Computer Science as a mathematical discipline

The rapid development of computer-related courses in schools and universities has been one of the most remarkable features of modern education. Few subjects have attracted such publicity in universities over the last decade as Computer Science, yet its essential substance has proved difficult to identify, and it has been given a wide variety of interpretations in practice. Outside universities, the focus has often appeared to be upon practical - often very **un**scientific - computing, rather than any associated science.

For those involved in teaching and research in Computer Science, the problem of improving the image of the subject is becoming ever more important. As public awareness and experience of computers grows, it is no longer possible to rely upon the vocational value of an education in computing skills to attract good minds to computer-related courses - it is essential to represent Computer Science as a discipline as coherent as any classical science, and in no way as ephemeral as the commercial and technical products of the "computer revolution".

Many have argued about the status of Computer Science as a *science*. We expect of a science a unifying concern; a framework within which integrated theories, principles and techniques can be consistently developed, and used to make the results of observation and experiment more comprehensible and predictable. It is very significant that a mature science encompasses both solutions and problems: indeed the fact that we can identify particular examples of fundamental unsolved problems of *physical science* is of itself an indication that we understand - or presume to understand - the nature of physical science.

Many influences have conspired to undermine the image of Computer Science as a science over the last 25 years. If indeed Computer Science were to be interpreted as "the science of computers" we might feel justified in dismissing its claims to be concerned with fundamental problems as profound as those characteristic of other sciences. If the products of commercial computing were in reality the comprehensive definitive solutions to the quintessential issues which they so often purport to be, if popular recreational software were truly representative of the horizons of those involved in studying applications of computers, there might be little future for Computer Science. The image of Computer Science is not improved by representing inadequate or inappropriate solutions to problems as universal and comprehensive panaceas. Paradoxically, those who would champion the cause of Computer Science as a science must emphasise not merely the development of solutions to problems inspired by computing, but the existence of fundamental characteristic unsolved problems.

It may be that Computer Science as a science is an unfashionable idea even amongst many of its professional exponents. In seeking parallels with other sciences, the relative artificiality of computer-related concerns superficially presents a problem. There has always been a strong prejudice towards the view that truly fundamental problems are - whatever it means - naturally occurring, or at any rate the products of divine rather than human intelligence. It will be appropriate to return to the discussion of such philosophical - nay verily theological - issues in due course, but enough to consider, in the first instance, how Computer Science is related to traditional mathematics, in which artificial worlds, and heretical models of the natural world, have often found a role.

In seeking to identify Computer Science as a mathematical discipline, it is helpful to look back at the work of pioneering figures in the subject, such as Backus, Minsky and McCarthy, who were involved about 1960 in developing the first applications of computers. The central concern in their research could be broadly described as *the representation and manipulation of data*: a concern which is both mathematical in nature, yet characteristic of Computer Science, and a convenient banner under which to bring together many subsequent developments in the subject.

Representation and manipulation of data is of course by no means a new theme in mathematics. The very simplest illustration of its importance is to be seen in the choice of Arabic rather than Roman notation for numerical calculation. The celebrated debate between Liebnitz and Newton over a notation for the calculus is another example. It should be recognised that the choice of a suitable notation is not the primary issue here; it is the choice of a data representation which properly reflects a conceptual model that leads in turn to appropriate notations. Identifying the key concepts which lie behind a computational problem is often intimately linked with finding an appropriate data representation, as is profoundly illustrated in the work of Gauss and Galois on algebraic numbers and polynomials, of Boole and Frege on propositional logic, and of Hilbert and Turing on constructive proof.

What is significantly different about the problems of data representation and manipulation presented by computer applications is the extraordinary variety of data to be represented, the diversity of representations and manipulations to be considered, and the formality with which these must be described. Such is the scale and complexity of automatic data processing that issues of program correctness, feasibility and efficiency are of primary importance. Characteristic of modern computer science is a need to model data at many different levels of abstraction, and to translate between representations which are suitable on the one hand for human interpretation, and on the other for mechanical processing.

The above discussion may be seen as depicting Computer Science as an extension of the mathematical theory of algorithms. The advent of computers has certainly brought new interest and vitality to this branch of mathematical research, as results such as Hendrik Lenstra's recent algorithm for integer factorisation illustrate. At the same time, Computer Science is essentially concerned with wider issues.

One of the most important concepts to emerge from Computer Science is that of the abstract machine: a computational model in which the machine code may comprise operations of a complex nature. Designing such abstract machines is not only - nor even principally - relevant for constructing real computer hardware; it is also implicit in "designing a programming language" and in "choosing data structures for a program". There is a close parallel between mathematical "problem solving", and computer programming, in that some problems have only *ad hoc* solutions, and others lie within the scope of a unifying mathematical theory. In Computer Science, there is a similar distinction between algorithmic problems whose solution is most conveniently represented in terms of a one-off abstract machine, and those which lie within a class of algorithmic problems for which a generic solution relative to an abstract machine model has been devised.

As in mathematics, it is the unifying theories which underlie classes of solutions to particular problems that give the subject its coherence and distinctive identity. The study of abstract machine models is a central concern of Computer Science, and is a fundamental point of connection between algorithms in practice and in theory. The theory of automata and formal languages, and its applications to the specification and analysis of the syntax of programming languages is perhaps the most significant illustration of this.

Unified descriptions of algorithms are of unprecedented importance in Computer Science, both because of the need to represent all data manipulation in terms of a common machine model (whence the study of compiling and information structures, and work on algorithms for parallel architectures and concurrent systems), and the need for "abstract machine models" for describing general purpose algorithms (whence the study of programming languages and paradigms). It is tempting to go further, and suggest that Computer Science has initiated an approach to algorithmic problems without precedent in mathematics, but this may be misleading. It is hard to imagine how an approach to the algorithmic problem

"given a quadratic equation with integer coefficients, what are all its solutions?"

which was based on the search for unifying principles could succeed without revisiting much classical number theory. Indeed, a study of Gauss's work on this algorithmic problem reveals that his solution is at one and the same time the foundation for the mathematical theory of quadratic forms and algebraic numbers, and the description of an abstract machine model within which all the subroutines required to solve integer quadratic equations can be programmed. Such links between computational problems and mathematical theories have been very important sources of new mathematics in the past, and will be equally important in the future.

The aspects of Computer Science discussed above are mainly motivated by the technical challenges of designing and programming computers. As the sophistication of computer hardware and software increases, and the range of applications of computers widens, the need to consider data representation and manipulation in totally novel ways has arisen. In contemporary applications of computing, the emphasis has moved towards data models of enormous complexity incorporating concepts at a much higher level of abstraction than was appropriate hitherto. The problem of reliably identifying features in a picture, which involves the elaboration of an abstract data model in some way equivalent to a conglomeration of coloured pixels, is one example of an outstanding unsolved problem which is arguably closely analogous to "solved" problems such as "parsing a string of characters" - though it is of course of far greater complexity. The data models which will have to be used for the specification of large software systems, for communicating sequential processes, for robotics, for VLSI architectures, and for representation of knowledge - and the design tools which must be developed to assist these specifications - will necessarily require considerable mathematical sophistication. Their development may also represent a change of emphasis: in modelling a data base for instance, the problem is not that of finding a suitable data representation for performing a single complex algorithmic task, but of integrating many different related - often simple - algorithmic actions. The contrast between a traditional calculator and contemporary spreadsheets illustrates the significant role which interaction plays in such applications.

It remains to examine the more controversial aspects of representing Computer Science as "the science of data representation and manipulation". It is easy to argue that the concept of data representation and manipulation is so vague as to be inappropriate for a characteristic theme. Although many research workers in Computer Science would discern such a unifying concern in the most well-established areas of the subject, they would be loath to presume that the challenges of Artificial Intelligence are of an essentially similar nature to those encountered in compiling, or conventional data processing. It is indeed difficult to imagine how it might be possible to find computational models which faithfully reflect cognitive aspects of human intelligence; on the other hand, it would have seemed altogether surprising a century ago that the notion of an algorithmic procedure could be so convincingly modelled simply by Turing machines. Identifying the characteristic nature, and classifying the unsolved problems, is a major philosophical problem for any science, and doubtless involves an act of faith.

William Kent, in his book "Data and Reality", sets out to examine some of the practical issues raised in modelling data within a data base. His analysis shows very clearly how attempts to use traditional mathematical models for a data base fail to address many fundamental problems, both technical and philosophical. In his concluding chapter, Kent explains how his involvement with questions of data representation have influenced his philosophical outlook on the nature of the "real world" for which Computer Science is sometimes supposed to have substituted artificial models. Many who have worked on extending the range of traditional applications of computers have been obliged to consider profound questions concerning human cognition. Many who have contemplated the problems of specifying and analysing systems of the complexity which have become commonplace in modern Computer Science have come to recognise the direct relevance of their research to parallel work on physical and biological systems. Audacious as it may seem, it is appropriate for Computer Science to stake its claim to fundamental characteristic research problems concerning such systems, for it may well be that the mathematical traditions from which Computer Science has developed offer the best propects for new insights in these classical research areas.