Exposition on Brandt24:

“Lower Bounds for Levin–Kolmogorov Complexity”

Hanlin Ren
IAS & MPI
h4n1in.r3n@gmail.com

September 10, 2026

Abstract

We present a simplified proof of the main result in [Bra24] showing that MKtP $\notin \mathsf{DTIME}[O(n)]$. The simplified proof was found by GPT-6 Astra.

Let U be a universal Turing machine. The *Levin–Kolmogorov complexity* of a string x is defined as the minimum of $|d| + \lceil \log_2 t \rceil$ over all pairs (d, t) such that $\mathrm{U}(d)$ prints x in t steps. We denote this quantity by $\mathrm{Kt}(x)$.

We require the universal Turing machine U to satisfy the following three properties:

1. *U is prefix-free.* That is, no valid program is a prefix of another valid program.

2. *U does not “cheat”.* That is, if $\mathrm{U}(d)$ prints a string of length n, then it runs at least n steps.

3. *U has logarithmic simulation overhead.* That is, if d is a program that runs in t steps, then $\mathrm{U}(d)$ runs in $O(t \log t)$ steps.

The first two properties imply the following lemma:

Lemma 1. *Let B_n denote the number of n-bit strings x such that $\mathrm{Kt}(x) \le n$. Then*

$$\sum_{n \ge 1} B_n \cdot (n/2^n) \le 1.$$

Proof. For every string x, define (d_x, t_x) to be the “Kt-witness” of x; that is, d_x is a program that prints x in t_x time and $|d_x| + \lceil \log_2 t_x \rceil = \mathrm{Kt}(x)$. We break ties arbitrarily.

Since our universal Turing machine is *prefix-free*, there are no $x \ne y$ such that d_x is a prefix of d_y. This implies the following version of *Kraft’s Inequality*:

$$\sum_{x \in \{0,1\}^*} 2^{-|d_x|} \le 1.$$

If x is an n-bit string and $\mathrm{Kt}(x) \le n$, then we have $|d_x| \le n - \lceil \log_2 n \rceil$, since *any program printing x needs to run for at least n steps*. This means that $2^{-|d_x|} \ge n/2^n$ and

$$\sum_{n \ge 1} B_n \cdot (n/2^n) \le \sum_{x \in \{0,1\}^*} 2^{-|d_x|} \le 1. \qquad \square$$

Note: The measure $\widetilde{\mathrm{Kt}}$ defined in [RS22] is not *prefix-free*, hence Lemma 1 fails for $\widetilde{\mathrm{Kt}}$. In fact it requires non-relativizing techniques to prove a super-linear time lower bound for $\mathrm{M}\widetilde{\mathrm{Kt}}\mathrm{P}$, but the techniques in [Bra24] and this note are relativizing.

Now we can prove the desired lower bound:

Theorem 2. $\mathrm{MKtP} \notin \mathsf{DTIME}[O(n)]$.

Proof. Assume, towards a contradiction, that there is a deterministic algorithm that decides MKtP in $O(n)$ time. Fix a large enough integer b, consider the following algorithm $\mathcal{A}$:

Algorithm 1 The algorithm $\mathcal{A}$

for $n \leftarrow b^2, b^2+1, \ldots$ **do**
- Enumerate the first $\ell_n := \lceil \frac{2^{n-b}}{n^2} \rceil$ length-n strings
- If any string has Kt complexity greater than n, print this string and halt

Does $\mathcal{A}$ halt?

- **If $\mathcal{A}$ halts**, then it prints a string x; let N denote the length of x, then $\mathrm{Kt}(x) \le N$.

  However, the running time of $\mathcal{A}$ is at most

  $$O\left(\sum_{n=b^2}^{N} \ell_n \cdot n + \mathrm{poly}(n)\right) \le O(2^{N-b}/N).$$

  Since U has logarithmic simulation overhead, $\mathrm{U}(\langle \mathcal{A} \rangle)$ prints x in $O(2^{N-b})$ steps.

  The description length of $\mathcal{A}$ is $O(\log b)$, as we only need to hardwire the value of b and an additional program of $O(1)$ length.

  It follows that $\mathrm{Kt}(x) \le O(\log b) + N - b \le N$, a contradiction.

- **If $\mathcal{A}$ does not halt**, then we have that $B_n \ge \ell_n$ for all $n \ge b^2$. (Recall that B_n is the number of n-bit strings x such that $\mathrm{Kt}(x) \le n$.)

  This implies that

  $$\sum_{n\ge 1} B_n \cdot (n/2^n) \ge \sum_{n \ge b^2} \ell_n \cdot (n/2^n) \ge 2^{-b} \cdot \sum_{n\ge b^2} 1/n = \infty,$$

  which contradicts Lemma 1 above. □

The above proof recovers the following corollaries in [Bra24]:

Corollary 3 (Corollary 3 in [Bra24])**.** *Suppose that* U *achieves constant-overhead simulation, then* $\mathrm{MKtP} \notin \mathsf{DTIME}[O(n^2)]$.

Proof Sketch. The proof is exactly the same as Theorem 2. Assume MKtP can be decided in $O(n^2)$ time. **If $\mathcal{A}$ halts** and prints a string of length N, then it runs in $O(2^{N-b})$ steps. If U achieves *constant-overhead simulation*, then $\mathrm{U}(\langle \mathcal{A} \rangle)$ still runs in $O(2^{N-b})$ steps, a contradiction. The case that **$\mathcal{A}$ does not halt** is exactly the same as in Theorem 2. □

Corollary 4 (Corollary 2 in [Bra24])**.** *Let* $\ln^{(i)} n$ *denote the* i *times iterated logarithm function (e.g.,* $\ln^{(0)} n = n$ *and* $\ln^{(3)} n = \ln\ln\ln n$*), then*

$$\mathrm{MKtP} \notin \bigcup_{k\ge 0} \mathsf{DTIME}\left[\prod_{i=0}^{k} \ln^{(i)} n\right].$$

Proof Sketch. Fix an integer $k \geq 0$. To prove that MKtP $\notin \mathsf{DTIME}\left[\prod_{i=0}^{k} \ln^{(i)} n\right]$, we run the proof of Theorem 2 with $\ell_n := \left\lceil \frac{2^{n-b}}{n^2 \prod_{i=1}^{k} \ln^{(i)} n} \right\rceil$.

Still, **if $\mathcal{A}$ halts**, then it prints a string x whose Kt complexity is too small, a contradiction; **if $\mathcal{A}$ does not halt**, then

$$\sum_{n \geq 1} B_n \cdot (n/2^n) \geq 2^{-b} \sum_{n \geq b^2} \frac{1}{\prod_{i=0}^{k} \ln^{(i)} n},$$

since this is a divergent sum, we still contradict Lemma 1 above. □

References

[Bra24] Nicholas Brandt. "Lower Bounds for Levin-Kolmogorov Complexity". In: *TCC (1)*. Vol. 15364. Lecture Notes in Computer Science. Springer, 2024, pp. 191–221 (cit. on pp. 1, 2).

[RS22] Hanlin Ren and Rahul Santhanam. "A Relativization Perspective on Meta-Complexity". In: *STACS*. Vol. 219. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 54:1–54:13 (cit. on p. 2).