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

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 $C(x) \le n + O(1)$ because the following program has length $n + O(1)$ and prints x.

  print(x)

  .

- Let $x = 0^n$. Then $C(x) \le \log_2 n + O(1)$ 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 $C(x) \le \log_2 \log_2 n + O(1)$.

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

What is 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 can't be right since then we could just use a device that accepts everything. In the next section we will define several definitions of the complexity of a string using finite automta 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

Def 2.1

1. A *DFA* is a deterministic finite automata.
2. An *incomplete DFA* is a DFA except that the transition function $\delta\colon Q\times\Sigma\to Q$ 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 non-deterministic finite automata.

Def 2.2 Let Σ be a finite alphabet. For all of the definitions of complexity A (and variants) they should really be $A_\Sigma(x)$ but we will just write $A(9x)$. All DFAs reference to are (Q,Σ,δ,s,F). Let $x\in\Sigma^*$.

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. $A^{\rm perm}-(x)$ 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 $f(q)=\delta(q,a)$ is a permutation.
4. $A^{\rm tra}-(x)$ 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 a $\sigma\in\Sigma$ so that the function $\delta(p,a)=q$.

Def 2.3 Let Σ be a finite alphabet. For all of the definitions of complexity A (and variants) they should really be $A_\Sigma(x)$ but we will just write

1. $A_{Ne}(x)$ 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*.)
2. $A_{Nu}(x)$ 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 u stands for *unique*.)

3. $A_{Nc}(x)$ 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 on accepting path visits every edge. (The c stands for *covering.*)

4. $A_{Np}(x)$ is the least number of states in the smallest NFA M such that (1) $M(x)$ accepts and there are L accepting paths (we use L later), and (2) for all $y \in \Sigma^n - \{x\}$, if y is accepting then the number of accepting paths is $< L$. (The p stands for *plurality.*)

5. $A_{\text{unamb}}(x)$ is the least number of states in the smallest NFA M such that (1) $M(x)$ accepts and there are L accepting paths (we use L later), (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects, and (3) for all $y \in \Sigma^*$, $M(y)$ has at most one accepting path. (The unamb stands for *unambiguous.*)

6. (We will define a complexity based on the number of edges in an NFA. There are other edge-based notions of complexity.) $E_{\text{NC}}(x)$ is the least number of edges a smallest NFA M such that (1) $M(x)$ accepts by a path that visits every edge. (2) for all $y \in \Sigma^n - \{x\}$, $M(y)$ rejects, and (3) for all $y \in \Sigma^*$, $M(y)$ rejects.

7. 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 bounds 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.

   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 $A(x) \neq A^{-1}(x)$.

   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