last chapter of the book deals with relativized complexity. The book ends with one of the most valuable parts: an index and bibliography of more than fifty pages.

I will illustrate the style of the book by considering some details that have bothered me over the past years in my work and teaching in the areas covered by the book.

One of the classic results on Turing machines that cannot be found anywhere in the literature is the result that a two-dimensional tape can be simulated on one-dimensional tapes with a constant factor overhead in space. This result, attributed to the GDR author Hemmerling, who wrote about this result in a technical report in 1979, can be found in the book. The authors were unaware of the fact that the result became folk wisdom in the West ten years earlier, but since that state of affairs has left no visible trace in the literature at all, they cannot be blamed for it. Still, by including it, the authors have become the recipients of the Dutch Windmill Tile I offered at the 1984 STOC in Washington D.C. for precisely such a reference.

Another troublesome spot in the traditional literature is the correct definition of the logarithmic space measure for the RAM. The authors provide the correct definition, but on equal footing with three other definitions, all of which are wrong from the perspective of the fundamental space invariance issue involved. The authors fail to stress the importance of providing the correct definition here.

The authors present a short list of NP-complete problems but not as a subject more prominent than closures of complexity classes under various formal language theory based operations, comparisons between various constrained types of Turing machine tapes, or fragments of abstract complexity theory. In fact their list contains twenty problems, the larger part from before 1975, and the entire presentation in the book fills about one page—mostly with remarks on the strengthening of special cases and references to the literature.

In concluding, I consider the book a valuable asset for any researcher in the area of complexity theory, but having the book in your institute library (just make sure it remains there) rather than on your private shelves will suffice for your everyday needs unless you are an encyclopedist yourself. The book should primarily be used as a source of results and references and not as a textbook or as course material.

PETER VAN EMDE BOAS

GREGORY J. CHAITIN. ***Algorithmic information theory.*** Cambridge tracts in theoretical computer science, no. 1. Cambridge University Press, Cambridge etc. 1987, xi + 175 pp.

In the focus of this book there is a kind of incompleteness result, presented in an idiosyncratic style. Another important goal of the book is the rewriting of the history of the field (in the author's words, "a smoothed-over story"), presenting the author as the sole inventor of its main concepts and results.

*Incompleteness.* Gödel's incompleteness theorem defines a universal recursively enumerable (r.e.) set $S$ and states that for every finitely axiomatized formal theory $T$ (assumed to be sound) there is a number $n$ for which the fact $n \notin S$ is true but not provable in $T$. This theorem does not exclude the possibility that a new axiom decides $S$ at an infinity of new places. An application of algorithmic information theory provides us with an explicit series of undecided problems about which our knowledge will forever be limited, essentially, by the size of our axiom system. Concerning this series, deduction can give us essentially no extra "information." The construction of such a series, as explicitly as possible, is the central problem of the book.

Let us fix an appropriate universal Turing machine $U$ (almost all of the usual constructions will do). Let $K(x)$ be the length of the shortest binary input causing $U$ to output the string $x$. The function $K(x)$ is noncomputable in a very strong sense. It converges to infinity: indeed, for each $k$ there are at most $2^k$ strings $x$ with $K(x) < k$. At the same time, as Kolmogorov's and Barzdin's theorems (Theorems 1.5 and 1.6 in Zvonkin and L. A. Levin's survey, *The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms*, ***Russian mathematical surveys***, vol. 25 no. 6 (1970), pp. 83–124) show, every partial recursive function that is a lower bound of $K(x)$ is bounded.

The above theorem implies that for each formal system $F$, there is a constant $k_F$ such that no statement of the form $K(x) > k_F$ is provable in $F$. (Even the constant $k_F$ can be bounded by the complexity of the formal system itself.) Chaitin, among others, explored in a ***Scientific American*** article (vol. 232 no. 5 (1975), pp. 47–52) some philosophical consequences of this amazing statement.

The problem of computing the complexity can be transformed into other problems to obtain explicit very undecidable functions. The following one was defined in a footnote of Levin's 1974 paper translated as *Laws of information conservation (nongrowth) and aspects of the foundation of probability theory*, ***Problems of information transmission***, vol. 10 no. 3 (for 1974, pub. 1976), pp. 206–210. Let $S$ be an appropriate universal r.e. set (almost all the usual constructions will do). Let $d_n$ be the number of those $x < 2^n$ contained in $S$. Then the binary representation of $d_n$ is a string of complexity $n$. It follows that no deduction from any theory of size $k$ can give us more than $k$ bits of $d_n$. (Levin's theory also encompasses processes more complicated than deduction.) It is also mentioned that Matiásévič's theorem on the Diophantine representation of r.e. sets lets us formulate the above series of problems easily in purely arithmetical terms.

A related example is given by Chaitin in *A theory of program size formally identical to information theory*, ***Journal of the Association for Computing Machinery***, vol. 22 (1975), pp. 329–340. For this, a universal Turing machine $T$ with self-delimiting input (no endmarker) is used as defined, for example, in the above Levin paper and in my 1974 paper translated as *On the symmetry of algorithmic information*, ***Soviet mathematics***, vol. 15 no. 5 (for 1974, pub. 1975), pp. 1477–1480. Let $\Omega$ be the halting probability of $T$ with a random (infinite) binary string as input. Chaitin proves that the string of the first $n$ binary digits of $\Omega$ has complexity nearly $n$. (It follows by a theorem of Schnorr that $\Omega$ is a *random* sequence in the sense of Martin-Löf.) The number $\Omega$ has some amusing properties and made it (via a writing of C. H. Bennett) to a Martin Gardner article in ***Scientific American***, vol. 241 no. 5 (1979), pp. 20–34.

The book under review gives a particularly explicit definition of the number $\Omega$. An exponential Diophantine equation $E(n)$ is given that has infinitely many solutions iff the $n$th digit of $\Omega$ is 1.

*The first half.* More than half of the 175 pages is taken up by the detailed description of two models of computation: register machines and pure Lisp, and by an encoding of the second to the first and the first, using a 1984 method of Jones and Matiásévič (LI 478(4)), into exponential Diophantine equations. Even some of the huge equations resulting from the composition of the two encodings are shown (they cover several pages), heralding the era of books written by computers and readable only by computers. These printouts serve hardly any didactic or scientific purpose, except to evoke some religious experience in the reader: "It is all on these four pages."

*History.* The main goal of the second part is the rewriting of the history of algorithmic information theory. I will recall this history in the next few paragraphs. A more complete account is given in my article *Randomness and probability—complexity of description* in the ***Encyclopedia of statistical sciences***, vol. 7, John Wiley & Sons, 1986, pp. 551–554.

Algorithmic information theory arose from some of the basic philosophical problems concerning *randomness*, *inductive inference*, and *information*. The strong incompleteness theorem standing in the focus of the present book is only one of the applications.

The problem of randomness is based on a paradox well known to the founders of probability theory. If somebody shows one hundred "heads" saying that this is the record of her coin-tossing experiment then we will doubt she says the truth. This is despite the fact that the all-head outcome is just as probable as any other outcome. The continuous development of algorithmic information theory started in the 1920's with R. von Mises's definition of an infinite binary sequence as random (a "Kollektiv") if the relative frequencies converge in any subsequence selected according to some (non-anticipating) rule (the notion of a "rule" was left undefined, see von Mises, ***Mathematical theory of probability and statistics***, edited by H. Geiringer, Academic Press, 1964). Wald and others, including Church, made von Mises's definition sound by restricting the rules to a countable set, e.g. to the set of recursive functions. Ville proved that the randomness tests arising from the von Mises selection rules can still violate the law of iterated logarithm, for example. He proposed that the set of tests be a countable set of arbitrary payoff functions (martingales) on the set of infinite sequences.

A satisfactory solution came finally along a line proposed by Laplace in ***A philosophical essay on probabilities***. Laplace stated that the proportion of "regular" sequences (whatever "regular" means) to the rest is small. Kolmogorov proved this by defining "regular" as "having a short description" in *Three approaches to the quantitative definition of information* (XXXIV 318). A description of an object means here a program causing an appropriate universal computer (almost any of the usual ones will do) to output the object. Kolmogorov also proposed descriptional complexity as the proper definition of *individual information content*. His definition of complexity was preceded by that of Solomonoff who, for doing *inductive inference* by Occam's razor, introduced the notion of *a priori probability* and descriptional complexity in *A formal theory of inductive inference, Part I*, ***Information and control***, vol. 7 (1964), pp. 1–22. Both Kolmogorov and Solomonoff proved, independently, the so-called invariance theorem stating that descriptional complexity is (to within an additive constant) independent of the choice of the universal computer interpreting the descriptions.

P. Martin-Löf defined randomness in 1966 for infinite sequences (XL 450). (His concept is the synthesis of Ville and Church, as noted in Schnorr's book, ***Zufälligkeit und Wahrscheinlichkeit***, Lecture notes in mathematics, vol. 218, Springer-Verlag, 1971.) He constructed a universal randomness test, and pointed out the close connection between the randomness of an infinite sequence and the complexity of its initial segments. Chaitin also introduced the notion of program length complexity in his 1966 paper, *On the length of programs for computing finite binary sequences* (XL 518(13)). The crucial invariance theorem appeared in the second part of the paper (XL 518(14)).

The first broad survey paper on the subject, with several new notions and results, was the paper of Zvonkin and Levin cited above. Both Levin and Chaitin were fascinated by the incompleteness results, but they reacted quite differently. Chaitin continues to give popular expositions, such as the book under review. Levin has been interested in technical generalizations. Part of these concerned processes other than deduction that can be permitted in gaining information. In his 1974 paper cited above and in *Randomness conservation inequalities; information and independence in mathematical theories*, ***Information and control***, vol. 61 (1984), pp. 15–37, he proved a so-called law of information conservation, stating that the amount of information in a sequence $\alpha$ about a sequence $\beta$ cannot be significantly increased by algorithmic processing of $\alpha$, even when using random number generators. These results are technically difficult: they depend on a careful choice of some intermediate notions. A framework is found in which the information conservation results are a natural consequence of the most general properties of randomness. In the later work, the information conservation theorem is applied to obtain novel models for intuitionistic mathematics.

The technical reason for the difference between Levin's and Chaitin's approaches can be found in the opposite conclusions they draw from the strong incompleteness theorems. Chaitin says, optimistically, in the present book: "Perhaps number theory should be pursued more openly in the spirit of experimental science! To prove more, one must sometimes assume more" (p. 160). Levin's information conservation theorem refers to many non-deductive processes and supports a pessimistic thesis, saying that any sequence that may arise in nature contains only a finite amount of information about any sequence defined mathematically. In his 1974 paper, Levin says: "Our thesis contradicts the conviction of some mathematicians that the truth of any true proposition can be established in the course of the development of science by means of informal methods (it is impossible to do so by formal methods, due to Gödel's theorem)."

The above Zvonkin–Levin paper defined a priori probability as a maximal (to within a multiplicative constant) semicomputable measure. With the help of this notion and a modified complexity definition, Levin gave a simple characterization of random sequences by the behavior of the complexity of their initial segments in *On the notion of random sequence*, ***Soviet mathematics***, vol. 14 no. 5 (for 1973, pub. 1974), pp. 1413–1416; see also my paper in ***Zeitschrift für mathematische Logik und Grundlagen der Mathematik***, vol. 26 (1980), pp. 385–394. Part of this characterization is found independently by Schnorr in *Process complexity and effective random tests*, ***Journal of computer and system sciences***, vol. 7 (1973), pp. 376–388. With the exception of Martin-Löf's first paper, the above history is ignored by Chaitin: he refers to random sequences as "Martin-Löf/Solovay/Chaitin random" (p. 138).

In the 1974 papers by Levin and me, Levin's technically superior version of descriptional complexity, the so-called self-delimiting complexity is introduced, and several of its information-theoretical properties are proved. Chaitin found part of these results independently and published them a year later in the 1975 paper mentioned above. Chaitin presents himself as the sole inventor of description complexity as well as its self-delimiting version.

Comparing the above history with Chaitin's book one concludes that the refereeing process of the new series (Cambridge tracts in theoretical computer science), of which the present book is the first title, must have been lax if the absence of references even to Kolmogorov went unnoticed.

*The second half*. The introduction to program-size complexity is as usable as some of the author's earlier expository writings on the subject (except that references to the other researchers are now completely missing). Since a programming language (pure Lisp) was fully defined in the first part, some of the additive constants in the formulas that refer to the length of some programs are now given explicitly.

The exposition of randomness and incompleteness suffers from the author's narrow focus on the number $\Omega$ defined above. It is easy to show by using standard techniques that $\Omega$ is random. Several general properties of random sequences (e.g. unpredictability), however, are introduced here as just the properties of the number $\Omega$. Also, many of the results would become much more transparent if the relation between randomness tests and general semicomputable measures was clarified, as done in Levin's 1973 work.

Theorem R8 gives an exponential Diophantine equation $E(n)$ that has infinitely many solutions iff the $n$th bit of $\Omega$ is 1. This follows immediately from the fact that $\Omega$ is the limit of a monotonic computable sequence and well-known earlier theorems. A random sequence was given by a two-quantifier formula in the Zvonkin–Levin paper cited above.

Theorem LB states that no formal theory can derive arbitrarily large lower bounds to the complexity of strings. As said above, this is an easy consequence of Kolmogorov's and Barzdin's theorems given in the Zvonkin–Levin paper. A more general statement is contained in Levin's Theorem 3 in *On the principle of conservation of information in intuitionistic mathematics*, ***Soviet mathematics***, vol. 17 (1976), pp. 601–605.

The rest of the book culminates in Theorem C that says that a theory can deduce by at most an additive constant more (scattered) bits of $\Omega$ than the complexity of its axioms. This is an immediate consequence of Levin's 1973 theorem on the relation between randomness tests and semicomputable measures.

*Conclusions.* Chaitin has done more than others to popularize some aspects of algorithmic information theory. The benefits of this activity are offset by his somewhat narrow interests ($\Omega$) and the way he ascribes all major achievements of the theory to himself.

PETER GACS

BARRY E. JACOBS. ***Applied database logic I: fundamental database issues.*** Prentice-Hall, Englewood Cliffs, N.J., 1985, xviii + 334 pp.

This book is an attempt to use some ideas from logic to solve a family of problems in the theory of databases. The problems all concern fundamental issues regarding the *representation* of *facts* (or perhaps non-factual states of affairs) about the world. Indeed, a database is a collection of such representations. The trouble is that there can be many superficially distinct ways of representing the same facts. One would like to be able to say, in a precise way, just what it means for this to be so, i.e., for two distinct databases to be different "views" of the same set of facts.

This problem has many practical projections in database theory and management. Suppose, for example, that a system has two views of the same set of facts, and that a user updates one view in some way, by modifying one of the databases. Under what conditions can we determine how to modify the other database view of the same set of facts, so that the two remain different views of the same facts? Or suppose a company wants to change the way it represents data, going from, say, a hierarchical database to a relational database. Under what conditions can one write a program to transform automatically hierarchical representations to relational representations of the same data?

These are the sorts of problems this book is trying to tackle. The logical tool used is the familiar model-theoretic notion of an *interpretation* of one theory in another. The basic idea is that two databases are different views of the world if each can be interpreted in the other. To make this idea work, however, the notion of an interpretation has to be extended to a richer class of languages than usual, due to the higher-order nature of many database constructs.

The basic notions are those of *database scheme S*, and a *typing T* for a database scheme $S$. A database scheme is roughly what we might call a signature for a language. A typing of a scheme basically tells you what types of objects are associated with the signature.

Start off with a set of what are called "names," some 0-order, some "higher-order." With each higher-order name $R$, a database scheme associates a unique finite sequence $(R_1, \ldots, R_n)$ of names, either higher-order, 0-order, or mixed. Intuitively, this association, which is called a "rule" of the database scheme $S$, determines the type of arguments of an object of type $R$. Different sorts of databases are determined by placing various restrictions on these schemes. For example, in a relational database scheme, only 0-order names can appear in the sequence $(R_1, \ldots, R_n)$ associated with each $R$. A network database scheme is one where no name $R$ appears in the list $(R_1, \ldots, R_n)$ associated with itself.

Next come the definitions of a language $L$ for a database scheme $S$, and then a database structure $\mathbf{M}$ for $L$. Once these definitions are in place, the notion of a database view is presented as a theory in the given language. The notion of consequence of a database view used in a semantic one: true in all models of the database view. Different database views can have different schemes, typings, languages, and theories. The key notion is that of an interpretation of one database view in another. Given what has come earlier, it is what one would expect. Databases are then construed as equivalence classes of database views under the relation of mutual interpretability.

One of the interesting features of these database languages and theories is that while the relation symbols are higher-order, they do not fit into the traditional hierarchy of types: first-order,