

Foreword

Turing's deep 1937 paper made it clear that Gödel's astonishing earlier results on arithmetic undecidability related in a very natural way to a class of computing automata, nonexistent at the time of Turing's paper, but destined to appear only a few years later, subsequently to proliferate as the ubiquitous stored-program computer of today. The appearance of computers, and the involvement of a large scientific community in elucidation of their properties and limitations, greatly enriched the line of thought opened by Turing. Turing's distinction between computational problems was rawly binary: some were solvable by algorithms, others not. Later work, of which an attractive part is elegantly developed in the present volume, refined this into a multiplicity of scales of computational difficulty, which is still developing as a fundamental theory of information and computation that plays much the same role in computer science that classical thermodynamics plays in physics: by defining the outer limits of the possible, it prevents designers of algorithms from trying to create computational structures which provably do not exist. It is not surprising that such a thermodynamics of information should be as rich in philosophical consequence as thermodynamics itself.

This quantitative theory of description and computation, or Computational Complexity Theory as it has come to be known, studies the various kinds of resources required to describe and execute a computational process. Its most striking conclusion is that there exist computations and classes of computations having innocent-seeming definitions but nevertheless requiring inordinate quantities of some computational resource. Resources for which results of this kind have been established include:

- (a) The mass of text required to describe an object;
- (b) The volume of intermediate data which a computational process would need to generate;
- (c) The time for which such a process will need to execute, either on a standard “serial” computer or on computational structures unrestricted in the degree of parallelism which they can employ.

Of these three resource classes, the first is relatively static, and pertains to the fundamental question of object describability; the others are dynamic since they relate to the resources required for a computation to execute. It is with the first kind of resource that this book is concerned. The crucial fact here is that there exist symbolic objects (i.e., texts) which are “algorithmically inexplicable,” i.e., cannot be specified by any text shorter than themselves. Since texts of this sort have the properties associated with the random sequences of classical probability theory, the theory of describability developed in Part II of the present work yields a very interesting new view of the notion of randomness.

The first part of the book prepares in a most elegant, even playful, style for what follows; and the text as a whole reflects its author’s wonderful enthusiasm for profundity and simplicity of thought in subject areas ranging over philosophy, computer technology, and mathematics.

J. T. Schwartz
Courant Institute
February, 1987