Review of

Automatic Complexity:
A Computable Measure of Irregularity
Author of Book: Bjørn Kjos-Hanseen
Author of Review: William Gasarch

1 Introduction

Kolmogorov defined the complexity of a string x, $C(x)$, as (roughly) the size of the smallest program (formally Turing Machines or some well-defined model) that prints that string. We give examples which all use a 2-letter alphabet.

- Let $x \in \{0,1\}^n$. Then $C(x) \leq n + O(1)$ because the following program has length $n + O(1)$ and prints x.

  print(x.)

- Let $x = 0^n$. Then $C(x) \leq \log_2 n + O(1)$ because the following program has length $\log_2 n + O(1)$ and prints x:

  For $i = 1$ to n print(0)

  (n is written in binary.)

  The complexity might be lower. For example, if $n = 2^m$ then $C(x) \leq \log_2 \log_2 n + O(1)$.

- Let x be formed by flipping a coin n times and recording. From the first example, $C(x) \leq n + O(1)$. With very high probability $C(x) \geq n + \Omega(1)$.

What if instead of using programs one used finite automata? Finite automata are *recognizers* not *generators* so instead of requiring the device to *output* x, we will require the device to *accept* x. That cannot be right, since otherwise we could simply use a device that accepts every string. In the next section we will define several definitions of the complexity of a string using finite automata that will deal with those issues.

That is the topic of this book; however, the book also goes in a few other directions such as infinite strings.

2 Definitions and More Definitions About Complexity

The book has many definitions. We give some of them here.

Def 2.1

1. A *DFA* is a deterministic finite automaton.
2. An *incomplete DFA* is a DFA except that the transition function

$$\delta : Q \times \Sigma \to Q$$

   may be partial. Acceptance is defined as $M(x)$ accepts if the string is processed completely and at the end is in an accept state.
3. An NFA is a nondeterministic finite automaton.

Def 2.2 Let Σ be a finite alphabet. Let $n \in \mathsf{N}$. For all of the definitions of complexity A (and variants) they should really be $A_\Sigma(x)$ but we will just write $A(x)$. All DFAs reference to assumed to be $(Q, \Sigma, \delta, s, F)$. Let $x \in \Sigma^n$ (so x is of length n).

1. $A(x)$ is the least number of states in the smallest DFA M such that (1) $M(x)$ accepts, and (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects.
2. $A^-(x)$ is the least number of states in the smallest partial DFA M such that (1) $M(x)$ accepts, and (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects.
3. $A^{\text{perm}}(x)$ is the least number of states in the smallest DFA M such that (1) $M(x)$ accepts, (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects, and (3) for all $\sigma \in \Sigma$ the function from Q to Q defined by $f(q) = \delta(q, \sigma)$ is a permutation.
4. $A^{\text{tra}}(x)$ is the least number of states in the smallest DFA M such that (1) $M(x)$ accepts, (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects, and (3) for all $p, q \in Q$ there exists a $\sigma \in \Sigma$ so that $\delta(p, \sigma) = q$.

5. $A_N(x)$ is the least number of states in the smallest NFA M such that (1) $M(x)$ accepts, and (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects, and (3) there is only one accepting path for x. The book also uses A_{Nu} for this notion to distinguish for other definitions (see below) of complexity using an NFA. The u stands for *unique*.

6. $A_{Ne}(x)$ is the least number of states in the smallest NFA M such that (1) $M(x)$ accepts, and (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects. (The e stands for *exact*.)

7. $A_{Nc}(x)$ is the least number of states in the smallest NFA M such that (1) $M(x)$ accepts, and (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects, (3) there is only one accepting path for x, and (4) the one accepting path visits every edge. (The c stands for *covering*.)

8. The book also has some definitions based on the number of *edges* in a DFA, partial DFA, NFA, and variants.

There are many more definitions as well. Page 77 is a chart of around 25 different definitions and how they relate.

3 Types of Theorem About Complexity

There are several types of theorems in this book.

1. An Upper or lower bound on the complexity of a parameterized set of strings is established. Examples:

   Theorem 1.60 $A(0^n1^n) = O(\sqrt{n})$.

   Theorem 1.74 If $a_1, \ldots, a_k$ are k different symbols then $A(a_1^n \cdots a_k^n) = O(n^{1-1/k})$ where the constant in the big-O depends on k.

   The proof of Theorem 1.74 used lemmas about solving Diophantine equations.

   Theorem 2.2 For all Σ, for all $x \in \Sigma^n$, $A_{Nu}(x) \leq \lfloor n/2 \rfloor + 1$.

   Theorem 5.5 For almost all $x \in \{0,1\}^n$, $A(x) \geq \frac{|x|}{13}$.

   One of the theorems used to prove Theorem 5.5 is:

Theorem 5.2 For all $x \in \{0,1\}^n$, $C(x) \leq 12A(x) + 3\log_2 x + O(1)$.

Theorem 5.2 was not proven in this book; however, a reference was given.

2. Theorems that show the different types of complexity are different.

   Examples:

   Theorem 2.19 There exists x such that $A(x) \neq A^-(x)$.

   The x was 001110 and the proof used a computer search.

   Theorem 4.46 $A^{\text{perm}} \neq A^{\text{tra}}$.

4 Other Concepts Covered

1. If $x \in \{0,1\}^n$ then $A_N(x) \leq b(n) = \left\lfloor \frac{n}{2} \right\rfloor + 1$. The *deficiency of* x is $b(n) - A_N(x)$. This concept is studied extensively.

2. In the usual Kolmogorov complexity $C(x|y)$ is the shortest TM that will, when ran with y on the tape, outputs x. This book considers an analogous notion for automatic complexity.

3. Infinite strings. Sample set of theorems: A string x is *square free* if there is no $w \neq e$ such that ww is a substring of x. If $|\Sigma| \in \{1,2\}$ then there are no infinite square-free strings. If $|\Sigma| = 3$ then there is an infinite square-free string.

4. Complexity: Some of the problems that arise from studying Automatic Complexity are NP-complete.

The above list is not complete.

5 Opinion

The concept of modifying Kolmogorov complexity with finite automata instead of TMs is interesting. Some of the theorems have clever proofs and/or bring up other branches of mathematics.

The book gave many different notions of complexity. I stated eight of them in section 2 of this review. This is a problem with the book: there

are too many definitions. Some of them are rather obscure, and not much is proved about them. The book would have been better with fewer definitions and more development of the central ones.

Nonetheless, the book is a good supplement to a course that covers automata theory or Kolmogorov complexity.