underlies the difference between two types of mathematics, existential and algorithmic (or, as P. Henrici calls them, dialectic and algorithmic). Mathematical existentialism (no relation to the philosophical existentialism of Sartre and others) is the program of adducing the existence of certain mathematical objects, independently of whether or not the line of reasoning leads to the real-world display of the objects. Algorithmic mathematics, on the contrary, stresses our ability to come up with algorithms and computer programs for doing the job. There are numerous places where both types of argument exist. If an existential argument is given, the question may then arise: does there exist an algorithmic construction? This inquiry leads to notions of *computability* [A1], [D1], [M1].

For a different line of thought, consider a physicist or engineer who may be interested in creating a mathematical model of a physical situation. He devises a set of equations. This, in itself, does not give him numerical answers. He devises an algorithm. This still does not give him answers. He then writes a program to implement the algorithm. He only gets answers when he runs the program on a computer.

He may not know whether the equations are a proper model. He may not know (in the logical sense) whether the program is a correct implementation of the algorithm.

Yet, despite this fog of uncertainty, common enough in the trade, the computer with the program sitting in it may be regarded *in its entirety* as an analog device for modelling the original physical situation, and its quality as such may be so judged.

21. *P* has an answer. I do not yet have a satisfactory algorithm for finding it.

This is what motivates a good deal of research and experimentation in numerical analysis.

Example. We do not yet have a satisfactory algorithm for the solution of the full Navier–Stokes equations of fluid theory.

The word 'satisfactory' may contain judgements that are not strictly mathematical. Why do I want the algorithm? How much am I willing to pay for running the program? How long am I willing to wait before the algorithm outputs? The famous example is that if I want to use the equations of fluid flow to predict Wednesday's weather, I don't want to wait until Thursday to get the answer.

22. I can prove that there are some mathematical "things" that I cannot compute.

This says that the mathematical universe is not synonymous with the computable universe. There is a barrier that separates what is computable from what is not, and the barrier is vague. One would like to think that whatever is useful is, in some sense, computable. This is far from clear. We should like to say, for example, that if a mathematical problem has to be solved for purposes of physics or technology, its answer must be computable.

This seems reasonable. Progress in our understanding of the physical universe has become synonymous with understanding through mathematics: the formulation and the successful description of mathematical models. At the bottom line—as Richard Feynman has emphasized—is the necessity of arriving at numbers through computers and computation. But this seems unreasonable, since the digital computer is only one type of real-world mechanism or process out of many. Why should it be the case that all physical processes be simulatable by digital devices? If this were the case, we would have a very significant reduction of the complexity of the universe and, simultaneously, a very significant limitation of the potentialities of the universe.

Example. I can prove that there are numbers that I cannot compute. To elaborate, a real number x $(0 < x < 1)$ is *computable* if there is a program in some idealized computer language, operating on some idealized computer with infinite storage—say, a Turing machine—such that upon the input $i = 1, 2, \ldots$ the program will output x_i, the ith decimal digit of x. Now the proof that there are noncomputable numbers is based upon the simple fact that computer programs are denumerable whereas the number of reals is nondenumerable [D1], [M1].

This proof is existential in quality, and while I am able to exhibit many computable numbers ($15/17$, e, π, πe), I am not able to exhibit a *noncomputable number*. How, indeed, can I exhibit one in its individuality, if such an exhibition must reveal the individual decimal digits and this would be done by means of a program?

For a discussion as to whether Gödel's work implies that the brain is better than a computer, see [H2], [W1].

23. I do not know the answer to *P*. But *P* contains a parameter in its statement and *P*(*n*) is a special case of *P*. I know the answer to *P*(1), *P*(2), *P*(3), . . . , *P*(*N*). If I knew the answer to *P*(*n*) for all positive *n*, I would know the answer to *P*.

This state is of *very* frequent occurrence. As examples, one may cite some of the famous unsolved problems of mathematics: Fermat's "Last Theorem," the Riemann Hypothesis, the distribution of the digits of the number π. In the last problem, let $P(n)$ be the statement that n consecutive 7's occur in the decimal expansion of π. Let P be the statement that an arbitrarily large number of consecutive 7's occur. Computations show that $P(1)$, $P(2)$, $P(3)$, $P(4)$ are true. The truth of P is not known.

I would guess (and I suppose that most mathematicians would guess) that P is true. This guess would be based on a feeling that the digits of π are totally mixed up, and any particular combination occurs infinitely often and in the right proportion.

Whence comes this feeling or intuition, considering that no mathematician has yet been able to extend this intuition into a formal proof?

The possibility of extrapolating from $P(1), P(2), \ldots, P(N)$ to P exists, but is treacherous. As an example, let $P(n)$ be the statement "the quantity $\sqrt{1141n^2+1}$ is not an integer." Now, $P(n)$ is true for $n = 1, 2, 3, \ldots, N = 30693385322765657197397207$. $P(n)$ is false for $N + 1$ (see [D3]).

Every unsolved problem is surrounded by a nexus of partial solutions and other relevant information. A good deal of research lives on the hope that a partial solution may be parlayed into a full solution.

24. P is one of a wide and important class of problems $\{P(\alpha,\beta,\gamma \ldots)$ indexed by parameters $\alpha,\beta,\gamma \ldots\}$, all of which have answers. I have a general algorithm for finding the answer upon input of any specific $\alpha,\beta, \gamma \ldots$.

Example. I have an algorithm for finding the sum of any number of integers.

Example. I have an algorithm for finding the complex solution to the quadratic equation $x^2 + \alpha x + \beta = 0$. On the basis of this algorithm, I can decide certain things about the solution (e.g., which solutions are real, etc.).

Example. I have an algorithm for deciding whether a system of m linear equations in n unknowns does or does not have a solution.

Example. There is a general algorithm (Tarski) for deciding whether statements in elementary geometry are true or false.

Notice that although these algorithms are constructive (or they wouldn't be algorithms), nothing is said about how long it would take a real-world machine to execute them. Until this can be arrived at in some way, the algorithm, paradoxically, retains an existential quality. In the last two examples cited, the computation times are excessively long, and I doubt whether programs have ever been attempted.

In each case, we are dealing with a *decision problem*, and in each of these cases the decision problem has been solved positively. In a certain sense, then, we have been able to "*automate out*" the problem, although the problem of whether we have effectively automated out the problem hinges on the running time. It would not do to have an algorithm for addition in which the sum of two ten digit numbers takes three hours to obtain. Even today, the problem of finding more and more efficient algorithms for *addition*, vis-a-vis parallel and vector machines, is wide open.

There are many, many wide and interesting classes of problems which can, in principle, be automated out. Among them are most of the problems that are given as drill in mathematical texts, up through, say, sophomore college level. This possibility poses serious questions for mathematical education. If a problem can effectively be automated out, in what way and to what extent should we study it and drill on it?

The question occurs with small children learning arithmetic, and it occurs equally well with college science majors learning about the stability of systems of ordinary differential equations. It is a question that has not yet been faced squarely.

25. I have an algorithm for solving a certain class of problem. Based on this algorithm, this class of problem has been automated out. However, there are theories of computable analysis, within which these algorithms are noncomputable.

The examples below occur in "Computable Analysis" by O. Aberth.

Example. For any fixed number b, there is no effective method of deciding for any number x whether or not any of the following are true: $x = b$, $x < b$, $x \leqslant b$, $x > b$, $x \geqslant b$.

Example. There is no effective method of determining whether a monic polynomial of degree greater than 2 has multiple zeros.

What, therefore, does it mean to be constructible, to have an algorithm? There are probably an unlimited number of senses in which the term might be used. Included among them is constructibility in the sense of the Intuitionistic School of mathematical philosophy.

As regards the examples just given, every practical numerical analyst knows that it may be an unstabilizing process to base a computer decision on whether $x = b$, when x and b are given approximately. Many programs in numerical analysis will function properly only when $x = b$ is interpreted with fuzzy leeway.

If an algorithm is not effective in the technical senses of Aberth or others, it may be evidence of the presence of numerical instabilities.

26. *P* is one of a general class of problems $P(\alpha,\beta,\gamma,\ldots)$ all of which have answers. I can prove that there is no general algorithm (of a certain class of algorithms) which gives the answer upon inputting the specific $\alpha,\beta,\gamma,\ldots$.

Example. There is no general solution to the polynomial equation $z^n + \alpha z^{n-1} + \beta z^{n-2} + \cdots + \gamma = 0$ expressible in terms of α, β, $\ldots$, γ by means of a finite number of rational operations and the extraction of jth roots (Abel, Galois).

Example. (Hilbert's 10th Problem) Find an algorithm to test whether a Diophantine equation of any degree and in any number of unknowns does or does not have an integer solution. This problem has been proved to be unsolvable (M. Davis, Robinson, Putnam, Matiyasevitch) [D2].

Example. The elementary functions cannot all be integrated in elementary terms (Liouville). The definition of an elementary function is expressible in recursive terms:

(1) A constant function, the identity function, and the exponential function are all elementary.

(2) If f and g are elementary, then $f+g$, $f-g$, fg, f/g $(g \neq 0)$, $f(g)$, f^{-1} (inverse of f) are elementary.

(3) Every elementary function is obtainable from (1) and (2) applied a finite number of times, and all functions so obtained are elementary.

Then $\int (e^x/x)\,dx$ is not elementary, nor is $\int e^{x^2}\,dx$ and many such examples.

And so . . .

The description of the variety of states of knowledge should, I believe, be pursued in its own right. Insofar as it presents the mathematical scene from the point of view of what is knowable, it would be valuable to elaborate the individual comments and examples.

Through this point of view we gather glimpses of unsuspected states of knowledge. We observe how the distinguishable states of mathematical knowledge have proliferated in time. We see how each individual piece of mathematics, considered at a fixed point in time, set in its own context of contiguous knowledge, really represents a unique knowledge state. Each individual problem unsolved leaves us in a state of suspense. We recognize that there are certain landmark problems in the epistemological sense—problems that have added significantly to the previous historical categories of awareness. We perceive that these categories are widened in response to the pressures of the general principles of mathematical growth: abstraction, generalization, mechanization, etc. We catch glimpses of the difficulties of finding an adequate framework for the representation and the storage of mathematical knowledge. We catch glimpses of the difficulties surrounding the notion of mathematical proof, derivation, computation, existence, formalization, truth. We gather that it is not possible to formalize completely the notion of what makes mathematical sense or what is an answer to a problem. We infer that, paradoxical as it might seem, the degree of formalization that is possible is a source of current strength, while the degree to which formalization is not possible is a source of future growth.

All the concerns of the philosophy of mathematics are elucidated by considering the epistemological element.

By considering states of knowledge, by considering the passage of a problem restlessly from one state to another, we help restore the time element to mathematics. It enables us to distinguish personal time, computational time, time interior to mathematics, historical time, and to observe how these different times have molded material, changed mathematical objects and demands, altered meaning, altered contexts, altered possibility.

The languages of current mathematical formalisms have no room for the time element. In this view, mathematics might even be defined as the one science in which time is missing. The observer is wiped out; the observer without whom the whole enterprise is meaningless. (See a nice essay on time in mathematics in [N2].) This is a deficiency, and consideration of states of knowledge helps to compensate for it.

The compilation of this material has left me with one overwhelming impression, and I would like to put it forward as my main point.

In the year 450 B.C. or thereabouts, the later School of Pythagoras proved that $\sqrt{2}$ could not be the ratio of two integers. Now this piece of mathematics, discovered at the very beginnings of the deductive method, was paradoxical and deeply disturbing. On the one hand, the $\sqrt{2}$ exists as the diagonal of the unit square, and has palpable reality; on the other hand, it cannot be a fraction. It is something that exists and yet does not exist. Legend has it that to celebrate this discovery, whether in elation or in shock, or for mystic ritual, the Pythagorean Brotherhood sacrificed a hecatomb of oxen. One might have supposed that the derivation of this contradiction through the power of pure thought would have condemned the deductive method to a stillbirth. After all, the mathematics of the ancient Egyptians did not have deductive proof, nor did the mathematics of Babylonia and the Near East, nor that of India or of the Orient. But the argumentative Greeks persisted; and in persisting, introduced a new and vitally significant element into the subject.

The story of the crisis that the Pythagorean discovery induced is well known, as is the subsequent recognition of the need for and the possibility of reconciliation. The lines along which this reconciliation was attempted by the Greeks, the geometrization of arithmetic, the efforts of Eudoxus, leading slowly to the formalization and the construction of the real number system in the 19th century, are all spelled out completely in books on mathematical history.

What I should like to emphasize here is this: in the wake of this ancient crisis, in the leap of thought by which the crisis was overcome, mathematics became aware of itself. It became aware of its own processes, of their power and inherent limitations. It became aware, likewise, of how impossibility is tentative and may be overcome. ***With this self-consciousness mathematics ceased to be the unwitting creation of mankind.***

Paul Valéry, the French poet, has written that the consciousness of creation is what distinguishes the artist. The Greeks insisted that $\sqrt{2}$ is not the ratio of integers and stood by the consequences. In so doing, they turned mathematics into art and the mathematician into an artist.

If mathematics is an art, it is one of the humanities. Reuben Hersh has written, "It is distinguished from the other humanities by its science-like quality. Its conclusions are compelling, like the conclusions of natural science. It is fallible, correctible, and meaningful."

Approached with desire, its silent formulas speak. If mathematics is an art, and has aspects in common with other arts, it is liable to parallel aesthetic criticism. We may read how Friedrich Schiller pointed to the excesses of self-consciousness among certain poets [B3] and wonder, similarly, how much self-consciousness can be good for mathematics.

REFERENCES

[A1] Oliver Aberth, Computable Analysis, McGraw-Hill, New York, 1980.

[A2] Bruce W. Arden (ed.), What Can be Automated?, M.I.T. Press, Cambridge, MA, 1980, Chapter 4.

[B1] E. J. Barbeau, Euler subdues a very obstreperous series, Amer. Math. Monthly 86 (1979) 356–371.

[B2] Charles H. Bennett, On Random and hard-to-describe numbers, IBM Watson Research Center Research Report RC 7483, Yorktown Heights, 1979.

[B3] Isaiah Berlin, "Against the Current," Essays on the History of Ideas, Viking Press, New York, 1980, p. 287.

[B4] N. Bourbaki, The architecture of mathematics, Amer. Math. Monthly 57 (1950) 221–232.

[B5] Percy W. Bridgman, A physicist's second reaction to Mengenlehre, Scripta Mathematica 2 (1934).

[C1] J. D. Cooper and M. J. Fisher (eds.), Software Quality Management, Petrocelli, New York, 1979.

[D1] Martin Davis, Computability and Unsolvability, McGraw-Hill, New York, 1958.

[D2] Martin Davis and Reuben Hersh, Hilbert's 10th problem, Scientific American 229 (November, 1973) 84–91.

[D3] P. J. Davis, Are there coincidences in mathematics? Amer. Math. Monthly, 88 (1981) 311–320.

[D4] Philip J. Davis and Reuben Hersh, The Mathematical Experience, Birkhaüser-Boston, 1981.

[D5] Hubert L. Dreyfus, What Computers Can't Do, Harper and Row, New York, 1972.

[D6] ———, A framework for misrepresenting knowledge, Philosophical Perspectives in Artificial Intelligence, M. Ringle (ed.), Harvester Press, Brighton, Sussex, 1979, pp. 124–136.

[D7] E. W. Dykstra, A Discipline of Programming, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[G1] Michael W. Gleason, Belief and Technique, Libra Publishers, Roslyn Heights, NY, 1981.

[G2] I. Gohberg, P. Lancaster and L. Rodman, Matrix Polynomials, Academic Press, New York, 1982.

[H1] Paul Halmos, Does mathematics have elements?, Mathematics Intelligencer 3 (1981) 147–153.

[H2] Douglas Hofstadter, Gödel, Escher, and Bach, Harper and Row, New York, 1979.

[L1] Imre Lakatos, Proofs and Refutations, J. Worral and E. Zahar (eds.), Cambridge University Press, New York, 1976.

[M1] Marvin Minsky, Computation, Finite and Infinite Machines, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[M2] ———, A framework for representing knowledge, The Psychology of Computer Vision, P. H. Winston (ed.), McGraw-Hill, New York, 1975.
[N1] Allen Newell, The heuristic of George Pólya and its relation to Artificial Intelligence, Dept. Comp. Sci., Carnegie-Mellon, 1980.
[N2] Didier Nordon, Les mathematiques pures n'existent pas!, Actes Sud., Paris, 1981.
[P1] G. Pólya, Mathematical Discovery, Wiley, New York, 1962.
[R1] M. O. Rabin, Theoretical impediments to artificial intelligence, Information Processing 74 (1974).
[R2] Edwina L. Rissland, The structure of mathematical knowledge, Technical Report 472, M.I.T. Artificial Intelligence Laboratory, Cambridge, MA, 1978.
[R3] Edwina L. Rissland, Understanding understanding mathematics, Cognitive Science 2 (1978).
[R4] Moshe F. Rubenstein, Patterns of Problem Solving, Prentice-Hall, Englewood Cliffs, NJ, 1972.
[S1] A. H. Schoenfeld, Teaching mathematical problem skills, Dept. Math, University of Rochester, Rochester, 1979.
[S2] L. F. Shampine and H. A. Watts, Global error estimation for ordinary differential equations, ACM Transactions on Mathematical Software 2 (1976) 172–186.
[S3] L. J. Stockmeyer and A. K. Chandra, Intrinsically difficult problems, Scientific American 240 (May, 1979) 140–159.
[W1] Judson C. Webb, Mechanism, Mentalism, and Metamathematics, D. Reidel, Dordrecht, 1980.
[W2] L. Wittgenstein, Bemerkungen über der Grundlagen der Mathematik, Oxford, 1956.