Finite model theory studies the expressive power of logical languages over collections of finite structures. Over the past few decades, deep connections have emerged between finite model theory and various areas in combinatorics and computer science, including complexity theory, database theory, formal language theory, and the theory of random graphs. Indeed, the recent surge of interest in finite model theory has, to a large extent, been fueled by the effort to develop tools to address problems that have arisen through pursuit of these connections. Leonid Libkin's new book, *The Elements of Finite Model Theory*, is a beautiful introduction to these developments, with special emphasis on topics of interest in computer science. The exposition is lucid throughout, with ample motivation for the major technical developments, and generous discussion of examples to illustrate the import of central concepts. The book is self-contained and makes an ideal text for self-study or for a "topics in logic course" aimed at undergraduate or graduate students in computer science or mathematics. Indeed, one of us just used the book as text for such a course this past semester, and we can report that the entire audience (two-thirds of whom were undergraduates), which included linguists and philosophers, as well as computer scientists and mathematicians, responded enthusiastically to the text. A noteworthy feature of the book from this perspective is its wealth of exercises: each of Chapters 3-13, which form the core of the book's technical development of finite model theory, ends with anywhere from a dozen to three dozen well-chosen problems. Some provide students the opportunity to test rather directly their mastery of the chapter's central ideas, while others engage additional topics, or present more challenging (in some cases, potential) applications of the newly acquired techniques. The author provides references for solutions of most of the less straightforward exercises, which proved valuable in piquing students' interest in exploring the literature.

The book begins with a discussion of examples from database theory, complexity theory, and formal language theory, which motivate the subject. Each example emphasizes results which connect logical definability and central notions from the given area. In database theory, for example, the queries expressible in the relational calculus are exactly those definable in first-order logic. Moreover, various queries which arise naturally in the database setting, such as reachability or cardinality comparison, can be shown to elude expression in first-order logic. In order to formulate such queries, one must enrich the relational calculus; various such enrichments correspond to natural extensions of first-order logic. A large part of the book is devoted to exploring powerful techniques for establishing that given queries are *inexpressible* in a given logic, and in exploring the logical and computational properties of several extensions of first-order logic. In particular, Chapters 3 and 4 focus on basic tools for analyzing the expressive power of first-order logic.

Chapter Three begins with classic examples which illustrate the use of the Compactness Theorem in establishing inexpressibility results for first-order logic over arbitrary structures, and then briefly explores the possibilities and limitations of this technique in finite model theory. Though it appears that the Compactness Theorem may be applied to study first-order definability over finite structures via the use of pseudo-finite structures (a structure is pseudo-finite, if every first-order sentence it satisfies is true in some finite structure), the breadth of such application seems limited. The Chapter then introduces a fundamental technique for establishing the inexpressibility of queries in first-order logic over finite structures. This technique derives from the Ehrenfeucht-Fraïssé Theorem, which offers a combinatorial characterization of elementary equivalence of relational structures: relational structures A and B satisfy the same first-order sentences of quantifier rank at most n, just in case the "Duplicator" has a winning strategy for the n-round Ehrenfeucht-Fraïssé game played on A and B, that is, a method which maintains a partial isomorphism between pebble-induced substructures, one for each move. Now, in order to show that a given boolean query Q, is not first-order definable over finite structures, one need only exhibit for each n a pair of finite structures, G and H, with $G \in Q$, $H \notin Q$, and a winning strategy for the Duplicator in the n-round E-F-game played on G and H. The author gives lucid and detailed illustrations of this technique including an analysis of the expressive power of first-order logic on finite linear orders, and a proof that the property of being a tree is not first-order over finite graphs. The author invites the reader to apply the game technique to establish that various properties, *e.g.*, being a balanced binary tree, are not first-order definable. This effectively makes the point that the combinatorics of game arguments can be burdensome, and that more efficient techniques would be most welcome.

Chapter Four is devoted to exploring several such techniques for establishing inexpressibility results, based on notions of "locality" for queries. It offers a clear exposition of the notions of Hanf and Gaifman locality and provides an incisive proof of Hanf's Theorem, which states that every first-order definable query is Hanf local – it cannot distinguish between two structures which witness identical numbers of neighborhoods of a fixed radius r depending on the query, and it's corollary that every first-order query is Gaifman local – a similar notion applied within a single structure, in that elements correspondingly situated in isomorphic neighborhoods are indistinguishable. A particularly useful result of the author in collaboration with Dong and Wong is the "Bounded Number of Degrees Property." The BNDP says that the number of distinct degrees witnessed by elements in the output relation of a first-order query is a function only of the maximal degree witnessed by elements in the input relations on the original structure, and not of the number of elements themselves. This makes transparent the reason that an ordering (which witnesses n distinct degrees on a domain of size n) cannot be defined over a class of bounded-degree structures. For readers who had taken up the earlier challenge, it is particularly satisfying to see the ease with which the BNDP can be applied to show that there is no first-order test for being a balanced binary tree.

Until Chapter Five, no consideration has been given to the source of our finite models. But "real life" structures are often derived from situations in which there is an implicit ordering of the underlying domain, even though that ordering may no longer be known. The ability of a first-order formula to utilize an ordering on the domain of the model is a powerful notion which stretches the limits of definability. If a structure has a "built in" ordering, then the notion of locality degenerates into the trivial (since all elements of the domain are comparable and hence of distance one or less). But far worse, an arbitrary ordering slapped onto the domain would allow expression of queries which are not isomorphism invariant – a critical tenet of definability regardless of the logic. These considerations lead naturally to the study of order-invariant definable properties: first-order queries which utilize a total ordering of the domain in their input, but whose output doesn't depend on the particular selection of that ordering. It is relatively straightforward to show that the expressive power of order invariant FO is strictly greater than ordinary FO, by considering the query which determines if an atomic Boolean algebra has an even number of atoms. Most importantly, order invariant FO retains the fundamental concept of locality. The proof of this result seems inherently difficult, but stands as a demonstration that the notion of order invariance has great merit.

Chapter Six begins to investigate computational complexity, from the perspective of finite model theory. The Chapter focuses on the model-checking problem, that is, the satisfaction relation viewed as a computational task: given as input a finite relational structure A and a sentence S, determine whether A satisfies S. The asymptotic complexity of the satisfaction relation can be viewed in two ways, depending on which parameter varies: the model or the sentence. If the sentence is fixed and the model is variable, we are studying the "data complexity" problem which is a hallmark in the computational complexity of algorithms (a fixed algorithm applied to arbitrarily large finite data). On the other hand, if the model is fixed and the sentence varies, we obtain the so-called "expression complexity." These notions are of evident interest from the point of view of database theory and formal methods for verification (where the underlying finite structure is large but constant). Further analysis of data complexity reveals that first-order logic extended with numerical predicates can be calibrated via the computational capabilities of constant depth boolean circuits. This connection is exploited to prove the inexpressibility of the parity query in first-order logic with arbitrary numerical predicates as a corollary of the well-known result due to Furst, Saxe and Sipser, and to Ajtai, that there is no constant depth polynomial size family of circuits that computes parity. The Chapter also introduces parametric complexity in application to the model-checking problem, which is of special interest from the database perspective, where typically queries are very small compared to the databases against which they are evaluated. The author highlights the use of Hanf-locality to demonstrate Seese's result that the model-checking problem for first-order logic on structures of a given bounded degree is fixed-parameter linear time. The Chapter closes with a detailed discussion of the complexity of evaluating conjunctive queries, which are among the most common database queries.

Chapter Seven is primarily devoted to the now classical connections between monadic second-order logic (MSO) and automata theory. It is well known that quantifying over subsets is a more tractable form of second-order definability. This is clearly illustrated by the close connections with first-order types which allows for the use of game theoretic techniques in inexpressibility results. Specializing to formulas whose monadic second-order quantification is entirely existential or universal leads to interesting separation results involving graph problems. For example, using monadic quantification, connectivity can be defined universally but not existentially. With graph reachability the situation is more subtle. In the undirected case, reachability is both existentially and universally definable, while in the directed case it is universally but not existentially definable. By restricting attention to the vocabulary of strings (a linear ordering together with unary predicates), the classical connection with regular languages is obtained via the collapse of MSO to its existential (or universal) fragment. But this does not leave out first-order logic, because on strings it defines precisely the star-free fragment of regular languages. The collapse of MSO extends to the vocabulary of trees. An automata theoretic characterization is crucial to yielding the linear-time evaluation algorithm which rounds out this chapter.

In Chapter Eight, yet another approach to enhancing the expressive power of first-order logic is covered. Postulating a separate numeric domain, counting quantifiers are added in the context of infinitary formulas. Bijective games used to characterize this counting logic show that, in a certain sense, this augmentation obeys the same locality restrictions as the underlying first-order logic. The natural connection with constant-depth threshold gates in circuit complexity is presented, along with the corresponding order-invariant extension of the counting logic. This connects well with aggregate operators in database query languages, which finishes this chapter. Overall, this gives the reader a well rounded and concise view of both the power and limitations of counting quantifiers within the context of first-order logic.

Chapter Nine introduces the technique of encoding Turing machine computations as finite structures. Via this technique, sentences of a given logic may represent certain computational problems. The Chapter presents two fundamental applications of this technique: Trakhtenbrot's Theorem and Fagin's Theorem. A sentence is finitely satisfiable, if and only if, it is true in some finite structure. Trakhtenbrot's Theorem shows that the r.e.-complete problem of testing whether a Turing machine halts on the empty tape, is effectively reducible to the problem of testing whether a first-order sentence is finitely satisfiable. As a corollary, the collection of "finitely valid" first-order sentences is co-r.e.-complete, from which it follows that there is no complete proof procedure for finite validity. Fagin's Theorem shows that existential second order logic captures the complexity class NP. That is, every query which is $\exists$SO definable is in NP and every query in NP is definable in $\exists$SO. Indeed, we can pass effectively from existential second order problem specifications to polynomial bounded nondeterministic Turing machines which accept the problem thus specified, and vice-versa. This result initiated a burst of activity in "descriptive complexity theory," the study of logics that capture other complexity classes. Many results were achieved, especially for finite structures with "built-in" relations such as order (see below). It remains a fascinating open question whether there is a logic that captures any complexity class below NP (on unordered structures).

Chapter Ten concerns fixed-point logics – recursive first-order formulas that are able to express many natural problems in a computationally efficient manner. The most common are based on inductive definitions of relations that always reach a fixed-point, either in a monotone or inflationary semantics. Monotone inductive definitions always give rise to a relational operator which determines a least fixed-point (LFP), whereas inflationary inductive definitions reach a fixed-point determined by a non-decreasing sequence of relations (IFP). Recursive definitions that may not determine an inductive fixed-point fall into the class of partial fixed-points (PFP), though these may not be computationally tractable. Although it is not always possible to effectively test if a recursively defined formula is monotone, formulas in which the relational recursion variable appears positively are always monotone. In this context, it is straightforward to define queries such as transitive closure and reachability, previously inexpressible in FO. Furthermore, arithmetic can be defined over a successor function. Since recursive definitions can be nested, it is important to obtain normal forms for fixed-point logics. It is relatively easy to show that simultaneous recursion can be eliminated by using a larger arity relation. A careful analysis of the stage comparison theorem specialized to finite structures affords elegant proofs that IFP=LFP, and that nested application of least fixed-points can be reduced to a single application. The connection with complexity classes occurs over ordered structures: LFP captures Ptime, whereas PFP captures Pspace. These are substantial indications that fixed-point logics are the right intermediary between first-order and second-order logic. In fact, further syntactic restrictions on the form of inductive definitions yield fruitful connections with both database theory and complexity theory. In particular, controlling the application of negation and not allowing universal quantification yields Datalog, an important query language in database theory. On the other hand, limiting the use of inductions to performing transitive closure operations captures NLOG over ordered structures (non-deterministic logspace), and yields the celebrated result of Immerman-Szelepcsényi that NLOG is closed under complementation. The proof given here follows Immerman's original line of thought, by eliminating negation from transitive-closure logic. The chapter closes with a discussion of whether there is a logic for Ptime, a fascinating question that cannot be explained in the scope of this review for reasons of length.

Chapter Eleven gives a systematic treatment of finite variable logics developed from the point of view put forth by the authors of this review together with Anuj Dawar while he was a graduate student at The University of Pennsylvania (now at Cambridge University). Though it may be infinitary, a formula in this logic can only mention a finite number of variables. At first blush, it might seem that this very unnatural appearing syntactic restriction results in only a finite number of inequivalent formulas for each L_k, the set of formulas in k variables. But nothing could be further from the truth. Consider the sequence of formulas φ_n which say there is a path of length n between x and y.

$$\varphi_1(x,y) \equiv E(x,y);$$
$$\varphi_{n+1}(x,y) \equiv \exists z[E(x,z) \wedge \exists x(x = z \wedge \varphi_n(x,y)] \quad n > 1.$$

For every n, each φ_n is in L_3 since variables are continually being reused inside their own scope. Moreover, the infinitary disjunction of these formulas is an expansion of the induction defining the transitive closure in fixed-point logic.

This example is illustrative of the most important reason for studying finite variable logic: its ability to embed all the fixed-point logics mentioned in the previous chapter (i.e. all relations that are recursively defined by a first-order formula can be indefinitely expanded using a bounded number of variables). Not surprisingly, finite variable logic can be characterized by a pebble-game in which pebbles are reused (picked up and placed somewhere else). These games are fundamental to showing that k-variable types (even infinitary ones) are definable by k-variable first-order formulas (finitary ones). The type-equivalence relation this induces on k-tuples of a finite structure is itself definable in LFP. But perhaps the deepest insight concerns the inductive definability of a pre-order on k-tuples which respects this equivalence relation – the so-called "ordering of types." Finally, this ordering of the types can be used to demonstrate a rather significant connection between fixed-point logics and complexity theory: LFP=PFP if and only if Ptime=Pspace. What makes this result of particular importance is that it holds over unordered structures! Previous transfer theorems between logic and complexity required the artificial introduction of an ordering.

Chapter Twelve introduces an active area of investigation at the interface between logic and combinatorics: random graphs and zero-one laws. We may view the simple graphs with node set $\{1, \ldots, n\}$ as a probability space $G_{n,p}$ by placing an edge between nodes i and j with probability p (if $p = \frac{1}{2}$ this determines the uniform distribution). Fix $0 < p < 1$, and write $\mathsf{P}_n(\varphi)$ for the probability that φ is satisfied in $G_{n,p}$. A logic L is said to satisfy a limit law, if $\lim_{n\to\infty} \mathsf{P}_n(\varphi)$ exists, for all L-sentences φ, and a zero-one law, if this limit probability is always either 0 or 1. Though many natural logics clearly fail to have limit laws, for example, the counting logics discussed above, existential second-order logic, *etc.*, investigators from the late sixties through the eighties of the preceding century showed that first-order logic and its fixed-point extensions satisfy the zero-one law. A most satisfying explanation for this phenomenon was provided by Kolaitis and Vardi who established that the infintary logic $L^{\omega}_{\infty\omega}$ satisfies the zero-one law. This provides further evidence of the fundamental interest of finite variable logics (which appear, at first glance, to be rather unnatural) in finite model theory. The proof establishes an interesting reduction, namely, for every k, there is a single first-order sentence with k variables, the "k-extension principle," which axiomatizes the complete theory of $L^{k}_{\infty\omega}$ sentences whose limit probability is 1. It follows at once that the set of first-order sentences with limit probability 1 relative to $G_{n,p}$ is independent of the choice of p between 0 and 1, and that this set of sentences is decidable. The author notes that this set of "stochastically valid" sentences is an ω-categorical theory whose unique countable model (known as the *random graph*) is, up to isomorphism, the collection of hereditarily finite sets with the symmetric closure of the membership as its edge relation.

Chapter Thirteen examines an entirely different approach to finite models – embedding them into a fixed underlying infinite model with its own relations –

reflecting the fact that data elements are often taken from the "real world", such as R^1 (temporal information) or R^2 (spatial information). A succinct treatment of the issues surrounding first-order queries in this setting is provided. First-order queries can exhibit information about not only how elements relate in the finite structure, but also how they relate in the infinite structure. Moreover, quantification can be over the entire infinite natural domain, or restricted to the finite active domain. In the context of a first-order query whose output is interpreted over just the finite model, the questions to be answered are: whether access to the additional relations in the larger vocabulary matters; and if quantifying over additional elements in the larger domain makes a difference. To answer these questions, infinitary techniques are introduced in order to leverage the finitary techniques of the previous chapters.

One of these techniques is the Ramsey property, applied to queries with the active domain semantics. It is used to show that generic first-order queries – those whose results are independent of the embedding – can be rewritten in order-invariant FO over the original finite model's vocabulary. This result is entirely general, regardless of the embedding structure. On the other hand, for queries that actually depend on the particular embedding, the additional relations clearly do matter. In this situation, the question becomes whether quantification over the additional elements is essential. In this case, it turns out to be desirable to consider theories admitting quantifier elimination. The author then shows that for the real ordered field, queries expressed in the natural domain semantics for FO can be rewritten using active domain semantics. This natural-active collapse is not a general property. The case of the random graph is most illuminating – it also has a decidable theory and admits quantifier elimination, but FO over the natural domain collapses only to MSO over the active domain. Fascinating connections between these phenomena and classical model-theoretic notions, such as o-minimality and the finite cover property, are explored in exercises.

The chapter closes with one of the most popular ways to use these ideas: constraint databases in which first-order queries are used to define infinite subsets of the real line or plane, by allowing the free variables to range over the natural domain. Formulas defining regions in a finitary manner take the place of strictly finite relations. A nice argument is given showing that topological connectivity of regions so definable in R^3 is not first-order expressible.

The book concludes with a selection of further applications of finite model theory to classical decision problems, modal logic, and constraint satisfaction. Here, as elsewhere throughout the book, the author's lucid exposition of each topic invites the reader to explore further. All in all, the *Elements of Finite Model Theory* is a wonderful text that rewards careful study with a deep appreciation and understanding of one of the most compelling applications of logic to the study of information and computation.