

Kolmogorov structure functions for automatic complexity

Bjørn Kjos-Hanssen*

January 31, 2020

Abstract

For a finite word w we define and study the Kolmogorov structure function h_w for nondeterministic automatic complexity. We prove upper bounds on h_w that appear to be quite sharp, based on numerical evidence.

1 Introduction

Shallit and Wang [4] introduced automatic complexity as a computable alternative to Kolmogorov complexity. They considered deterministic automata, whereas Hyde and Kjos-Hanssen [3] studied the nondeterministic case, which in some ways behaves better. Unfortunately, even nondeterministic automatic complexity is somewhat inadequate. The string 00010000 has maximal nondeterministic complexity, even though intuitively it is quite simple. One way to remedy this situation is to consider a structure function analogous to that for Kolmogorov complexity.

The latter was introduced by Kolmogorov at a 1973 meeting in Tallinn and studied by Vereshchagin and Vitányi [6] and Staiger [5].

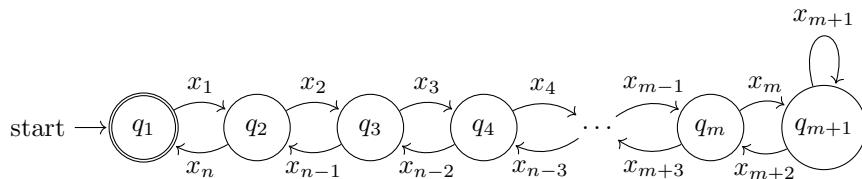


Figure 1: A nondeterministic finite automaton that only accepts one string $x = x_1 x_2 x_3 x_4 \dots x_n$ of length $n = 2m + 1$.

*This work was partially supported by a grant from the Simons Foundation (#315188 to Bjørn Kjos-Hanssen). The author also acknowledges the support of the Institute for Mathematical Sciences of the National University of Singapore during the workshop on *Algorithmic Randomness*, June 2–30, 2014.

The Kolmogorov complexity of a finite word w is roughly speaking the length of the shortest description w^* of w in a fixed formal language. The description w^* can be thought of as an optimally compressed version of w . Motivated by the non-computability of Kolmogorov complexity, Shallit and Wang studied a deterministic finite automaton analogue.

Definition 1 (Shallit and Wang [4]). *The automatic complexity of a finite binary string $x = x_1 \dots x_n$ is the least number $A_D(x)$ of states of a deterministic finite automaton M such that x is the only string of length n in the language accepted by M .*

Hyde and Kjos-Hanssen [3] defined a nondeterministic analogue:

Definition 2. *The nondeterministic automatic complexity $A_N(w)$ of a word w is the minimum number of states of an NFA M , having no ϵ -transitions, accepting w such that there is only one accepting path in M of length $|w|$.*

The minimum complexity $A_N(w) = 1$ is only achieved by words of the form a^n where a is a single letter.

Definition 3. *Let $n = 2m + 1$ be a positive odd number, $m \geq 0$. A finite automaton of the form given in Figure 1 for some choice of symbols x_1, \dots, x_n and states q_1, \dots, q_{m+1} is called a Kayleigh graph¹.*

Theorem 4 (Hyde [2]). *The nondeterministic automatic complexity $A_N(x)$ of a string x of length n satisfies*

$$A_N(x) \leq b(n) := \lfloor n/2 \rfloor + 1.$$

Proof. If n is odd, then a Kayleigh graph witnesses this inequality. If n is even, a slight modification suffices, see [2]. \square

The structure function of a string x is defined by $h_x(m) = \min\{k : \text{there is a } k\text{-state NFA } M \text{ which accepts at most } 2^m \text{ strings of length } |x| \text{ including } x\}$. In more detail:

Let

$$S_x = \{(q, m) \mid \exists \text{ } q\text{-state NFA } M, x \in L(M) \cap \Sigma^n, |L(M) \cap \Sigma^n| \leq b^m\}.$$

Then S_x has the upward closure property

$$q \leq q', m \leq m', (q, m) \in S_x \implies (q', m') \in S_x.$$

From S_x we can define the structure function h_x and the dual structure function h_x^* .

¹ The terminology is a nod to the more famous Cayley graphs as well as to Kayleigh Hyde's first name.

Definition 5 (Vereshchagin, personal communication, 2014, inspired by [6]).
In an alphabet Σ containing b symbols, we define

$$h_x^*(m) = \min\{k : (k, m) \in S_x\} \quad \text{and}$$

$$h_x(k) = \min\{m : (k, m) \in S_x\}.$$

Remark 6. *On the one hand, h mimics the structure function as defined by Kolmogorov. On the other hand, h^* has a natural domain $[0, n]$ whereas the domain of h is initially $[1, \infty)$, until some upper bound on the automatic complexity is proved, at which point it becomes $[1, \lfloor n/2 \rfloor + 1]$. One often prefers that a function have a simple domain and a complicated range rather than the other way around, e.g., consider the case of the range of a computable function on \mathbb{N} (which is only computably enumerable).*

History of the structure function. Kolmogorov first introduced the structure function in a talk at The Third International Symposium on Information Theory, June 18–23, 1973, Tallinn, Estonia, Soviet Union. The meeting coincided with a Nixon/Brezhnev meeting in the U.S. Kolmogorov was born in 1903 hence 70 years old at the time. The results were not published until they appeared as an abstract of a talk for the Moscow Mathematical Society [1] in *Uspekhi Mat. Nauk* in the Communications of the Moscow Mathematical Society, page 155 (in the Russian edition, not translated into English). The talk was given on April 16, 1974 and was entitled “Complexity of algorithms and objective definition of randomness”.

2 Basic properties

Definition 7. *The entropy function $\mathcal{H} : [0, 1] \rightarrow [0, 1]$ is given by*

$$\mathcal{H}(p) = -p \log_2 p - (1-p) \log_2(1-p).$$

Remark 8. *Throughout the paper, \log (with no subscript) denotes either the natural logarithm $\ln = \log_e$, or \log_b where the value of b is immaterial.*

Theorem 9. *For $0 \leq k \leq n$,*

$$\log_2 \binom{n}{k} = \mathcal{H}(k/n)n + O(\log n).$$

Proof. For $u \in \mathbb{N}$, let

$$S_u = \sum_{k=2}^u \log k, \quad I_u = \int_1^u \log x \, dx, \quad \text{and} \quad J_u = \int_2^{u+1} \log x \, dx.$$

Let

$$\alpha_n = \log \binom{n}{k} = S_n - S_k - S_{n-k}.$$

Note $I_u \leq S_u \leq J_u$ and

$$J_u - I_u = \int_u^{u+1} \log x \, dx - \int_1^2 \log x \, dx \leq \log(u+1),$$

Thus up to $O(\log n)$ error terms we have

$$\begin{aligned} \alpha_n &= \int_1^n \log x \, dx - \int_1^k \log x \, dx - \int_1^{n-k} \log x \, dx \\ &= (n \log n - n) - (k \log(k) - k) - [(n-k) \log(n-k) - (n-k)] \\ &= n \log n - k \log(k) - (n-k) \log(n-k) \\ &= -k \log(k/n) - (n-k) \log\left(1 - \frac{k}{n}\right) \end{aligned}$$

and hence

$$\log_2 \binom{n}{k} = -k \log_2(k/n) - (n-k) \log_2(1 - k/n) = \mathcal{H}(k/n) \cdot n.$$

□

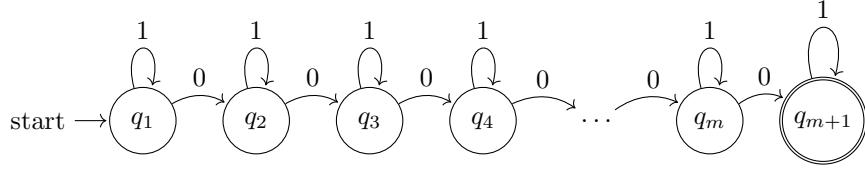


Figure 2: An automaton illustrating multi-run complexity for a string of length n containing m many 0s, and $n - m$ many 1s.

Theorem 10. Suppose the number of 0s in the binary string x is $p \cdot n$. Then

$$h_x^*(\mathcal{H}(p)n) \leq pn + O(\log n).$$

Proof. Consider an automaton M as in Figure 2 that has $p \cdot n$ many states, and that has one left-to-right arrow labeled 0 for each 0, and a loop in place labeled 1 for each consecutive string of 1s. Since M accepts exactly those strings that have $p \cdot n$ many 0s, the number of strings accepted by M is $\binom{n}{p \cdot n}$. By Theorem 9 this is $\leq 2^k$ approximately when $\mathcal{H}(p)n \leq k$, and we are done. □

Example 11. A string of the form $0^a 1^{n-a}$ satisfies $h_x^*(\log_2 n) = 2$ whereas $h_x^*(0)$ may be $n/2$. For instance 0011 has $h_x^*(2) = 2$. On the other hand $h_x^*(1) = 3$ which is why this string is more complicated than 0110.

Theorem 12. For any x of length n ,

$$1 \leq h_x^*(m) \leq n - m + 1 \text{ for } 0 \leq m \leq n.$$

Proof. $1 \leq h_x^*(n - k) \leq k + 1$ because we can start out with a sequence of determined moves, after which we accept everything, as in Figure 3. □

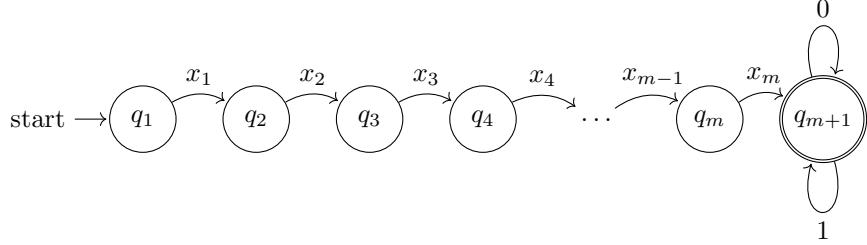


Figure 3: An automaton illustrating the linear upper bound on the automatic structure function from Theorem 12.

3 Upper bounds on structure function for automatic complexity

Definition 13. *The dual automatic structure function of a string x of length n is a function $h_x^* : [0, n] \rightarrow [0, \lfloor n/2 \rfloor + 1]$. We define the asymptotic upper envelope of h^* by*

$$\widetilde{h}^*(a) = \limsup_{n \rightarrow \infty} \max_{|x|=n} \frac{h_x^*([a \cdot n])}{n}, \quad \widetilde{h}^* : [0, 1] \rightarrow [0, 1/2]$$

where $[\cdot]$ is the nearest integer function. Let

$$\tilde{h}(p) = \limsup_{n \rightarrow \infty} \max_{|x|=n} \frac{h_x([p \cdot n])}{n}, \quad \tilde{h} : [0, 1/2] \rightarrow [0, 1].$$

Theorem 14 (Main Theorem). *Assume x is a binary string, so the alphabet size $b = 2$. The asymptotic upper envelope \tilde{h} of the automatic structure functions h_x satisfies*

$$\tilde{h}(p) \leq \psi(p) := \begin{cases} \mathcal{H}(\frac{1}{2} - p), & \frac{\sqrt{3}}{4} \leq p \leq \frac{1}{2}, \\ 2 - \alpha p, & \frac{1}{\alpha-1} \leq p \leq \frac{\sqrt{3}}{4}, \\ 1 - p, & 0 \leq p \leq \frac{1}{\alpha-1}, \end{cases}$$

where

$$\alpha = \frac{4}{\sqrt{3}} \left(2 - \mathcal{H} \left(\frac{1}{2} - \frac{\sqrt{3}}{4} \right) \right) = \mathcal{H}' \left(\frac{1}{2} - \frac{\sqrt{3}}{4} \right) \approx 3.79994,$$

$$\alpha = 2 \log_2(2 + \sqrt{3}).$$

As Theorem 14 shows, the largest number of paths is obtained by going *fairly* straight to the loop state; spending half the time looping and half the time meandering; and then finally going equally fairly straight to the start state. The optimal value of r obtained shows that half of the time between first reaching the loop state and finally leaving the loop state should be spent

looping. Figures 4 and 5 show our upper bounds for the automatic structure function.

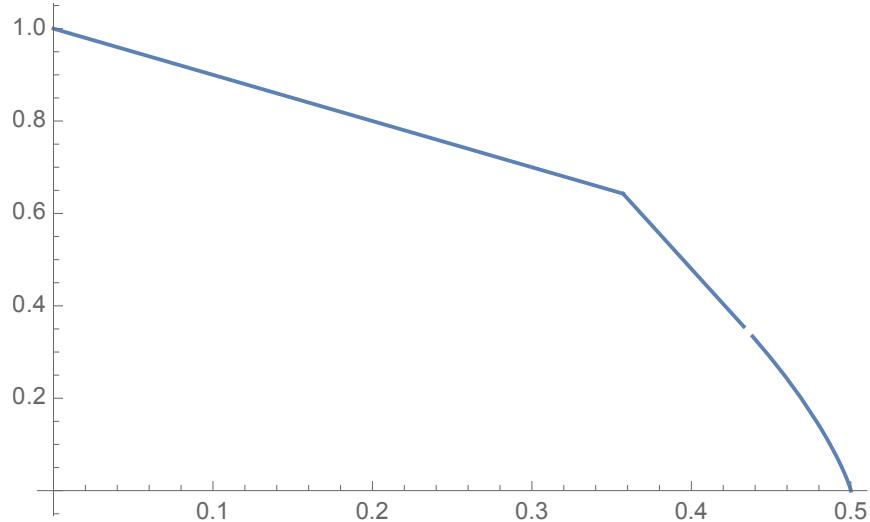


Figure 4: Bounds for the automatic structure function for alphabet size $b = 2$ when $\vec{c} = (1, -1, 0)$; see Theorem 14. Figure produced using Mathematica with $y = \min(2 - 2\log_2(2 + \sqrt{3})x, 1 - x)$ for $0 \leq x \leq \frac{\sqrt{3}}{4}$, and $y = -(\frac{1}{2} - x)\log_2(\frac{1}{2} - x) - (\frac{1}{2} + x)\log_2(\frac{1}{2} + x)$ for $\frac{\sqrt{3}}{4} \leq x$.

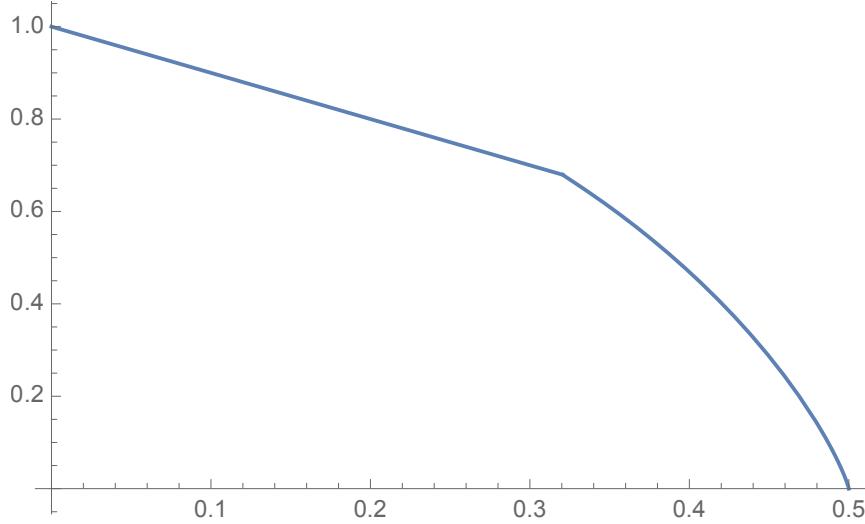


Figure 5: Bounds for the automatic structure function for alphabet size $b = 2$; see Theorem 14. When $\vec{c} = \frac{1}{2}(1, -1, 1)$, the entropy function is used on $[0.33, 0.5]$, the tangent line on $[0.3205, 0.330]$, and $y = 1 - x$ on $[0, 0.3205]$. Figure produced using Mathematica with $y = \min(\sqrt{2} - 2.29244x, 1 - x)$ for $0 \leq x < 0.330$, and $y = -(\frac{1}{2} - x) \log_2(\frac{1}{2} - x) - (\frac{1}{2} + x) \log_2(\frac{1}{2} + x)$ for $0.330 \leq x$.

4 Proof of Theorem 14

Consider a path of length n through a Kayleigh graph with $q = pn$ many states. Let t_1 be the time spent before reaching the loop state for the first time. Let t_2 be the time spent after leaving the loop state for the last time. Let s be the number of self-loops taken by the path. Let us say that *meandering* is the process of leaving the loop state after having gone through a loop, and before again going through a loop. For fixed p let

$$\gamma(t_1, t_2, s, n) = \binom{t_1}{\frac{t_1-pn}{2}} \binom{t_2}{\frac{t_2-pn}{2}} \binom{\chi(n, t, s)}{s} \binom{\chi(n, t, s) - s}{\frac{\chi(n, t, s) - s}{2}} b^s$$

where $\chi(n, t, s) = n - t$ and $t = t_1 + t_2$. (By Lemma 15, we can also let $\chi(n, t, s) = (n - t + s)/2$, since the number of non-loops between loops must be even. This gives a better upper bound.)

Then the number of such paths is

$$N \leq \sum_s \sum_{t_1} \sum_{t_2} \gamma(t_1, t_2, s, n) \tag{1}$$

since half of the meandering times must be backtrack times.

Lemma 15. Suppose $0 \leq k \leq n$ with $n - k$ even. The number of k -element subsets of $\{1, \dots, n\}$ where the number of other elements between consecutive elements in the subset is always even is

$$\binom{(n-k)/2+k}{k}.$$

Proof. The other elements come in pairs hence by merging the pair to one there are only $(n-k)/2$ of them. \square

For instance, if $n = 6$ and $k = 2$, we get $\binom{4}{2} = 6$. Since

$$\limsup_{n \rightarrow \infty} \frac{\log_b \sum_1^n a_i}{n} \leq \limsup_{n \rightarrow \infty} \frac{\log_b(n \cdot \max a_i)}{n} = \limsup_{n \rightarrow \infty} \frac{\log_b \max a_i}{n},$$

the sums can be replaced by maxima, i.e.,

$$\limsup_{n \rightarrow \infty} \frac{\log_b N}{n} \leq \limsup_{n \rightarrow \infty} \frac{\log_b \gamma(t_1, t_2, s, n)}{n}, \quad (t_1, t_2, s) \in \arg \max_{(t_1, t_2, s)} \gamma(t_1, t_2, s, n).$$

By Theorem 9,

$$\limsup_{n \rightarrow \infty} \frac{\gamma(t_1, t_2, s, n)}{n} \leq \limsup_{n \rightarrow \infty} \frac{\delta(t_1, t_2, s, n)}{n}$$

where δ is

$$\begin{aligned} & \sum_{i=1}^2 t_i \mathcal{H}_b \left(\frac{1}{2} - \frac{pn}{2t_i} \right) + (\chi(n, t, s)) \mathcal{H}_b \left(\frac{s}{\chi(n, t, s)} \right) + (\chi(n, t, s) - s) \mathcal{H}_b \left(\frac{1}{2} \right) + s \\ &= \sum_{i=1}^2 t_i \mathcal{H}_b \left(\frac{1}{2} - \frac{pn}{2t_i} \right) + (\chi(n, t, s)) \mathcal{H}_b \left(\frac{s}{\chi(n, t, s)} \right) + \chi(n, t, s) \log_b 2 + (1 - \log_b 2)s, \end{aligned}$$

where $\mathcal{H}_b = \mathcal{H}/\log_2 b$ and $t = t_1 + t_2$. Note that $\mathcal{H}_b(1/2) = 1/\log_2 b$. Now let $\Delta(T_1, T_2, r) = \delta(T_1 n, T_2 n, rn, n)/n$ for any n . It does not matter which n , since with $T = T_1 + T_2$, $\chi(n, t, s) = c_n n + c_t t + c_s s$ gives

$$\frac{1}{n} \chi(n, Tn, rn) = c_n + c_t T + c_s r$$

and

$$\frac{rn}{\chi(n, Tn, rn)} = \frac{r}{c_n + c_t T + c_s r}$$

and hence $\Delta(T_1, T_2, r)$ equals

$$\begin{aligned} & \sum_{i=1}^2 T_i \mathcal{H}_b \left(\frac{1}{2} - \frac{p}{2T_i} \right) + (c_n + c_t T + c_s r) \mathcal{H}_b \left(\frac{r}{c_n + c_t T + c_s r} \right) \\ &+ (c_n + c_t T + c_s r) \log_b 2 + (1 - 1/\log_2 b)r. \end{aligned}$$

Lemma 16. $\Delta(T_1, T_2, r)$ is maximized at $T_1 = T_2$.

Proof. Rewriting with $T = T_1 + T_2$ and $\epsilon = T_1 - T_2$, it suffices to show that with $g(x) = x\mathcal{H}(1/2 - 1/x)$, the function $f(\epsilon) = g(x + \epsilon) + g(x - \epsilon)$ is maximized at $\epsilon = 0$. This is equivalently to g being concave, which is a routine verification:

$$\begin{aligned} g'(x) &= \mathcal{H}(1/2 - 1/x) + x\mathcal{H}'(1/2 - 1/x)(1/x^2) = \mathcal{H}(1/2 - 1/x) + \mathcal{H}'(1/2 - 1/x)/x \\ g''(x) &= \mathcal{H}'(1/2 - 1/x)/x^2 + \mathcal{H}''(1/2 - 1/x)(1/x^2)(1/x) + \mathcal{H}'(1/2 - 1/x)(-1/x^2) \\ g''(x) &= \mathcal{H}''(1/2 - 1/x)/x^3 < 0 \end{aligned}$$

□

Definition 17 (Logit function). For any real $b > 1$,

$$\text{logit}_b(x) = \log_b \left(\frac{x}{1-x} \right), \quad \text{logit}_b : (0, 1) \rightarrow \mathbb{R}.$$

A graphic of the logit function is given as Figure 6.

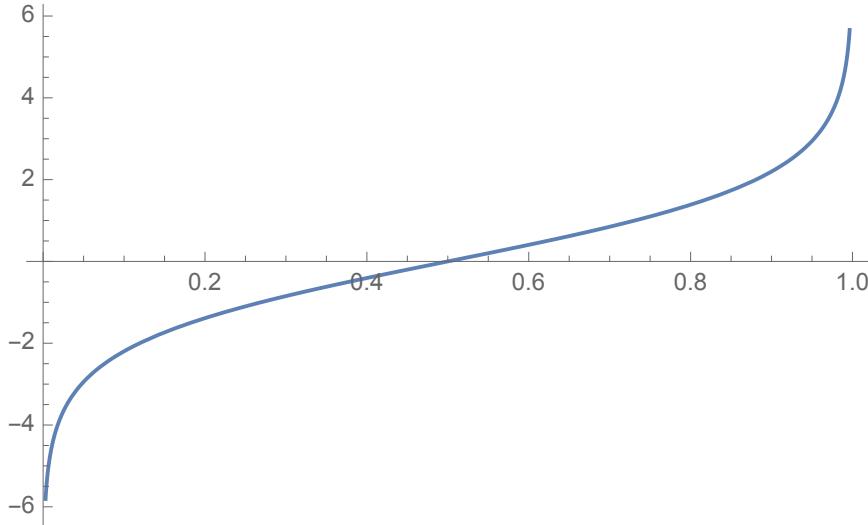


Figure 6: The logit function for $b = e$. Figure produced using Mathematica with $y = \log \left(\frac{x}{1-x} \right)$.

Definition 18 (Logistic sigmoid function). For any real $b > 1$,

$$\zeta_b(y) = \frac{1}{b^{-y} + 1}, \quad \zeta_b : \mathbb{R} \rightarrow (0, 1).$$

Lemma 19. For any real $b > 1$, the logit function $\text{logit}_b(x)$ is a strictly increasing bijection. Its inverse is the logistic sigmoid function $\zeta_b(y)$.

In light of Lemma 16, we now let $\Delta(T, r) = \Delta(T/2, T/2, r)$, so that

$$\begin{aligned}\Delta(T, r) &= T\mathcal{H}_b\left(\frac{1}{2} - \frac{p}{T}\right) + (c_n + c_t T + c_s r)\mathcal{H}_b\left(\frac{r}{c_n + c_t T + c_s r}\right) \\ &\quad + (c_n + c_t T + c_s r)\log_b 2 + (1 - \log_b 2)r.\end{aligned}$$

In the following Lemma it is useful to have the default case $(c_n, c_t, c_s) = (1, -1, 0)$. The other case of interest is $(c_n, c_t, c_s) = (1/2, -1/2, 1/2)$.

Lemma 20. *For fixed p , b , and T , the function $r \mapsto \Delta(T, r)$ has a unique maximum where*

$$r = (c_n + c_t T + c_s r)\frac{b}{b + 2^{1-c_s}},$$

hence after optimizing on r , $\Delta(T) - T\mathcal{H}_b\left(\frac{1}{2} - \frac{p}{T}\right) =$

$$\begin{aligned}(c_n + c_t T + c_s r)\mathcal{H}_b\left(\frac{b}{b + 2^{1-c_s}}\right) &+ (c_n + c_t T + c_s r)\log_b 2 \\ &+ (1 - \log_b 2)\left[(c_n + c_t T + c_s r)\frac{b}{b + 2^{1-c_s}}\right],\end{aligned}$$

i.e.,

$$\begin{aligned}\Delta(T) &= T\mathcal{H}_b\left(\frac{1}{2} - \frac{p}{T}\right) + (c_n + c_t T + c_s r)\underbrace{\left\{\mathcal{H}_b\left(\frac{b}{b + 2^{1-c_s}}\right) + \log_b 2 + (1 - \log_b 2)\left[\frac{b}{b + 2^{1-c_s}}\right]\right\}}_{c_b :=} \\ &= T\mathcal{H}_b\left(\frac{1}{2} - \frac{p}{T}\right) + \frac{(c_n + c_t T)}{\left(1 - c_s \frac{b}{b + 2^{1-c_s}}\right)} c_b \\ &= T\mathcal{H}_b\left(\frac{1}{2} - \frac{p}{T}\right) + (1 - T) \underbrace{\frac{c_n c_b}{\left(1 - c_s \frac{b}{b + 2^{1-c_s}}\right)}}_{d_b} \quad \text{if } c_n = -c_t.\end{aligned}$$

namely

$$\begin{aligned}r &= \frac{(c_n + c_t T)\frac{b}{b + 2^{1-c_s}}}{\left(1 - c_s \frac{b}{b + 2^{1-c_s}}\right)} = \frac{c_n + c_t T}{\left(\frac{b + 2^{1-c_s}}{b} - c_s\right)} \\ &= \begin{cases} (1 - T)\frac{b}{b + 2}, & (c_n, c_t, c_s) = (1, -1, 0) \\ \frac{1}{2}(1 - T)/\left(\frac{b + \sqrt{2}}{b} - \frac{1}{2}\right), & (c_n, c_t, c_s) = (1/2, -1/2, 1/2) \end{cases}\end{aligned}$$

Proof. We have

$$\frac{d}{dx}\mathcal{H}_b(x) = -\text{logit}_b(x).$$

Thus by Lemma 19, the inverse function of $\mathcal{H}'_b(x)$ is $y \mapsto \zeta_b(-y) = \frac{1}{b^y+1}$. Thus, we calculate

$$\begin{aligned} \frac{\partial \Delta}{\partial r} &= \mathcal{H}'_b(r/(c_n + c_t T + c_s r)) + 1 - \log_b 2 + c_s \log_b 2 = 0 \quad \text{iff} \\ \frac{r}{c_n + c_t T + c_s r} &= (\mathcal{H}'_b)^{-1}(\log_b 2 - 1 - c_s \log_b 2) = \frac{1}{b^{\log_b 2 - 1 - c_s \log_b 2} + 1} \\ &= \frac{1}{\frac{2^{1-c_s}}{b} + 1} = \frac{b}{b + 2^{1-c_s}}, \end{aligned}$$

as desired. We also have

$$\frac{d}{dx} \text{logit}_b(x) = \frac{1}{\ln b} \cdot \frac{1}{x(1-x)}.$$

Note that if $(c_n, c_t, c_s) = (1, -1, 0)$ then $T < 1$ gives $r > 0$. Hence

$$\begin{aligned} \frac{\partial^2 \Delta}{\partial r^2} &= \frac{\partial}{\partial r} \left(-\text{logit}_b \left(\frac{r}{c_n + c_t T + c_s r} \right) \right) \\ &= (-1) \frac{\partial}{\partial r} \left(\frac{r}{c_n + c_t T + c_s r} \right) \cdot \frac{1}{\ln(b)} \cdot \frac{1}{x(1-x)} \Big|_{x=\frac{r}{c_n + c_t T + c_s r}} \\ &= (-1) \left(\frac{c_n + c_t T + c_s r - r c_s}{(c_n + c_t T + c_s r)^2} \right) \cdot \frac{1}{\ln(b)} \cdot \frac{1}{x(1-x)} \Big|_{x=\frac{r}{c_n + c_t T + c_s r}} < 0 \end{aligned}$$

provided $c_t = -c_n < 0$ and $c_s > 0$, as in our two cases. \square

In light of Lemma 20, we now let $r = (1-T) \frac{b}{b+2}$ in $\Delta(T, r) = \Delta(T/2, T/2, r)$, giving

$$\begin{aligned} \varphi(T, p) &:= T \mathcal{H}_b \left(\frac{1}{2} - \frac{p}{T} \right) + (1-T) \mathcal{H}_b \left(\frac{r}{1-T} \right) + (1-T) \log_b 2 + (1-\log_b 2)r \\ &= T \mathcal{H}_b \left(\frac{1}{2} - \frac{p}{T} \right) + (1-T) \mathcal{H}_b \left(\frac{b}{b+2} \right) + (1-T) \log_b 2 + (1-\log_b 2)(1-T) \frac{b}{b+2} \end{aligned}$$

which we will call $\varphi(T, p)$.

To simplify calculations to come, we make Definition 21.

Definition 21 (Abbreviations).

$$\begin{aligned} T(p) &:= \frac{2p}{\sqrt{1 - \frac{4}{(b+2)^2}}} = \frac{4p}{\sqrt{3}}, \quad b = 2. \\ \varphi(T, p) &:= \Delta \left(T, (1-T) \frac{b}{b+2} \right) \\ \beta(T) &:= \frac{1}{2} - \frac{p}{T}. \end{aligned}$$

Note that $\alpha_2 = \frac{2 \cosh^{-1}(2)}{\ln 2} = 2 \log_2(2 + \sqrt{3})$. Then

$$\lim_{n \rightarrow \infty} \frac{\log_2 N}{n} \leq \varphi(T, p) = T \mathcal{H}_b(\beta(T)) + (1 - T)c_b.$$

Lemma 22. Suppose $0 \leq 2p \leq T \leq 1$ and $b \geq 2$. Suppose

$$\varphi(T, p) = T \mathcal{H}_b(\beta(T)) + (1 - T)d_b.$$

for some constant d_b . Then we have

$$0 < \frac{\partial \varphi}{\partial T} \iff T < T(p)$$

$$\text{where } T(p) = \frac{2}{\sqrt{1-4b^{-2d_b}}} p.$$

Proof. We have, using the further abbreviation $\beta = \beta(T)$,

$$\begin{aligned} \frac{\partial \varphi}{\partial T} &= \mathcal{H}_b(\beta) + T \mathcal{H}'_b(\beta) \cdot \frac{p}{T^2} - d_b \\ &= \mathcal{H}_b(\beta) + \mathcal{H}'_b(\beta) \frac{p}{T} - d_b = \mathcal{H}_b(\beta) + \mathcal{H}'_b(\beta)(1/2 - \beta) - d_b \end{aligned}$$

Note that $b^{-\mathcal{H}_b(x)} = x^x(1-x)^{1-x}$ and $b^{-\mathcal{H}'_b(x)} = x/(1-x)$. Thus now $0 < \partial \varphi / \partial T$ iff $b^{-0} > b^{-\partial \varphi / \partial T}$ iff

$$1 > \beta^\beta (1 - \beta)^{(1-\beta)} \left(\frac{\beta}{1 - \beta} \right)^{1/2 - \beta} b^{d_b} = \beta^{1/2} (1 - \beta)^{1/2} b^{d_b}, \quad \text{iff}$$

$$1 > \beta(1 - \beta)b^{2d_b}.$$

Since $0 \leq \beta \leq 1/2$, this gives

$$\begin{aligned} \beta &< \frac{1 - \sqrt{1 - 4b^{-2d_b}}}{2}, \\ \frac{p}{T} = \frac{1}{2} - \beta &> \frac{\sqrt{1 - 4b^{-2d_b}}}{2}, \quad \text{and} \\ T &< \frac{2}{\sqrt{1 - 4b^{-2d_b}}} p. \end{aligned}$$

□

Corollary 23. Suppose $(c_n, c_t, c_s) = (1, -1, 0)$. Suppose $0 \leq 2p \leq T \leq 1$ and $b \geq 2$. Then we have

$$0 < \frac{\partial \varphi}{\partial T} \iff T < T(p) := \frac{2}{\sqrt{1 - 4b^{-2d_b}}} p.$$

Proof. We let

$$d_b = \frac{c_n c_b}{1 - c_s \frac{b}{b + 2^{1-c_s}}}$$

$$= \begin{cases} c_b, & \text{if } (c_n, c_t, c_s) = (1, -1, 0), \\ \frac{\frac{1}{2}c_b}{1 - \frac{1}{2}\frac{b}{b+\sqrt{2}}}, & \text{if } (c_n, c_t, c_s) = \frac{1}{2}(1, -1, 1). \end{cases}$$

If additionally we set $b = 2$ then this is

$$= \begin{cases} 2, & \text{if } (c_n, c_t, c_s) = (1, -1, 0), \\ \sqrt{2}, & \text{if } (c_n, c_t, c_s) = \frac{1}{2}(1, -1, 1). \end{cases}$$

We apply Lemma 22. Then $0 < \partial\varphi/\partial T$ iff

$$\beta < \frac{1 - \sqrt{1 - 4b^{-2d_b}}}{2} = \frac{1 - \sqrt{3/4}}{2}, \quad \text{if } b = 2 \text{ and } (c_n, c_t, c_s) = (1, -1, 0), \text{ and}$$

$$\frac{p}{T} = \frac{1}{2} - \beta > \frac{\sqrt{1 - 4b^{-2d_b}}}{2} = \frac{\sqrt{3/4}}{2}, \quad \text{under the same condition.}$$

So

$$T < \frac{2}{\sqrt{1 - 4b^{-2d_b}}} p = \begin{cases} (2.3094)p = \frac{4}{\sqrt{3}}p, & b = 2 \text{ and } (c_n, c_t, c_s) = (1, -1, 0) \\ (3.0259)p = \frac{4}{\sqrt{4 - 4^{2-\sqrt{2}}}}p & b = 2 \text{ and } (c_n, c_t, c_s) = \frac{1}{2}(1, -1, 1). \end{cases}$$

Note that $b \geq 2$ and

$$\begin{aligned} 4b^{-2c_b} &= \left(\frac{4}{b(b+2)} \right)^2 b^{-2(\log_b 2-1)} \\ &= \left(\frac{4}{b(b+2)} \right)^2 \frac{b^2}{4} = \frac{4}{(b+2)^2} \leq \frac{1}{4} < 1 \end{aligned}$$

give $1 - 4b^{-2c_b} > 0$, as required. \square

Let $L_b = \sqrt{1 - 4b^{-2d_b}}$. Note that $T(p) \leq 1$ iff

$$p \leq L_b/2 = \frac{\sqrt{1 - 4b^{-2d_b}}}{2} = \begin{cases} \frac{\sqrt{3}}{4} = 0.433, & b = 2 \text{ and } (c_n, c_t, c_s) = (1, -1, 0) \\ \frac{\sqrt{4 - 4^{2-\sqrt{2}}}}{4} = 0.330 & b = 2 \text{ and } (c_n, c_t, c_s) = \frac{1}{2}(1, -1, 1). \end{cases}$$

and

$$\begin{aligned} \varphi(T(p), p) &= T(p)\mathcal{H}_b \left(\frac{1}{2} - \frac{p}{T(p)} \right) + (1 - T(p))d_b \\ &= \frac{2p}{L_b} \mathcal{H}_b \left(\frac{1}{2} - \frac{L_b}{2} \right) + (1 - \frac{2p}{L_b})d_b \\ &= d_b - \underbrace{\left(d_b - \mathcal{H}_b \left(\frac{1}{2} - \frac{L_b}{2} \right) \right)}_{\alpha_b} \frac{2}{L_b} p \end{aligned}$$

Note

$$L_2 = \begin{cases} \sqrt{3}/2 & \vec{c} = (1, -1, 0) \\ \sqrt{4 - 4^{2-\sqrt{2}}}/2 = \sqrt{1 - 4^{1-\sqrt{2}}} & \vec{c} = \frac{1}{2}(1, -1, 1), \end{cases}$$

so

$$\mathcal{H}_2 \left(\frac{1}{2} - \frac{L_2}{2} \right) = \begin{cases} 0.354579, & \vec{c} = (1, -1, 0) \\ 0.656615, & \vec{c} = \frac{1}{2}(1, -1, 1). \end{cases}$$

Now we need

$$\alpha_2 = \begin{cases} \left(2 - \mathcal{H}_2 \left(\frac{1}{2} - \frac{L_2}{2} \right) \right) \frac{2}{L_2} = 2 \log_2(2 + \sqrt{3}) = 3.7999 & \vec{c} = (1, -1, 0) \\ \left(\sqrt{2} - 0.656615 \right) \frac{2}{\sqrt{1 - 4^{1-\sqrt{2}}}} = 2.29244 & \vec{c} = \frac{1}{2}(1, -1, 1). \end{cases}$$

Hence

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{\log_2 N}{n} &\leq \psi(p) := \varphi(\min\{1, T(p)\}, p) \\ &= \begin{cases} \varphi(1, p) = \mathcal{H}_b(1/2 - p), & p \geq L_b/2; \\ \varphi(T(p), p) = d_b - \alpha_b p, & p \leq L_b/2. \end{cases} \end{aligned}$$

Note

$$\psi'(p) = \begin{cases} \partial_1 \varphi(1, p) \cdot 0 + \partial_2 \varphi(1, p) \cdot 1 & p < L_b/2 \\ \partial_1 \varphi(T(p), p) \cdot T'(p) + \partial_2 \varphi(T(p), p) \cdot 1 & p > L_b/2. \end{cases}$$

We can see that ψ will be differentiable at the breakpoint as follows: by lemma above, $\partial_1 \varphi(T, p) = 0$ exactly at $T = T(p)$, so the first terms are both 0. The second terms are equal since $T(p) = 1$ when $p = L_b/2$. That is, we apply the following lemma with $a = 1$ and $L = L_b/2$.

Lemma 24. Suppose $\varphi(T, p)$ is differentiable. Let $T(p)$ be the value of T such that $\partial_1 \varphi(T, p) = 0$, let L (depending on a) be such that for all p ,

$$T(p) \geq a \quad \text{iff} \quad p \leq L,$$

and define the function ψ by

$$\psi(p) = \varphi(\min(a, T(p)), p).$$

Then ψ is differentiable at L .

Proof.

$$\psi'(p) = \begin{cases} \partial_1 \varphi(a, p) \cdot 0 + \partial_2 \varphi(a, p) \cdot 1 & p < L \\ \partial_1 \varphi(T(p), p) \cdot T'(p) + \partial_2 \varphi(T(p), p) \cdot 1 & p > L. \end{cases}$$

□

Another way is to note that $\frac{d}{dp}\varphi(1, p) = \log_2(\frac{1}{2} - x) - \log_2(\frac{1}{2} + x)$, which at $\sqrt{3}/4$ is $2\log_2(2 - \sqrt{3}) < 0$. On the other hand $\frac{d}{dp}\varphi(T(p), p) = -\alpha_b = -2\log_2(2 + \sqrt{3}) = 2\log_2(\frac{2-\sqrt{3}}{(2+\sqrt{3})(2-\sqrt{3})})$, so ψ is actually differentiable at the breakpoint when $\vec{c} = (1, -1, 0)$. In fact, we have differentiability for any \vec{c} with $c_n = -c_t$, by the identity

$$\log_b\left(\frac{1}{2} - L_b/2\right) - \log_b\left(\frac{1}{2} + L_b/2\right) = -\alpha_b = -\left(d_b - \mathcal{H}_b\left(\frac{1}{2} - \frac{L_b}{2}\right)\right) \frac{2}{L_b}$$

which follows from (and is equivalent to)

$$b^{-2d_b} = \frac{1}{4} - \frac{L_b^2}{4},$$

where $L_b = \sqrt{1 - 4b^{-2d_b}}$.

Consequently $\tilde{h}^*(\psi(p)) \leq p$. Since \tilde{h} is decreasing it follows that $\tilde{h}(p) \leq \psi(p)$.

This completes the proof of Theorem 14.

References

- [1] Meetings of the Moscow Mathematical Society. *Uspehi Mat. Nauk*, 29(4(178)):153–160, 1974.
- [2] Kayleigh Hyde. Nondeterministic finite state complexity. Master’s thesis, University of Hawaii at Manoa, U.S.A., 2013.
- [3] Bjørn Kjos-Hanssen and Kayleigh Hyde. Nondeterministic automatic complexity of almost square-free and strongly cube-free words. In *COCOON 2014*, volume 8591 of *Lecture Notes in Comput. Sci.*, pages 61–70. Springer, Heidelberg, 2014.
- [4] Jeffrey Shallit and Ming-Wei Wang. Automatic complexity of strings. *J. Autom. Lang. Comb.*, 6(4):537–554, 2001. 2nd Workshop on Descriptive Complexity of Automata, Grammars and Related Structures (London, ON, 2000).
- [5] Ludwig Staiger. The Kolmogorov complexity of infinite words. *Theoret. Comput. Sci.*, 383(2-3):187–199, 2007.
- [6] Nikolai K. Vereshchagin and Paul M. B. Vitányi. Kolmogorov’s structure functions and model selection. *IEEE Trans. Inform. Theory*, 50(12):3265–3290, 2004.