

Book review of *Descriptive Complexity* by Neil Immerman (Graduate texts in Computer Science, Springer, 1999). Appeared in the **Bulletin of Symbolic Logic** vol. 7 no. 4 December, 2001 pages 525-527.

Overview

In a slim and neatly packaged volume of 250 pages, Neil Immerman has written a definitive reference on descriptive complexity, the field which attempts to elucidate the study of resource-bounded computation by describing the semantic complexity of an algorithm by its syntactic formulation in logic. As such, it is a specialty within (and I might add the primary motivating force behind) the burgeoning area of definability over finite structures known as finite model theory. What makes this text different from other references is its emphasis on logical definability over classes of structures whose vocabulary is assumed to include not only a linear ordering but also arithmetic on the domain.

Background

What are the main ideas behind this logical view of computation? Instead of quantifying asymptotic physical resources such as time and space required to execute an algorithm on a given model of computation, logic provides a machine-independent approach to the inherent asymptotic difficulty of a problem. Problems are naturally expressed as mappings from finite input structures to output structures - otherwise known as queries on databases. Tractable queries (those computable in polynomial-time or logarithmic-space for example) can be expressed using formulas drawn from extensions/restrictions of first order/second order logic. Because the focus is on characterizing asymptotic resources for finite computations, these logical descriptions lead naturally to the study of definability over finite structures: finite model theory.

Using logical formulas to describe computations has a long history. Consider the class of *binary string* structures given by a finite successor chain with a unary predicate. Call a set R of binary strings *regular* if membership in the corresponding class of structures R' can be recognized by a finite state automata (a type of computer which changes state as it moves along the chain). An old (1960) theorem of Büchi says that R is regular if and only if R' is definable in monadic second order logic. One of the best places to study these results is a nice book by Howard Straubing [*Finite Automata, Formal Logic, and Circuit Complexity*, Birkhauser, Boston, 1994]. Broader connections between logical definability and computational complexity were realized later, when in 1974, Fagin showed that definability in full existential second-order logic captured nondeterministic polynomial time bounded computation (**NP**) over unordered structures. The next decade brought refinements of the technique to yield tantalizing results on ordered structures that captured **P** (polynomial-time) via inductive definability (LFP for least-fixed-point logic) and **L** (logarithmic-space) via restricted recursion using path quantifiers (TC for transitive-closure logic). Even a correspondence with elementary definability in first-order logic was achieved, as those computations which could be computed in a constant amount of time on the CRAM, a concurrent random access machine (a powerful parallel model of computation in which multiple processors have simultaneous non-conflicting access to a global memory). For an alternative characterization, see my own ["A Constant-Space Sequential Model of Computation for First-Order Logic", *Information and Computation* Vol. 143, pp. 231-250 (1998).] However, these last results hold only with the additional numerical predicate *bit* added to the vocabulary (equivalent to having + and *).

Format

These descriptive classes are all explained in great detail with many examples and exercises throughout the book. Proofs of theorems are complete and kept as simple as possible. Running commentary through most proofs makes it possible to understand the gist of an argument without getting bogged down in technicalities if one so desires. Supplementary figures are drawn with great care and detail, and are a welcome aide to many arguments.

Contents

After two background chapters in logic and complexity respectively, the core material of the book commences with first-order reductions (elementary definability turns out to be such a weak class from a complexity theoretic viewpoint that its primary use appears to be in relating queries of higher definitional complexity). These reductions are used to display complete problems for the classes **L**, **NL**, and **P**. From this base, it is natural to consider inductive definitions, which are shown to express exactly those queries in **P** over ordered structures. A finer analysis, based on the closure ordinal, reveals a tight (linear) correspondence between the depth of an induction and the parallel time it takes to evaluate it on the CRAM. Further topics in this section explore the correspondence between the number of variables and the number of processors required, and connections with other forms of parallel computation such as Boolean circuits and alternating models. Another important covered topic is the locality of first-order queries and their linear-time complexity on structures of bounded degree.

The book then turns to more advanced material, starting with second-order definability. Here the reader will find a detailed proof of the correspondence between **NP** and existential second order logic, along with coverage of NP-completeness and the polynomial hierarchy (= full SO logic). Restricting attention to monadic second-order yields lower bounds (via games) for graph connectivity. Attention then turns to a finer analysis of fixed-point (inductive) definitions. Notable among the results is the author's own that fixed-point is closed under negation on classes of finite structures (one of the few pleasant places where finitary definability is better behaved than its infinitary counterpart). A similar normal form for transitive-closure logic - a natural restriction of fixed-point logic to recursion along paths - holds only over ordered finite structures. Following on this is the author's celebrated result that nondeterministic space is closed under complement.

The remaining chapters show increased specialization, with material on the use of non-monotone inductions (partial fixed-point logic) to characterize polynomial space, and the connection between Boolean circuits and built-in relations that depend only upon the size of the domain. The most important of these relations is a total ordering ($<$), whose role in logics describing feasible computations such as polynomial time appears indispensable, even though the resulting query is independent of that order. Unfortunately, this has the defect of allowing definitions with no effective way of deciding whether they might yield order-dependent non-queries (which break isomorphism types over the original unordered structures). The big question then becomes: *Is there a recursive enumeration of the polynomial time queries?* A positive resolution would imply a database programming language which expresses exactly all feasible queries. Subsequent examination of graph coloring and logics that can count provides some insight into this fundamental problem, but any answer remains elusive.

An equally important task would be the use of finite model theory techniques to separate complexity classes. Progress in this direction is illustrated by the use of arguments which attempt to show that certain problems are not first-order expressible (with and without counting). However, current techniques do not work over vocabularies which are strong enough to describe conventional machine based computations. It is unfortunate that we do not have any techniques from finite model theory which imply separations of complexity theoretic significance in this setting.

The book concludes with applications to database theory, and computer aided verification, along with the author's personal views on the significance of descriptive complexity to the larger picture of computational complexity (such as P versus NP) and computational correctness (software reliability). All in all, this book brings a perspective of the exciting area of complexity theory within the reach of the logician. Neil Immerman has done a tremendous service to future researchers, and I for one can acknowledge that his contribution will be greatly appreciated.

STEVEN LINDELL
HAVERFORD COLLEGE