Guest Editorial

C.A.R. Hoare

1994

1. "If we knew present position and momentum of every particle in the Universe, we could in principle predict their positions and momenta at any time in the future."

2. "If we knew a complete and consistent axiomatisation of set theory, then the whole of mathematical truth could in principle be deduced from it by mechanically checkable proof in the predicate calculus."

3. "All properties of a program and all consequences of executing it in any given environment can in principle be found out from the text of the program itself by means of purely deductive reasoning."

This is a collection of three highly influential principles. The first of them is a summary of the inspiration that has guided the work of natural scientists since the days of Newton. The second has inspired the development of modern logic and the foundations of mathematics since the days of Frege. And the third, which I wrote myself in 1969, has been the inspiration of my own research and that of other computing scientists for the last twenty five years.

But the principles are all in fact literally *impossible*! Heisenberg has refuted the first, Godel has refuted the second, and Cook has refuted the third. That is the fate of most scientific principles: they need to be adjusted in the light of further knowledge — knowledge that could only be gained in pursuit of the very principle that it refutes. Even so, the original principle retains its value as an approximation and as a succinct statement of the goals of an entire research discipline.

Another goal of scientific research is to produce results that may one day find useful application in engineering practice. Certainly, the results of

Newtonian mechanics were immediately applicable in increasing the accuracy and range of artillery fire. In the same way, one might hope that the results of research in logic might improve the productivity of mathematicians and the reliability of their proofs; and I have often expressed the hope that research in Computing Science might improve the reliability and quality of computer programs, or at least the languages in which they are expressed.

These hopes too have been falsified by events. Both mathematicians and programmers regularly achieve quite acceptable productivity and accuracy, even without the use or study of the theory which underlies the practice of their profession. Indeed, the direct introduction of strictly formal methods into their daily practice would be incredibly boring, cumbersome and counterproductive. It would be like trying to build a bridge on the basis of quantum-theoretic calculations — a totally quixotic endeavour. Like other extreme methods, formal proof must be reserved for extreme circumstances, where no other normal method would be adequate. Nevertheless, the existence of formality as an agreed court of final appeal is of great value, even if it is very expensive and very rarely invoked. It provides absolute criteria on which the more usual and less cumbersome procedures of the lower courts should be based. Even just the threat of appeal is a constant incentive to the meticulous observance of these normal procedures.

There is no reason to suppose that the bridge between the theory of computing and the practical methods suited for information engineering will be any shorter or easier to develop than in any other branch of science and engineering. Set theory provides a foundation for topology, which provides a general framework for analysis; on top of this, the differential and integral calculi have developed a wide range of symbolic transformations and solution methods for many classes of equations that arise in engineering practice. The main goal at all the higher levels is to replace deductive proofs by symbolic calculations, and preferably to replace these too by books and tables of standard results. This is the kind of knowledge that can then be incorporated in computerised aids for automation of part of the design engineer's task. Absence of this laborious development has seriously delayed practical application of computerised tools in information engineering. Most of the work still remains to be done.

The development of an applied mathematics of computing requires first an adequate understanding of some range of computer applications, and of the computing methods, algorithms and paradigms most appropriate for their implementation. Second, it requires deployment of a range of mathematical technologies, including not only the axiomatic and deductive methods of logic, but the mathematical modelling of denotational semantics, and algebraic theories based on equational presentations. But most of all, the researcher must be motivated by an intense desire to make the link between mathematics and the reality of information engineering. Although the motivation is towards application, the methods and concepts of computing are drawn primarily from the world of pure mathematics — a paradox which may explain some of the slow progress in our field.

In addition to this application-driven work, continued exploration of the foundations has an absolutely vital role in any scientific discipline. The most important results are those which reveal and build on the similarities and links between the various branches of the subject, and thereby illuminate the structure of the subject as a whole. It is only this understanding that enables specialists in the various branches to communicate and cooperate on significant projects requiring a combination of their techniques and skills. It is only this understanding that forms the basis of a rational and progressive education in the subject. And it is only this that counteracts the tendency of any new and poorly understood discipline to split itself into warring schools, each claiming and devoting exclusive attention to a single design paradigm, notational framework, semantic method, or axiomatic presentation. Such fragmentation is encouraged by research funding agencies, which insist on promises that a single solution can be found for all problems.

But most important of all is research that both provokes and then satisfies the universal human curiosity about how things work and why, no matter whether the things are heavenly bodies or quarks, nuclear power stations or computer programs, or even the properties of some interesting set of mathematical postulates. Investigation of applications often helps in this too: it suggests interesting concepts and conjectures; it provides new intuitions and insights, together with examples and counter-examples to test them; and finally it provokes new questions and often a new approach to their answers. In this way, mathematics has been constantly rejuvenated by its applications. It is often impossible to tell in advance whether the main immediate contribution of research will be pure or applied; but we are driven by the hope that, in the long run at least, it will be both.

There are many unpredictable accidental, historical, commercial and political reasons why the results of Computing Science have taken so long to find application in engineering practice. Such factors are outside the province of Science; and they must certainly remain outside the influence or control of an individual scientist, on pain of loss of scientific objectivity, judgment or even integrity. But contributions to the advancement of scientific knowledge and mathematical understanding are in themselves of constant and cumulative value. The scientist who does not find delight or at least consolation in this thought will find it hard to endure the rigours of research during the long delays before its more practical application.