THE BULLETIN OF SYMBOLIC LOGIC
Volume 18, Number 4, Dec. 2012

REVIEWS

The Association for Symbolic Logic publishes analytical reviews of selected books and articles in the field of symbolic logic. The reviews were published in ***The Journal of Symbolic Logic*** from the founding of the JOURNAL in 1936 until the end of 1999. The Association moved the reviews to this BULLETIN, beginning in 2000.

The Reviews Section is edited by Steve Awodey (Managing Editor), John Burgess, Mark Colyvan, Anuj Dawar, Noam Greenberg, Rahim Moosa, Ernest Schimmerling, Alex Simpson, Kai Wehmeier, and Matthias Wille. Authors and publishers are requested to send, for review, copies of books to *ASL, Box 742, Vassar College, 124 Raymond Avenue, Poughkeepsie, NY 12604, USA.*

In a review, a reference "JSL XLIII 148," for example, refers either to the publication reviewed on page 148 of volume 43 of the JOURNAL, or to the review itself (which contains full bibliographical information for the reviewed publication). Analogously, a reference "BSL VII 376" refers to the review beginning on page 376 in volume 7 of this BULLETIN, or to the publication there reviewed. "JSL LV 347" refers to one of the reviews or one of the publications reviewed or listed on page 347 of volume 55 of the JOURNAL, with reliance on the context to show which one is meant. The reference "JSL LIII 318(3)" is to the third item on page 318 of volume 53 of the JOURNAL, that is, to van Heijenoort's *Frege and vagueness*, and "JSL LX 684(8)" refers to the eighth item on page 684 of volume 60 of the JOURNAL, that is, to Tarski's *Truth and proof*.

References such as 49***5*** or 280***1*** are to entries so numbered in *A bibliography of symbolic logic* (the JOURNAL, vol. 1, pp. 121–218).

JAN KRAJÍČEK. ***Forcing with random variables and proof complexity.*** London Mathematical Society Lecture Note Series, vol. 232. Cambridge University Press, 2011, xvi + 247 pp.

Bounded arithmetic has many intimate connections with feasible computational complexity and questions related to the P versus NP problem. Indeed, the original definition of bounded arithmetic, in the form of $I\Delta_0$, by R. Parikh was motivated by connections with linear space computation. It was subsequently recognized that the Δ_0-definable sets are exactly the sets computable in the linear time hierarchy (a subclass of linear space, but not known to be a proper subclass). The early research by C. Dimatracopoulos, J. Paris, A. Wilkie, and others found many connections between bounded arithmetic and computational complexity. With the definition of the bounded arithmetic theories PV by S. Cook, and S^i_2 and T^i_2 by this reviewer, and many subsequent works, the connections between bounded arithmetic and computational complexity became central. In these theories, the core motivations were to characterize the provably total functions of logical theories in terms of computational complexity: the complexity classes considered are feasible, or near-feasible, such as log space, polynomial time, non-deterministic polynomial time, polynomial space, etc.

Bounded arithmetic is also closely connected to propositional proof complexity. There are two primary connections. First, J. Paris and A. Wilkie showed that certain proofs in bounded arithmetic can be translated into polynomial size, or quasipolynomial size, constant depth propositional proofs. A different kind of correspondence between PV (and S^1_2) and

1079-8986/12/1804-0004/$1.50
DOI:10.2178/bsl.1804040

extended Frege proof systems was found by S. Cook. Extended Frege proofs are the usual "textbook style" proof systems based on modus ponens with proof length measured in terms of number of inferences. Under both translations, propositional proofs turn out to be non-uniform versions of proofs in theories of bounded arithmetic. This is analogous to the way that Boolean circuits are essentially non-uniform versions of algorithms. This has been a useful analogy in some ways, as concepts and tools from complexity theory have been used to establish lower bounds in proof complexity. In other ways, however, it has been a frustrating analogy, as there has been less success in establishing lower bounds on proof complexity than in proving lower bounds on circuit complexity. A particularly important and tantalizing problem of this type is to prove a super-polynomial lower bound on the lengths of proofs of constant depth Frege systems for a Boolean language enlarged with unbounded fanin parity gates or, more generally, gates for modular counting mod p. There is hope that this might be achievable as there are lower bounds, due to R. Smolensky and A. Razborov, on the expressive power of formulas in these proof systems for fixed primes p.

It has been a long-standing goal to use model-theoretic techniques for theories of arithmetic to prove lower bounds in computational complexity or proof complexity. Influential early work includes M. Ajtai's work giving lower bounds on the complexity of constant depth formulas for counting: this, along with the work of Furst–Saxe–Sipser was later improved by Yao–Håstad style switching lemmas. Ajtai's work suggested that forcing methods might allow lower bounds to be proved by model-theoretic methods. Another intriguing early work giving model-theoretic constructions of models of arithmetic was the restricted ultraproduct construction of nonstandard models of arithmetic by S. Kochen and S. Kripke; however, this never found application to bounded arithmetic.

The goal of the book under review is to develop model-theoretic methods to attack problems in proof complexity and in bounded arithmetic. One of the main goals was to establish lower bounds for constant depth Frege proofs with parity gates or gates for modular counting mod p. Although this remains an open problem, the book describes a new set of tools for creating models of bounded arithmetic with forcing, and contains many new techniques that are not available elsewhere in the literature.

The first five chapters of the book introduce a forcing method for constructing Boolean-valued models of arithmetic. (Five chapters may sound like a lot, but the chapters are quite short, and five chapters takes one only up to page 46.) The forcing construction starts with an $\aleph_1$-saturated model $\mathcal{M}$ of true arithmetic, and a set $\Omega \in \mathcal{M}$, and selects a set F of functions $f\colon \Omega \mapsto |\mathcal{M}|$. This forms the first-order universe of a structure; measure-theoretic considerations, using Loeb's measure, give a Boolean valuation for sentences over this structure. After the basic definitions and theorems, there are some striking results on how quantifiers can be witnessed (approximately) with elements of F. The fifth chapter concludes by extending these definitions to second order structures.

The next four short chapters develop models of arithmetic created from functions f which are computable with small decision trees (these functions are called "rudimentary"), establish induction and comprehension principles, and discuss a general framework for quantifier elimination. The subsequent three chapters develop a "tree model" that corresponds to models built from functions that are computable with a special type of Boolean decision trees. This incorporates a novel and ingenious construction: the functions are computed by decision trees which are stratified into k levels for some natural number k, plus have small rudimentary decision trees at the bottom level. (Krajíček does not use the terminology "stratified", however.) This is a crucial innovation that makes it possible exploit the switching lemma and a lower bound for parity to prove quantifier elimination properties and to give a witnessing theorem for the second order bounded arithmetic theory V_1^0. The theory V_1^0 can be viewed as an analogue of the theory $I\Delta_0(R)$ where R is a second order predicate.

Chapters 13–16 take the reader to page 98, and give constructions of models of bounded arithmetic based on functions computable by decision trees based on evaluation of low degree (subpolynomial degree) polynomials mod 2. Algebraic models of arithmetic are defined using these algebraically defined functions and a first-order mod 2 quantifier is introduced. Then, with the aid of the Razborov–Smolensky construction, induction, comprehension, and elimination of quantifiers are proved to hold, a witnessing theorem is proved, and an independence result is obtained.

Chapters 17–22 next address the question of lower bounds for constant depth proofs of the pigeonhole principle. For constant depth Frege systems, exponential lower bounds are known already due to work by M. Ajtai, by S. Bellantoni, T. Pitassi, and A. Urquhart, by J. Krajíček, by T. Pitassi, P. Beame, and R. Impagliazzo, and by J. Krajíček, P. Pudlák, and A. Woods based on random restrictions and switching lemmas. These chapters give a model-theoretical construction for nonstandard models where the pigeonhole fails using decision trees with nodes that query values of the pigeonhole principle. (Here again, Krajíček uses the technique of adding small rudimentary circuits to the leaves of the decision tree.) These models, along with a reflection principle and an appeal to the switching lemma, then establish exponential lower bounds for constant depth Frege proofs. Chapter 22 discusses the prospects for establishing superpolynomial lower bounds for proofs of the pigeonhole principle in constant depth Frege systems augmented with parity gates.

The next chapters take up several short topics about independence results for fragments of bounded arithmetic, oracles, and pseudorandom number generators. It is particularly striking how well the model-theoretic approach can accommodate pseudorandom number generators in a natural way. The final part of the book takes up the subject of τ-tautologies, also known as "proof complexity generators". The τ-tautologies have been introduced earlier by Krajíček and independently by M. Alekhnovich, E. Ben-Sasson, A. Razborov and A. Wigderson. These tautologies were inspired in part by the Nisan–Wigderson pseudorandom number generators, and are often conjectured to be examples of tautologies that are hard for strong propositional proof systems such as extended Frege. Chapters 29 and 30 give a survey of prior work on τ-tautologies that can be read independently of the rest of the book. The final chapter then discusses how to formulate some of the central results about τ-tautologies in the model-theoretic forcing framework. There is a technical error in one of the core proofs; however, the applications of the theorem in the second part of the chapter are all correct. Corrections to this part can be found on the book's errata page and a detailed correction in a 2012 preprint by Krajíček entitled *Pseudo-finite hard instances for a student-teaching game with a Nisan–Wigderson generator*.

The book is arranged into very short chapters, which may be a little disconcerting at first, but quickly becomes very comfortable. If nothing else, one feels like one is making good progress in reading through multiple chapters in one sitting. More importantly, the chapter lengths are appropriate for introducing topics and results incrementally.

My overall opinion of the book is highly positive. The book is a research-level exposition of new topics that have not appeared in the literature. It gives a fundamentally new approach to model-theoretic forcing, as well as to independence results in bounded arithmetic and proof complexity. The author's goal for the book was to use these methods to establish new proof complexity lower bounds. This has not yet come to fruition; but nonetheless, the directions are highly intriguing as an new approach for attacking fundamental problems in proof complexity. The first parts of the book should be interesting to anyone working in model theoretic constructions for non-standard models of arithmetic. The book as a whole will be interesting to researchers working on the common interface between bounded arithmetic, model theory, and proof complexity.

Sam Buss

Department of Mathematics, University of California, San Diego, La Jolla, California 92093-0112, USA. sbuss@math.ucsd.edu.

JEFFREY SHALLIT and MING-WEI WANG. *Automatic complexity of strings*. ***Journal of Automata, Languages and Combinatorics***, vol. 6 (2001), pp. 537–554.

CRISTIAN S. CALUDE, KAI SALOMAA and TANIA K. ROBLOT. *Finite-state complexity and randomness*. ***Theoretical Computer Science***, vol. 412 (2011), no. 41, pp. 5668–5677.

CRISTIAN S. CALUDE, KAI SALOMAA and TANIA K. ROBLOT. *State-size hierarchy for finite-state complexity*. ***International Journal of Foundations of Computer Science***, vol. 23 (2012), no. 1, pp. 37–50.

Algorithmic randomness works to capture what it means for an individual object to be structured or predictable. An extremely active branch of computability theory has risen out of interpreting "algorithmic" in terms of Turing machines. Downey and Hirschfeldt's recent book, ***Algorithmic randomness and complexity***, BSL XVIII 126, is an encyclopedic treatment of the developments of algorithmic randomness. A central theme in this field is that an object being random is correlated with it being hard to describe.

It is natural to ask whether insights from this subject could be applied in practice. However, the central tool in calibrating algorithmic randomness, Kolmogorov complexity, is an uncomputable function on finite strings. Attempts to approximate this complexity have led to resource bounded notions of randomness, where we cap the computation time of the underlying Turing machines. Interesting recent work has connected resource bounded Kolmorgorov complexity with compression algorithms, including Lempel–Ziv codes.

Imposing a more stringent restriction, randomness could be defined in terms of finite state machines. Finite automata are good models of linear time, on-line computation. Moreover, many questions about languages recognized and generated by finite automata are decidable and sometimes lead to feasible algorithms.

The first paper under review defines a notion of descriptive complexity (or randomness) in terms of finite-state machines, translating Sipser's 1983 (resource-bounded) CD complexity to this context. The automaton complexity of x, $A(x)$, is defined to be the number of states in the smallest automaton such that the only string of length $|x|$ that is accepted by the automaton is x itself. An automaton which has this property is said to *uniquely accept* x. Shallit and Wang prove that the string x is efficiently recoverable from (knowledge of $|x|$) and any automaton which uniquely accepts it. Thus, uniquely accepting automata give encodings of the corresponding strings. Moreover, they may have significantly shorter descriptions than the string itself: for each n, the string 0^n is uniquely accepted by a single-state automaton. (This property suggests that $A(x)$ is an analogue to length-conditional Kolmogorov complexity rather than the usual Kolomogorov complexity.)

As one might expect, $A(x)$ is a computable function. A crude upper bound is given by observing that for each string of length n, there is a trivial automaton of size $n + 1$ which uniquely accepts it. Shallit and Wang give a somewhat tighter upper bound for the complexity (using the ratio between the length of the string and the size of the alphabet), proved by reference to longest repeated subwords of the string. The computability of $A(x)$ follows immediately from the existence of an upper bound: we can enumerate all automata which have fewer than the upper-bound many states and check whether each uniquely accepts x.

Many of the results proved in the paper draw on techniques from combinatorics on words and word equations to give lower bounds on the automatic complexity of strings. For example, for almost every string x, the automatic complexity of x is proved to be at least $|x|/C$ (the paper proves the result for $C = 13$ but mentions an improvement to $C = 7$). A similar analysis proves that there are many automatically-incompressible strings.

The last main section of the paper extends the definition of automatic complexity to infinite strings. Results in this vein would present interesting parallels to what is known in algorithmic randomness about random and far-from-random reals. Moreover, the study of automatic complexity of infinite strings would continue the tradition of automatic structures (as introduced by Khoussainov and Nerode in 1995) to analyze presentations of infinite objects using automata. Two candidate measures of complexity are given by the limsup and the liminf of the ratio $A(x)/|x|$. If an infinite string is ultimately periodic then a single finite automaton can uniquely accept infinitely many of its initial segments. In this case, the infinite string has automatic complexity equal to zero (in both these measures). However, the converse does not hold: Shallit and Wang give an example of an infinite string which is not ultimately periodic but which has automatic complexity equal to zero. This suggests many open questions, such as classifying the infinite strings which have automatic complexity equal to zero or those which are maximally complicated in some sense.

More recent contributions to finite-state randomness are found in the papers by Calude, Salomaa, and Roblot. Their definition of finite-state complexity follows more closely the notions of algorithmic randomness à la Downey-Hirschfeldt; the finite state machines *generate* the strings they describe rather than *accept* them. They use *transducers*, finite state machines which realize input-output functions. Transducers are specified by a finite set of states, a transition function, and an output function which assigns to each state and input symbol an output word. The output generated by an input string is the concatenation of the output words produced at each state visited while reading the input. A pair (T, p) consisting of a transducer and a finite string describes x if x is the output of T when running on input p. The finite-state complexity of a string, $C(x)$, is defined as the infimum of $|T|+|p|$ over all descriptions of x. However, $|T|$ must be defined in terms of an encoding (and enumeration) of all transducers.

A main focus of this paper is to analyse the properties of enumerations of transducers that make $C(x)$ computable. In particular, $|T|$ is defined by encoding transducers as finite strings such that the set of encodings of transducers is regular. Calude, Salomaa, and Roblot prove that such an enumeration exists and can even be made regular. The rest of their results fix such an enumeration; it is important to remember that the complexity measures associated to different choices for the enumeration may be very different. As in the case of automatic complexity, there is a simple upper bound: $C(x) \leq |x| + 8$ (the additive constant would be different if we choose a different enumeration from the fixed one), which comes from combining the size of the identity transducer with the Invariance Theorem, one of the main theorems in the paper. This theorem says that for every transducer, T, $C(x) \leq C_T(x) + |T|$, where $C_T(x)$ is the finite-state complexity restricted to descriptions which use T. In particular, $C(x)$ is computable. It is noteworthy that the Invariance Theorem holds even though there is no universal transducer: for any regular enumeration of all the transducers, no single transducer is able to simulate all others.

The paper includes several upper and lower bounds on $C(x)$ for natural classes of strings. For example, $C(0^n) = \Theta(\sqrt{n})$. The proofs involve balancing the number of states in the transducer and the length of the input description. Analogously with the notion of random strings, finite-state incompressible strings are defined and exhibited. The proof uses a nice combination of de Bruijn strings and grammar compression techniques.

Calude, Salomaa, and Roblot continue to explore the connection between the encoding of transducers and finite-state complexity in their second paper under review. This paper defines the state-size hierarchy: classifying strings based on the number of states in a transducer which is associated with a minimal description of the string. In particular, the authors prove that there is no upper bound for the number of states in transducers which give minimal descriptions. This distinguishes finite-state complexity from an earlier proposal (by Charikar et al., STOC 2002) in which the transducers involved could always be assumed to have one state.

A central open question for both automatic complexity (Shallit–Wang) and finite-state complexity (Calude, Salomaa, Roblot) is whether the functions $A(x)$ and $C(x)$ are NP-complete. Beyond that, the results in these papers suggest many lines of research that could serve as parallels for algorithmic randomness and sharpen our understanding of the descriptive strength of finite-state machines.

Mia Minnes

Department of Mathematics, University of California San Diego, La Jolla, CA 92093-0112, USA. minnes@math.ucsd.edu.