# Lower Bounds for Levin–Kolmogorov Complexity

Nicholas Brandto

Department of Computer Science ETH Zurich Zurich, Switzerland nicholas.brandt@inf.ethz.ch

The hardness of Kolmogorov complexity is intricately connected to the existence of one-way functions and derandomization. An important and elegant notion is Levin's version of Kolmogorov complexity, Kt, and its decisional variant, MKtP. The question whether MKtP can be computed in polynomial time is particularly interesting because it is not subject to known technical barriers such as algebrization or natural proofs that would explain the lack of a proof for MKtP  $\notin$  P.

We take a major step towards proving  $\mathsf{MKtP} \notin \mathsf{P}$  by developing a novel yet simple diagonalization technique to show *unconditionally* that  $\mathsf{MKtP} \notin \mathsf{DTIME}[\mathcal{O}(n)]$ , i.e., no deterministic linear-time algorithm can solve  $\mathsf{MKtP}$  on every instance. This allows us to affirm a conjecture by Ren and Santhanam [RS22] about a non-halting variant of Kt complexity.

Additionally, we give *conditional* lower bounds for MKtP that tolerate either more runtime or one-sided error. If the underlying computational model has a linear-time universal simulation, e.g. random-access machines, then we obtain a quadratic lower bound, i.e., MKtP  $\notin$  DTIME[ $\mathcal{O}(n^2)$ ].

### Contents

1	Introduction	1
2	Contributions & Related Work	2
3	Technical Overview	5
4	Preliminaries	10
<b>5</b>	Formal Results	12
6	Proof of Lemma 1	18
7	Acknowledgments	26
А	Technical Lemmas	29

# 1 Introduction

The formal concept of "complexity" was spearheaded in the 1960's by Solomonoff [Sol60; Sol64a; Sol64b], Kolmogorov [Kol63; Kol65], and Chaitin [Cha66; Cha69]. Ideas and techniques from meta-complexity—the computational hardness of complexity—have diffused into adjacent subfields like learning theory, deran-domization and cryptography (see Section 2 for related work). We refer to Trakhtenbrot [Tra84] for a historical survey of complexity and to the more recent survey by Allender [All21].

In this work we focus on Levin's notion of Kolmogorov complexity Kt [Lev84], which elegantly incorporates a time bound and thus evades the undecidability of the original Kolmogorov complexity. The Levin–Kolmogorov complexity of a given string x is the minimum over all programs that produce x of the sum of the program's length plus the logarithm of its runtime, i.e.,  $\operatorname{Kt}(x) = \min_{\Pi \mapsto x} (|\Pi| + \lceil \log_2(t) \rceil)$  where  $\Pi$  computes the string x in time t. Its decisional problem is defined as  $\mathsf{MKtP} := \{(x, k) \mid \operatorname{Kt}(x) \leq k\}$ . For an in-depth introduction to meta-complexity problems we refer the reader to [LV08].

In fascinating works Liu and Pass [LP20; LP21b] uncover a surprising connection between derandomization and the existence of one-way functions (OWF) through Kt complexity. On the one hand, they show that (weak) derandomization  $BPP \neq EXP$  is equivalent to the *zero*-sided average-case hardness of MKtP, and on the other that the existence of OWFs is equivalent to the *two*-sided average-case hardness of MKtP. One-way functions are central to modern cryptography: they characterize *symmetric* cryptography, dubbed "Minicrypt" by Impagliazzo [Imp95]. They are necessary and sufficient for: digital signatures [Rom90], (cryptographic) pseudorandom generators [BM82; HIL<sup>+</sup>99], pseudorandom functions [GGM84], private-key encryption [GM84], commitment schemes [Nao91] and much more. Moreover, the existence of OWFs is itself equivalent to the hardness of many other meta-complexity problems (see at the end of Section 2).

These cross connections add to the importance of understanding the hardness of Kolmogorov complexity. While most variants of complexity have (reasonable) unconditional lower bounds (again see Section 2 for related work) and despite the plausible conjecture  $\mathsf{MKtP} \notin \mathsf{NP}$ , only a comparatively weak unconditional lower bound for Kt complexity is known. Namely, Hirahara [Hir20b] shows that the Ktrandom strings  $R_{\mathrm{Kt}} \coloneqq \{x \mid \mathrm{Kt}(x) \ge |x|\}$  are immune<sup>1</sup> to the circuit class P-uniform  $\mathsf{ACC}^0$  (constant depth circuits with constant-modulo gates). Now, one might ask:

#### Why are there no stronger lower bounds for Kt complexity?

The reason that Hirahara's approach fails for stronger classes is that it requires a satisfiability (SAT) solver of the given class. In fact, Hirahara shows that immunity of  $R_{\rm Kt}$  for class C result is indeed equivalent to a SAT solver for C—which explains the lack of a stronger immunity lower bound. However, even considering a weaker (compared to immunity) *worst-case* lower bound, the EXP-completeness of MKtP under BPP reductions [LP21b] explains why there is no worst-case lower bound against probabilistic polynomial-time algorithms (BPP); because it would imply BPP  $\neq$  EXP which itself is subject to the relativization barrier [BGS75]. In the face of this barrier we might ask about an even weaker worst-case lower bound against a deterministic polynomial-time algorithms (P). Even proving the comparatively weaker statement MKtP  $\notin$  P (mentioned e.g. in [Oli19; Hir20b]) is a longstanding open problem at least since Allender, Buhrman, Koucký, van Melkebeek, and Ronneburger [ABK<sup>+</sup>02] posed it explicitly in 2002. This is particularly interesting because MKtP  $\notin$  P is *not*<sup>2</sup> subject to technical barriers like algebrization [AW08; IKK09; AB18] or natural proofs [RR97]. Given the lack of barriers it is not clear whether relativizing techniques suffice to prove MKtP  $\notin$  P. That our lower bounds relativize can be taken as a hint that relativizing techniques might in fact be strong enough to prove MKtP  $\notin$  P.

#### 2 Contributions & Related Work

Our main contribution is a new diagonalization technique tailored to Kt complexity. Using our technique we give the first unconditional lower bound of Kt complexity against a uniform time class. This constitutes a significant step towards proving  $MKtP \notin P$ .

While our diagonalization strategy is fairly simple, its analysis is somewhat involved and simplifying it would be interesting on its own. We stress that our approach differs strongly from all previous approaches like the one of Hirahara [Hir20b] or for randomized complexity notions [Oli19; Hir22b]. A major technical difficulty for Kt lower bounds based on diagonalization is that the diagonalization algorithm for Kt needs to be *deterministic*, and thus no probabilistic tools from complexity theory are available. In Section 3 we explain why this leads to a black-box barrier for diagonalization-based proofs and how our technique overcomes it. Also, note that derandomization is not useful here because a) we are interested in an *unconditional* bound, and b) Liu and Pass [LP21b] already show that (weak) derandomization implies a stronger zero-sided lower bound. Our main result is summarized as follows:

<sup>&</sup>lt;sup>1</sup> No infinite subset of  $R_{\rm Kt}$  is in P-uniform ACC<sup>0</sup>.

 $<sup>^2</sup>$  Ren and Santhanam [RS22] show that the relativization barrier applies to the problem of approximating MKtP.

**Theorem 1.** The Levin–Kolmogorov complexity cannot be decided in deterministic linear time in the worst-case, i.e.,  $\mathsf{MKtP} \notin \mathsf{DTIME}[\mathcal{O}(n)]$ .

On the Kt notion of Ren and Santhanam. Because our lower bound relativizes we can partially affirm a conjecture (Open Problem 4.7.) by Ren and Santhanam [RS22]. They introduce a "non-halting" variant  $\widetilde{Kt}$  of Levin–Kolmogorov complexity whose definition<sup>3</sup> is almost identical to the standard Kt complexity except that the witness program producing a given string need not halt after writing the string on its tape. Ren and Santhanam conjecture that—despite their close definitions—the two notions behave quite differently in that infinitely many strings x have  $\widetilde{Kt}(x) \leq Kt(x)$ . By analyzing the proof of Theorem 1 we can give a concrete example affirming their conjecture. Concretely, infinitely many prefixes of Chaitin's constant  $\Omega$  have  $\widetilde{Kt}(\Omega_1 || ... || \Omega_\ell) <_{io} K(\Omega_1 || ... || \Omega_\ell) \leq Kt(\Omega_1 || ... || \Omega_\ell)$ . To see this assume the opposite (all-but-finitely many prefixes have  $\widetilde{Kt}(\Omega_1 || ... || \Omega_\ell) \geq_{abf} Kt(\Omega_1 || ... || \Omega_\ell)$ ), then our proof of Theorem 1 allows us to prove the linear-time hardness of  $\widetilde{Kt}$  relative to any oracle. However, Ren and Santhanam [RS22] already give an oracle relative to which  $\widetilde{Kt}$  is computable in linear time. Pushing the limits of our technique we find  $\widetilde{Kt}(\Omega_1 || ... || \Omega_\ell) \leq_{io} Kt(\Omega_1 || ... || \Omega_\ell) - \Theta(\ln \ln(\ell))$  falling short of the stronger conjecture  $\widetilde{Kt}(\Omega_1 || ... || \Omega_\ell) \leq_{io} Kt(\Omega_1 || ... || \Omega_\ell) / \Theta(1)$  as required by Ren and Santhanam.

In particular, relative to their oracle Kt can be approximated in linear time to within a multiplicative factor of  $2+\epsilon$  for any  $\epsilon > 0$ . Our relativizing result is compatible with [RS22] because Ren and Santhanam only show that proving hardness of Kt for small thresholds  $\leq n/(2+\epsilon)$  requires a non-relativizing proof but we show hardness of Kt for a large threshold  $\geq n$ . Consequently, showing (worst-case) hardness of Kt for small thresholds seems qualitatively harder than for large thresholds. This should be contrasted with recent developments [LP21a; LP23b] where the worst-case hardness (of a conditioned version) of K<sup>t</sup> for different thresholds between  $n^{\delta}$  and n-2 is equivalent (Thm 1.1. in [LP23b]).

Comparison to Hirahara's lower bound. Hirahara [Hir20b] shows an incomparable unconditional lower bound for Kt complexity, namely, that the Kt-random strings  $R_{\text{Kt}}$  are immune to P-uniform ACC<sup>0</sup> (see [All21] for a nice description of Hirahara's approach). Compared to Hirahara's immunity lower bound (no infinite subset can be decided), our result is weaker in that it only provides worst-case hardness (no algorithm can decide correctly for *every* string). On the other hand, our lower bound holds against deterministic linear time DTIME[ $\mathcal{O}(n)$ ] which—we argue—is closer to P than the rather weak circuit class P-uniform ACC<sup>0</sup> for which Hirahara's lower bound holds. The only case in which our result would be subsumed by [Hir20b] is the implausible case that P = P-uniform ACC<sup>0</sup> which would already imply MKtP  $\notin$  P and in fact a nontrivial SAT solver for P.

We emphasize that our proof strategy differs conceptually from the one in [Hir20b]. The approach of Hirahara is based on the "algorithmic method" of Williams [Wil13; Wil14] where a nontrivial satisfiability algorithm for a circuit class yields a lower bound against that class. Obtaining a stronger immunity of  $R_{\rm Kt}$  using the Hirahara–Williams approach is equivalent to satisfiability algorithms for stronger circuit classes which may be subject to known barriers such as algebrization [AW08; IKK09; AB18] or natural proofs [RR97]. In comparison, our approach opens new avenues for improved lower bounds that possibly evade these barriers. See Section 3 for a discussion of the limitations of our technique and possible ways to overcome them.

Stronger conditional bounds. By analyzing our approach for the proof of Theorem 1 we are able to give conditional lower bounds which either tolerate larger runtime or one-sided error.

**Theorem 2.** For each time bound  $\mathfrak{t}(n) \geq n$  at least one of the following is true:

<sup>&</sup>lt;sup>3</sup> Formally, Ren and Santhanam [RS22] define their  $\widetilde{\text{Kt}}$  notion not relative to any UTM but more informally "over all machines". We thus consider a  $\widetilde{\text{Kt}}$  notion that is defined formally analogously to our notion Definition 2.

1. MKtP  $\notin$  DTIME[t],

2. MKtP  $\notin \operatorname{Heur}_{\gamma_{\mathsf{fp}},\gamma_{\mathsf{fn}}}\mathsf{DTIME}[\mathcal{O}(n)]$  with no false positive error  $\gamma_{\mathsf{fp}}(n) \coloneqq 0$  and false negative error  $\gamma_{\mathsf{fn}}(n) \coloneqq \frac{1}{2n\mathfrak{t}(2n)} - \frac{2}{2^n}$ ,

More related work. In recent years there has been a flurry of meta-complexity results—too many to discuss here ([Hir18; Oli19; GII<sup>+</sup>19; Ila20a; Hir20c; Ila20b; Hir20a; Hir20b; LOS21; RS21; HN22; LO22; LOZ22; Hir22a; AHT23; LP23c; LP23a; BLM<sup>+</sup>23; MP24b; MP24a; LP24] to name only a few). Here, we restrict ourselves to some Kt-related notions and their resp. lower bounds to contextualize our lower bound for MKtP.

The canonical time-bounded variant  $\mathsf{MK}^{\mathsf{t}}\mathsf{P}$  [Kol63; Sip83; Har83; Ko86] of Kolmogorov complexity is parameterized over some time bound  $\mathfrak{t}$  and limits the witness program of a given string x to run in time at most  $\mathfrak{t}(|x|)$ . Limiting the witness program's runtime makes this notion computable, opposed to standard Kolmogorov complexity. For exponential time bounds  $\mathfrak{t}$  Hirahara [Hir20b] shows that  $\mathsf{MK}^{\mathsf{t}}\mathsf{P}$  is EXP-complete under ZPP reductions and even that the set of  $\mathsf{K}^{\mathsf{t}}$ -random strings is immune to  $\mathsf{P}$  (no infinitely large subset of  $\mathsf{K}^{\mathsf{t}}$ -random strings is in  $\mathsf{P}$ ).

Allender, Buhrman, Koucký, van Melkebeek, and Ronneburger [ABK<sup>+</sup>06] show that the Levin– Kolmogorov complexity MKtP is EXP-complete under P/poly or NP reductions, i.e., MKtP  $\in$  P/poly  $\iff$  EXP  $\subseteq$  P/poly. Liu and Pass [LP21b] improve this to BPP reductions, i.e., MKtP  $\in$  BPP  $\iff$  EXP = BPP. Thus, any nontrivial derandomization BPP  $\neq$  EXP is equivalent to a lower bound MKtP  $\notin$  BPP against bounded-error probabilistic TMs. In turn, this means that any barrier preventing us from proving BPP  $\neq$  EXP also prevents us from proving the randomized lower bound MKtP  $\notin$  BPP. In contrast, our lower bound MKtP  $\notin$  DTIME[ $\mathcal{O}(n)$ ] is much weaker both in the quantitative runtime (linear vs. polynomial) as well as the computational model (deterministic vs. probabilistic)—and thus evades known barriers.

Oliveira [Oli19] introduces rKt—a randomized version of Levin–Kolmogorov complexity—where the witness program of a given string x must produce that string x on at least a 2/3-fraction of randomnesses. This randomized complexity is BPE-complete (Lemma 12 in [Oli19]) and Oliveira shows hardness of his notion against quasipolynomial time bounded-error TMs, i.e., MrKtP  $\notin$  BPTIME[ $n^{\log(n)}^{\Theta(1)}$ ]. Later Hirahara [Hir22b] improves that bound to GapMrKtP  $\notin$  io-BPTIME[ $2^{\epsilon n}$ ] for any  $\epsilon \ge 0$ . Oliveira [Oli19] also gives a potential avenue toward proving MKtP  $\notin$  P via the implication MrKtP  $\in$  Promise-EXP  $\Longrightarrow$  MKtP  $\notin$  P.

For a nondeterministic NEXP-complete complexity notion KNt Allender, Koucký, Ronneburger, and Roy [AKR<sup>+</sup>11] show unconditionally that the set of KNt-random strings is not in NP  $\cap$  co-NP.

The canonical problem for circuit complexity is nowadays called the minimum circuit size problem (MCSP) [KC00]. It has been previously considered by Trakhtenbrot [Tra84] (Task 4), and Levin reportedly delayed the publication of his work on NP-completeness to include MCSP. Since  $MCSP \in NP$  an unconditional lower bound seems unlikely; the question is rather whether MCSP is NP-complete which is related to major open questions in theoretical computer science. We refer the interested reader to [All21; AIV21] and references therein for more details about the NP-completeness of MCSP.

Oliveira, Pich, and Santhanam [OPS19] give "hardness magnification" results for gap versions of MKtP and MCSP. They establish that slightly improved lower bounds for these problems can be "magnified" to strong lower bounds. The reason why we cannot use their result to magnify our linear-time lower bound is a difference in the parameter regime (similar to [RS22]). They consider the hardness of distinguishing strings of low complexity from string of even lower complexity (e.g.  $n^{\epsilon} + \Theta(\log n)$  vs.  $n^{\epsilon}$ ). On the other hand, we crucially use the fact (as our counter assumption) that we are able to exactly compute the complexity of a given string  $x \in \{0, 1\}^n$  even when  $Kt(x) \approx n$ .

Huang, Ilango, and Ren [HIR23] show unconditional hardness of an oracle variant of the minimum circuit size problem (MOCSP) using a cryptographic tool called witness encryption [GGS<sup>+</sup>13].

Connection to one-way functions. In recent years there has emerged a research effort to characterize one-way functions (OWF) by the hardness of meta-complexity problems. As an incomplete list: OWFs are equivalent to the mild two-sided hardness of  $MK^tP$  [LP20], the two-sided hardness of MKtP [LP21b], the two-sided hardness<sup>4</sup> of an (NP-complete) conditional variant McKTP [ACM<sup>+</sup>21] of Allender's KT complexity [All01], the mild two-sided hardness of (parameterized versions of)  $MK^tP$  against sublinear time over a smooth range of parameters [LP21a], the mild average-case hardness of the probabilistic MpK<sup>t</sup>P (introduced in [GKL<sup>+</sup>22]) for polynomial t [LP23c], the *worst-case* hardness of a promise version of MK<sup>t</sup>P (with small computational depth) [LP23b], the hardness of a distributional variant of Kolmogorov complexity under the assumption NP  $\notin$  io-P/poly [Hir23].

#### 3 Technical Overview

To simplify this overview, we assume that the UTM  $\mathcal{U}$  simulates any given program  $\Pi$  without any overhead. In the formal proof we will account for the logarithmic overhead of the UTM.

A natural approach to proving lower bounds for a given meta-complexity problem is to assume that the problem is easy and then leverage an efficient solver for that problem to quickly construct a highly complex string (w.r.t. to the given complexity measure). The historical proof of the undecidability of standard Kolmogorov complexity as well as Hirahara's much more sophisticated lower bound for Kt complexity [Hir20b] are instantiations of this approach.

To directly apply this approach to Kt complexity it is useful to define what we call the "critical threshold"  $\theta_{\Pi,t} \coloneqq |\Pi| + \lceil \log_2(t) \rceil$  of a given TM  $\Pi$  after t steps of its execution. We will assume that the decision problem MKtP can be worst-case decided by a TM  $\Pi_{\text{Kt}}$  in linear time. Then we construct a TM  $\Pi_{i}$  (using  $\Pi_{\text{Kt}}$  as a subroutine) that quickly outputs a Kt-random string z (i.e.,  $\text{Kt}(z) \ge |z|$ ). To reach a formal contradiction, our TM  $\Pi_{i}$  must in t steps produce a Kt-random string z that is strictly longer than the critical threshold  $\theta_{\Pi_{i},t}$ , i.e.,  $\theta_{\Pi_{i},t} \ge \text{Kt}(z) \ge |z| \ge \theta_{\Pi_{i},t}$  where the first inequality is by the definition of Kt complexity and the fact that  $\Pi$  outputs z in t steps, the second inequality is the Kt-randomness of z, and the last inequality is by assumption. (In this overview, we gloss over some minor definitional details that are rigorously taken care of in the formal proof.)

Black-box barrier. A conceptual problem to the algorithmic approach for a lower bound for MKtP is that we know little about the structure of the Kt-random strings  $R_{\text{Kt}} := \{x \mid \text{Kt}(x) \ge |x|\}$ . We say a TM  $\Pi_{\text{BB}}$  yields a contradiction in a black-box way, if given access to any set of potentially Kt-random strings  $R \ne \{0,1\}^*$  it produces a string  $z \notin R$  in t steps such that  $\theta_{\Pi_{\text{BB}},t} \le |z|$ . Intuitively, a potential TM  $\Pi_{\text{BB}}$ ignores the structure of the set R since it works for any arbitrary R. Such a  $\Pi_{\text{BB}}$  cannot exist because we can define  $R_{\Pi_{\text{BB}}} := \{0,1\}^* \setminus \{z \mid \Pi_{\text{BB}} \text{ queries } z \text{ to its oracle or outputs } z \text{ in } t \text{ steps and } \theta_{\Pi_{\text{BB}},t} \le |z|\}$ that breaks  $\Pi_{\text{BB}}$ . This black-box barrier explains why a lower bound for deterministic Kt is so hard to obtain (as opposed to randomized rKt<sup>5</sup>). So, for our algorithmic approach to succeed we need to exploit some property exhibited by the actual set of Kt-random strings  $R_{\text{Kt}}$  but not by any set  $R_{\Pi_{\text{BB}}}$ . Before we explain how, let us first present our rather simple strategy for a TM  $\Pi_4$ .

Our search strategy. As a first step we use the length-monotonic depth-first-search described in Fig. 1. The high-level idea is to traverse the binary tree of finite strings starting with the string 0.6 Whenever the *i*-th string  $z_i$  is visited our search algorithm TRAVERSE queries  $z_i$  to its oracle  $R_{\text{Kt}}$  and if  $z_i \in R_{\text{Kt}}$ descends to the next length with  $z_{i+1} \coloneqq z_i || 0$  (the left child of  $z_i$ ), otherwise it continues with the lexicographically next string of the same length  $z_{i+1} \coloneqq \text{next}(z_i)$  (the right neighbor of  $z_i$ ). See Fig. 2 for

<sup>&</sup>lt;sup>4</sup> Here, the error probabilities are not equal for both directions.

 $<sup>^5</sup>$  It is not even clear how  $R_{\Pi_{\sf BB}}$  would be defined for probabilistic  $\Pi_{\sf BB}.$ 

<sup>&</sup>lt;sup>6</sup> We choose to start with 0 instead of  $\varepsilon$  because it simplifies some edge cases.



**Fig. 1:** Our (simplified) traversal algorithm.

Fig. 2: Exemplary run of TRAVERSE: white strings are Kt-random.

an exemplary run of TRAVERSE. Crucially, the length of the visited strings is non-decreasing. We note that our TRAVERSE algorithm doesn't terminate and hence does not suffice for a proper contradiction (even if it visits a Kt-random string quickly enough). To actually reach a contradiction we have to a) construct a TM  $\Pi_{\text{TRA}}$  implementing TRAVERSE that at some point visits a string  $\check{z}$  within  $\hat{t}$  steps s.t.  $\theta_{\Pi_{\text{TRA}},\hat{t}} \leq |\check{z}|$ , and b) implement a mechanism s.t.  $\Pi_{\text{TRA}}$  also recognizes this fact—so that it can terminate and output  $\check{z}$ .

As a stepping stone it will be useful to see that TRAVERSE visits infinitely many different strings  $(z_i)_{i \in \mathbb{N}}$ . This follows from the existence of at least one Kt-random string of each length on which TRAVERSE descends to the next length. Moreover, we observe that TRAVERSE never "wraps around". That is TRAVERSE never reaches an all 1s string at the right border of the binary tree. Assuming an infinite (1-random) string s whose every prefix is Kt-random, this is also easy to see. Whenever TRAVERSE visits a prefix  $z_i = s_1 ||...||s_\ell$  it descends to the next string  $z_{i+1} \coloneqq s_1 ||...||s_\ell||0$ —thus always staying "to the left" of the infinite string s in the binary tree. Glossing over a minor technical issue, we can take Chaitin's constant  $\Omega$  (encoded in binary) to be such an infinite 1-random string. In fact, the 1-randomness of  $\Omega$  is the crucial information about the actual set of Kt-random strings  $R_{\rm Kt}$  that allows our TRAVERSE algorithm to sidestep the aforementioned black-box barrier.

Analysis. Next, we analyze the behavior of TRAVERSE to prove that after some  $\hat{t}$  steps TRAVERSE visits some Kt-random string  $z_i$  s.t.  $\theta_{\Pi_{\text{TRA},\hat{t}}} \leq |z_i|$ . Let  $Z \coloneqq \{z_i \mid i \in \mathbb{N}\}$  be the set of visited strings. Let  $i_{\ell} \coloneqq |Z \cap \{0,1\}^{\leq \ell}|$  be the number of strings visited of length at most  $\ell$ . Let  $Z_{\ell} \coloneqq Z \cap \{0,1\}^{\ell} = \{z_{i_{\ell-1}} \mid |0, ..., z_{i_{\ell}}\}$  be the set of visited strings of length exactly  $\ell$ . Let  $S_{\ell} \coloneqq \{z \in \{0,1\}^{\ell} \mid \text{int}(z) \geq (n_{i_{\ell-1}})\} \subset \{0,1\}^{\ell}$  be the lexicographical successors of  $Z_{\ell}$  (the right neighbors of  $Z_{\ell}$ ). Now, note that because TRAVERSE doesn't wrap around, it holds that  $Z_{\ell+1} \cup S_{\ell+1} = (\{z_{i_{\ell}}\} \cup S_{\ell}) \mid \{0,1\}$  and thus  $|Z_{\ell+1}| + |S_{\ell+1}| = 2|S_{\ell}| + 2$ . Let  $\gamma_{\ell} \coloneqq |Z_{\ell}|/|Z_{\ell} \cup S_{\ell}|$  be the fraction of strings of length  $\ell$  that TRAVERSE actually visits to the strings that it could potentially visit. By recursion the number of visited strings of length  $\ell$  can be expressed as  $|Z_{\ell}| = \gamma_{\ell} \sum_{\kappa=1}^{\ell} 2^{\kappa} \prod_{i=\ell-\kappa+1}^{\ell} (1-\gamma_i)$ . For our approach we'd like  $i_{\ell}$  and thus  $|Z_{\ell}|$  to be asymptotically small. An informal argument for this is that the formula for  $|Z_{\ell}|$  expresses a "self-limiting" behavior that emerges from our TRAVERSE algorithm. Namely, the faster  $\gamma_i$  goes to 0 the smaller  $|Z_{\ell}|$  because  $|Z_{\ell}|$  depends linearly on  $\gamma_{\ell}$ . On the other hand,  $|Z_{\ell}|$  depends on the product  $\prod_{i=j-\kappa+1}^{\kappa} (1-\gamma_i)$  which is closer to 1 the faster  $\gamma_i$  goes to 0. These antagonistic influences suggest there is some asymptotic rate of  $\gamma_i$  that leads to an asymptotically maximal  $|Z_{\ell}|$ . This behavior of  $|Z_{\ell}|$  can also be captured informally from the algorithmic view of TRAVERSE. Whenever  $|Z_{\ell-1}|$  is large this means that TRAVERSE moves far to the right, forcing the next  $|Z_{\ell}|$  to be small because only few strings remain to the right. In this manner, there cannot be many successive  $|Z_{\ell}|$  that are large. Turning back

to the more formal analysis, we can bound the number of visited strings of length at most  $\ell$  by

$$i_{\ell} \coloneqq \sum_{j=1}^{\ell} |Z_j| = \sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} \prod_{i=j-\kappa+1}^{j} (1-\gamma_i) \le \sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} e^{\sigma_{j-\kappa} - \sigma_j}$$
(1)

where  $\sigma_j \coloneqq \sum_{i=1}^{j} \gamma_i$ . Using the following technical lemma we can bound this quantity.

**Lemma 1 (Infinitely-often bound).** For any sequence  $(\gamma_j)_{j \in \mathbb{N}}$  with  $\gamma_j \in [0,1]$  and  $\sigma_\ell := \sum_{i=1}^{\ell} \gamma_i$  it holds that infinitely often  $\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} e^{\sigma_j - \kappa - \sigma_j} \leq_{io} 2^{\ell} / \ell \ln(\ell)$ .

The rigorous proof of Lemma 1 is contained in Section 6. This means that for infinitely many "critical" lengths  $\hat{\ell}$  when TRAVERSE visits the last string  $z_{i_{\hat{\ell}}} \in \{0,1\}^{\hat{\ell}} \cap R_{\mathrm{Kt}}$  it took at most  $\mathcal{O}(i_{\hat{\ell}} \cdot \hat{\ell})$  steps to do so—assuming MKtP  $\in$  DTIME[ $\mathcal{O}(n)$ ]. Recall that for any length  $\ell$  it holds that  $z_{i_{\ell}} \in R_{\mathrm{Kt}}$  because  $z_{i_{\ell}}$  is the last string of length  $\ell$  that TRAVERSE visits from which it descends to the next length. Now, if TRAVERSE were to output any such  $z_{i_{\hat{\ell}}}$ , then we would reach the contradiction

$$\hat{\ell} \le \operatorname{Kt}(z_{i_{\hat{\ell}}}) \le |\Pi_{\mathsf{TRA}}| + \left\lceil \log_2 \left( \mathcal{O}\left(i_{\hat{\ell}} \cdot \hat{\ell}\right) \right) \right\rceil \le \hat{\ell} - \ln \ln \left(\hat{\ell}\right) + \mathcal{O}(1) .$$
(2)

The missing piece is hence to construct a TM  $\Pi'_{\mathsf{TRA}}$  that implements TRAVERSE such that it is aware of its own critical threshold—so it knows when to output a string  $z_{i_{\ell}}$ . A generic approach is to simply simulate TRAVERSE with a  $\mathcal{O}(n \ln(n))$  slowdown. However, this would result in

$$\hat{\ell} \leq \operatorname{Kt}(z_{i_{\hat{\ell}}}) \leq |\Pi_{\mathsf{TRA}}| + \left\lceil \log_2 \left( \mathcal{O}\left(i_{\hat{\ell}} \cdot \hat{\ell} \cdot \ln\left(i_{\hat{\ell}} \cdot \hat{\ell}\right) \right) \right) \right\rceil \leq \hat{\ell} + \ln\left(\hat{\ell}\right) - \ln\ln\left(\hat{\ell}\right) + \mathcal{O}(1)$$
(3)

which does not suffice for a contradiction. Hence, we let  $\Pi'_{\mathsf{TRA}}$  count the size  $|Z_{\ell}|$  not one-by-one but only once it reaches a Kt-random string (the last string of each length). This way, each length  $\ell$  incurs an additive runtime overhead of  $\mathcal{O}(\ell)$  instead of  $\Omega(|Z_{\ell}|\ell \ln(\ell))$ . Due to space restrictions and because the details of the step-counting don't provide much conceptual insight, we defer these details to the formal proof of Theorem 1.

On Lemma 1. In the previous paragraph we have bounded the runtime of our contradicting TM  $\Pi'_{\text{TRA}}$ in terms of the number of visited strings  $i_{\ell}$  which in turn can be bounded by the term in Lemma 1. As alluded to earlier the specific term in Lemma 1 arises from the "self-limiting" behavior of our TRAVERSE algorithm. Recall that on the binary tree TRAVERSE only moves to the right neighbor or the left child of the current string. Fix some length  $\hat{\ell}$ . If for many of the previous lengths  $\ell \leq \hat{\ell}$  the TM TRAVERSE visited few strings, then the number  $i_{\hat{\ell}}$  of visited strings at length  $\hat{\ell}$  will be small by definition. On the other hand, if TRAVERSE visits many strings of length  $\ell$ , then TRAVERSE moves farther to the right, leaving fewer strings of subsequent lengths to be potentially visited. With this intuition in mind, it remains to prove Lemma 1 formally, though we defer the rigorous proof of Lemma 1 to Section 6. Instead, here we give a superficial sketch of our proof.

The basic idea is to prove Lemma 1 by contradiction, hence we may assume

$$\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa-\ell} e^{\sigma_{j-\kappa}-\sigma_j} \ge 1/\ell \ln(\ell)$$
(4)

for all  $\ell \in \mathbb{N}$ . We sum this inequality and bound the inner sum on the left-hand side of Eq. (4) by  $2^{j-\ell+1}$ (using the trivial inequality  $\sigma_j - \sigma_{j-\kappa} \ge 0$ ) to obtain a first lower bound for  $\sigma_{\ell}$ , i.e.,

$$2\sigma_{\hat{\ell}} \ge \sum_{\ell=1}^{\hat{\ell}} \sum_{j=1}^{\ell} \gamma_j 2^{j-\ell+1} \ge 2\sum_{\ell=1}^{\hat{\ell}} \frac{1}{\ell \ln(\ell)} \approx \int \frac{1}{\hat{\ell} \ln(\hat{\ell})} \mathrm{d}\hat{\ell} \in \Omega\left(\ln\ln\left(\hat{\ell}\right)\right) \,. \tag{5}$$

Here, we use the crucial property that the antiderivative of  $1/x \ln(x)$  is superconstant.

In the next steps we reuse the same strategy. Instead of bounding the inner sum on the left-hand

side of Eq. (4) trivially by  $2^{j-\ell+1}$  we use the stronger bound from Eq. (5) to obtain an even stronger bound  $\sigma_{\hat{\ell}} \in \Omega(\ln(\hat{\ell})^{1/17})$ . Reapplying the same strategy a third time finally yields the lower bound  $\sigma_{\hat{\ell}} \in \Omega(\ln(\hat{\ell})^3)$  which is strong enough to yield a contradiction to Eq. (4). In this brief sketch we glossed over many details and refer the interested reader to the formal proof in Section 6. However, a quick sanity check may be in order at this point. If  $\gamma_j \leq 1/j \ln(j)$ , then Lemma 1 holds trivially. Considering slightly larger  $\gamma_j \coloneqq \epsilon/j$  for any constant  $\epsilon > 0$  (thus  $\sigma_j \approx \epsilon \ln(j)$ ) yields  $\mathcal{O}(\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} e^{\sigma_{j-\kappa} - \sigma_j}) =$  $\mathcal{O}(\sum_{j=1}^{\ell} \frac{1}{j} \sum_{\kappa=1}^{j} 2^{\kappa} (1 - \kappa/j)^{\epsilon}) = \mathcal{O}(2^{\ell}/\ell^{1+\epsilon})$  which is also consistent with Lemma 1.

*Robustness.* While (unbounded) Kolmogorov complexity is quite robust against definitional changes (by invariance theorems), resource-bounded notions of complexity are more sensitive. There are many ways of defining Kt complexity formally: representative variations include [Hir20b; LP21b; RS22]. Our Definition 2 essentially corresponds to the one in [Hir20b]. In general, the notion of Kt complexity depends—aside from the underlying computational model—on whether

- the runtime is measure in terms of the number of steps of the *simulated* program  $\Pi$  (direct time) or the *simulating* universal machine  $\mathcal{U}$  (universal time),
- the witness program  $\Pi$  produces the entire string x (global compression) on the empty input or outputs the *i*-th bit on input bin(*i*) (local compression),
- the universal machine is "prefix-free" (Kt) or "plain" (Ct).

First, we state that currently our technique only works for prefix-free complexity because it requires an (infinite) strings whose prefixes are Kt-random, and we only know such a string for prefix-free complexity (there is no plain 1-random string; Section 6.1 in [DH10]). Though, conceivably one might find another way of arguing the "no-wrap-around" property of our search algorithm, to extend our result to plain Ct complexity.

Second, we remark that resource-bounded *universal time* complexity is a somewhat fragile notion that does not enjoy an invariance theorem. The reason is that one can always modify a universal machine to artificially run an arbitrary amount of time to increase the  $\lceil \log_2(\mathfrak{t}_{\mathcal{U}}(\Pi)) \rceil$  term arbitrarily, yet the machine remains universal. So, to remain a meaningful notion we only consider universal machines with at most logarithmic overhead.

All of our formal results are stated for the setting of global compression, universal time measurement and logarithmic simulation overhead. Each result can be adapted to other settings as follows: When considering direct time measurement or models with constant simulation overhead,<sup>7</sup> we can actually strenghen our lower bound to DTIME[ $\mathcal{O}(n^2)$ ] because (to reach a contradiction) our search algorithm saves a factor of roughly  $\ell$  on level  $\ell$  in runtime—which can be spend on an (assumed) more expensive DTIME[ $\mathcal{O}(\ell^2)$ ] solver for MKtP. Independently, if we consider local compression, we have to reduce the lower bound for global compression by a linear factor. The reason is that witness programs for global compression are required to run for at least |x| steps but witness programs for local compression are only required to run for  $\mathcal{O}(\log_2 |x|)$  steps. We summarize the instantiations of our main result in different settings in Fig. 3.

Limitations and stronger conditional lower bounds. As presented, our strategy using  $\Pi'_{\mathsf{TRA}}$  cannot (unconditionally) tolerate any errors because

- if  $\Pi'_{\mathsf{TRA}}$  obtains a false negative query response (false high complexity), then it outputs a string that is not actually Kt-random which does not violate the definition of Kt-randomness, and
- if  $\Pi'_{\mathsf{TRA}}$  obtains a false positive query response (false low complexity), then it may skip (from left to right) the separating line defined by Chaitin's 1-random constant, thus potentially increasing the runtime prohibitively.

 $<sup>^7</sup>$  E.g. random-access machines and Kolmogorov–Uspensky machines.

Lower bound	Global compression	Local compression
Direct time	$DTIME[\mathcal{O}(n^2)]$	$DTIME[\mathcal{O}(n)]$
Universal time $+$ const. overhead	$DTIME[\mathcal{O}(n^2)]$	$DTIME[\mathcal{O}(n)]$
Universal time $+ \log$ overhead	$DTIME[\mathcal{O}(n)]^{\ (*)}$	-

**Fig. 3:** Our lower bounds for several definitional variantions of Kt complexity. As a rule of thumb going from global to local, or from constant to logarithmic simulation overhead decreases the lower bound by a linear factor. Our results are formally stated for the setting (\*).

However, we can conditionally tolerate some (false negative) one-sided errors. For example, suppose  $\mathsf{MKtP} \in \mathsf{DTIME}[\mathcal{O}(n^2)]$  can be worst-case decided in quadratic time by a TM  $\Pi_{\mathrm{Kt},n^2}$ , and  $\mathsf{MKtP} \in \mathsf{Heur}_{0,\gamma_{\mathrm{fn}}}\mathsf{DTIME}[\mathcal{O}(n)]$  can be decided in linear time with false negative probability  $\gamma_{\mathrm{fn}}(n) \in \mathrm{o}(1/n^2)$  (and no false positives) by a TM  $\widetilde{\Pi}_{\mathrm{Kt},n}$ . Then we can construct a modified TM  $\Pi'_{\mathrm{TRA}}$  which for each visited string  $z_i$  first queries  $z_i$  to the quicker linear-time heuristic  $\widetilde{\Pi}_{\mathrm{Kt},n}$ . If  $\widetilde{\Pi}_{\mathrm{Kt},n}$  outputs  $\widetilde{b} = 0$  (high complexity),  $\Pi''_{\mathrm{TRA}}$  queries  $z_i$  to the slower  $\Pi_{\mathrm{Kt},n^2}$  to obtain the definitive answer  $b = 0 \iff z_i \in R_{\mathrm{Kt}}$ . If  $\widetilde{b} = 0 \wedge b = 0$ , then  $\Pi''_{\mathrm{TRA}}$  descends to  $z_{i+1} \coloneqq z_i || 0$ , otherwise  $z_{i+1} \coloneqq \mathrm{next}(z_i)$ . First, note that  $\Pi''_{\mathrm{TRA}}$  visits exactly the same set of strings (in the same order) as TRAVERSE. In contrast to the unconditional case, however, we find that whenever  $\Pi''_{\mathrm{TRA}}$  visits a string  $z_i$  of critical length  $\hat{\ell}$  it took at most

$$\mathcal{O}\left(\sum_{\ell=1}^{\hat{\ell}} |Z_{\ell}| \cdot \ell + 2^{\ell} \gamma_{\mathsf{fn}}(\ell) \cdot \ell^2\right) = \mathcal{O}\left(i_{\hat{\ell}} \cdot \hat{\ell} + 2^{\hat{\ell}} \gamma_{\mathsf{fn}}\left(\hat{\ell}\right) \cdot \hat{\ell}^2\right) \subseteq o\left(2^{\hat{\ell}}\right)$$
(6)

steps because at length  $\ell$  there are at most  $2^{\ell}\gamma_{fn}(\ell)$  strings on which  $\hat{H}_{Kt,n}$  gives a false negative answer.<sup>8</sup> Consequently, when  $\Pi''_{TRA}$  visits and outputs such a string  $z_{i_{\ell}}$  it yields the contradiction

$$\hat{\ell} \leq \operatorname{Kt}(z_{i_{\hat{\ell}}}) \leq |\Pi_{\mathsf{TRA}}| + \left\lceil \log_2\left(o\left(2^{\hat{\ell}}\right)\right) \right\rceil \leq \hat{\ell} - \omega(1) + \mathcal{O}(1) .$$

$$\tag{7}$$

On overcoming the limitations. First, we want to point out a curious effect reminiscent of Williams's algorithmic method where a computational upper bound implies another lower bound. Namely, any nontrivial *worst-case* upper bound for MKtP (Item 1 in Theorem 2 is false) implies an improved linear-time lower bound for MKtP with one-sided error (Item 2 in Theorem 2 is true).

Above we state that—at first glance—our approach cannot tolerate any errors unconditionally. In truth, our approach actually tolerates some false positive error, e.g.  $\gamma_{fp}(n) \leq 1/4n \ln(n+1)^2$ . Recall that the reason given above for not tolerating false positive errors is because then our algorithm might "skip" Chaitin's 1-random constant and thus the recursion formula  $Z_{\ell+1} \cup S_{\ell+1} = (\{z_{i_\ell}\} \cup S_\ell) || \{0, 1\}$  no longer holds. In turns outby Lemma 5that there are many Kt-random strings of each length, in fact, they have an arbitrary (constant) density. Thus, it is fine for our algorithm to skip some prefixes of 1-random strings in each length, because as long as we only skip a few, there will always be sufficiently many remaining "to the right" of our algorithm's current position.

**Corollary 1.** Let  $\gamma_{\mathsf{fp}}$  be a false positive error rate s.t.  $\sum_{\ell=1}^{\infty} \gamma_{\mathsf{fp}}(\ell) \leq 1,^9$  and let  $\gamma_{\mathsf{fn}}(n) \coloneqq 0$  be no false negative error. The Levin-Kolmogorov complexity cannot be decided in deterministic linear time even with some false positive error, i.e.,  $\mathsf{MKtP} \not\in \mathsf{Heur}_{\gamma_{\mathsf{fp}},\gamma_{\mathsf{fn}}}\mathsf{DTIME}[\mathcal{O}(n)].$ 

The requirement  $\sum_{\ell=1}^{\infty} \gamma_{\mathsf{fp}}(\ell) \leq 1$  is sufficient because in length  $\ell$  we skip up to  $2^{\ell} \gamma_{\mathsf{fp}}(\ell)$  strings. This means that at length  $\hat{\ell}$  we might have skipped up to  $\sum_{\ell=1}^{\hat{\ell}} 2^{\hat{\ell}-\ell} \cdot 2^{\ell} \gamma_{\mathsf{fp}}(\ell)$  strings (all strings of length  $\hat{\ell}$ 

<sup>&</sup>lt;sup>8</sup> Here, we naturally assume that  $2^{\ell} \gamma_{fn}(\ell)$  is non-decreasing.

<sup>&</sup>lt;sup>9</sup> That is, there exists some constant c such that  $\sum_{\ell=1}^{\infty} \gamma_{fp}(\ell) \leq c \leq 1$ .

that have a skipped string as a prefix) whereas we have at least  $2^{\hat{\ell}}(1-\hat{\epsilon})$  many (prefixes of) 1-random strings in length  $\hat{\ell}$  (for any  $\hat{\epsilon} \in \mathbb{R}_{\geq 0}$ ). Thus,  $\sum_{\ell=1}^{\infty} \gamma_{\mathsf{fp}}(\ell) \leq 1$  ensures that in each length there are always more prefixes of 1-random strings than are potentially skipped for  $\hat{\epsilon} \coloneqq (1-\sum_{\ell=1}^{\infty} \gamma_{\mathsf{fp}}(\ell))/2$ .

Another obvious question is whether our technique is capable of proving a worst-case bound beyond linear time. By an improved analysis of the proof of Lemma 1 we can push our bound to slightly superlinear time (resp. superquadratic for the corresponding settings in Fig. 3).

**Corollary 2.** The Levin–Kolmogorov complexity cannot be decided in deterministic slightly superlinear time in the worst-case, i.e.,  $\mathsf{MKtP} \notin \bigcup_{k \in \mathbb{N}} \mathsf{DTIME}[\prod_{i=0}^{k} \ln^{(i)}(n)]$  where  $\ln^{(i)}$  is the *i* times iterated logarithm (with  $\ln^{(0)}(n) := n$ ).

Corollaries 1 and 2 can be combined. The reason for this somewhat peculiar time bound  $\mathfrak{t}(n) := \prod_{i=0}^{k} \ln^{(i)}(n)$  is that the antiderivative (of its reciprocal)  $\int 1/\mathfrak{t}(x) dx = \ln^{(k+1)}(x) \in \omega(1)$  is superconstant which is the property that we need to get our proof of Lemma 1 going(compare to the previous paragraph on Lemma 1 and see the end of Section 6 for a sketch).

In contrast, going to some polynomial lower bound, i.e.,  $\mathsf{MKtP} \notin \mathsf{DTIME}[n^{1+\epsilon}]$  for some  $\epsilon > 0$  seems challenging. With our current proof strategy this would require a stronger version of Lemma 1 in the form of  $\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} / e^{\sigma_j - \sigma_{j-\kappa}} \leq 2^{\ell} / \ell^{1+\epsilon}$  for some  $\epsilon > 0$ . However, this cannot hold for arbitrary  $\gamma_j$  because of the following counter example:  $\gamma_j := \epsilon/j$  implies  $\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} / e^{\sigma_j - \sigma_{j-\kappa}} \in \Theta(2^{\ell} / \ell^{1+\epsilon})$ . Nonetheless, there is a possibility to achieve such a stronger bound by leveraging more structure of  $R_{\mathrm{Kt}}$ 

Nonetheless, there is a possibility to achieve such a stronger bound by leveraging more structure of  $R_{\rm Kt}$  to restrict the space of possible  $\gamma_j$  (as we did by integrating Chaitin's constant in our analysis). We hope that our new technique inspires further research into even better diagonalization approaches. For example, it could be that adding multiple 0's to a high-complexity string or moving more steps to the right on a low-complexity string might yield a better lower bound with an adapted analysis.

#### 4 Preliminaries

Notation. Real functions are usually denoted by Greek letters  $\gamma$ ,  $\theta$ ,  $\varepsilon$ , etc., while natural/bit functions by Fraktur script  $\mathfrak{t}$ ,  $\mathfrak{f}$ , etc. Languages are denoted by the uppercase letter L. The empty string is denoted by  $\varepsilon$ . Integers related to sizes are denoted by lowercase Latin letters n, m, c, while indices are denoted by i,  $j, k, \kappa$ . Strings are denoted by lowercase Latin letters x, y, z, etc. Turing machines (TM) are denoted by caligraphic letters  $\mathcal{U}, \mathcal{M}$  as well as  $\Pi$  for the code of a TM. Complexity classes are denoted in sans-serif letters  $\mathsf{P}, \mathsf{NP}, \mathsf{EXP}$ , etc.

For convenience we add framed boxes with explanations of relevant (in-)equalities.

Notation 1 (Functional inequalities). Let  $\mathfrak{f}, \mathfrak{g} : \mathbb{N} \to \mathbb{R}$  be two functions. We write

$$\mathfrak{f} \leq \mathfrak{g} \quad \Longleftrightarrow \forall n \in \mathbb{N}: \qquad \qquad \mathfrak{f}(n) \leq \mathfrak{g}(n) \tag{8}$$

$$\mathfrak{f} \leq_{\mathsf{abf}} \mathfrak{g} \Longleftrightarrow \exists n_0 \in \mathbb{N} \ \forall n \ge n_0 : \mathfrak{f}(n) \le \mathfrak{g}(n) \tag{9}$$

$$\mathfrak{f} \leq_{\mathsf{io}} \mathfrak{g} \iff \forall n_0 \in \mathbb{N} \ \exists n \ge n_0 : \mathfrak{f}(n) \le \mathfrak{g}(n) \tag{10}$$

when  $\mathfrak{f}$  is less or equal to  $\mathfrak{g}$  on all inputs, on all but finitely many inputs, or on infinitely many inputs. Note that

$$\mathfrak{f} \leq \mathfrak{g} \implies \mathfrak{f} \leq_{\mathsf{abf}} \mathfrak{g} \iff \overline{\mathfrak{f}} >_{\mathsf{io}} \mathfrak{g} \implies \mathfrak{f} \leq_{\mathsf{io}} \mathfrak{g} . \tag{11}$$

It may be that  $\mathfrak{g} \leq_{io} \mathfrak{f}$  while simultaneously  $\mathfrak{g} \geq_{io} \mathfrak{f}$ . Sometimes we abuse notation and write  $\mathfrak{f}(n) \leq_{abf} \mathfrak{g}(n)$  to mean  $(n \mapsto \mathfrak{f}(n)) \leq_{abf} (n \mapsto \mathfrak{g}(n))$ .

Notation 2 (Languages). Let  $L \subseteq \{0,1\}^*$ , then for any  $x \in \{0,1\}^*$  we use the abbreviated notation  $L(x) = 1 \iff x \in L$  and  $L(x) = 0 \iff x \notin L$ .

Notation 3 (Integers and strings). Let  $\operatorname{int} : \{0,1\}^* \to \mathbb{N} : x \mapsto 2^{|x|+1} + \sum_{i=1}^{|x|} 2^{i-1}x_i$  be the canonical lexicographical bijection between strings and integers. Let  $\operatorname{bin} \coloneqq \operatorname{int}^{-1}$  be its inverse operation. Let  $\operatorname{next}(x) \coloneqq \operatorname{bin}((\operatorname{int}(x)+1) \mod 2^{|x|}+2^{|x|})$  be the function that returns the lexicographically next string of the same length.

*Computational Model.* We present our result for Turing machines but they carry over to over computational models. We discuss this in more detail in Section 3.

We generally assume a Turing machine (TM) has fixed number of tapes, one of which is a read-only the input tape, one a write-only output tape, and the rest are read-write work tapes. This naturally extends to oracle machine with a dedicated oracle tape, although we will not need it in this work.

Let  $\mathcal{M}$  be a deterministic Turing machine (TM). For any  $x \in \{0,1\}^*$  denote by  $\mathcal{M}(x) \in \{0,1\}^* \cup \{\bot\}$ the content of the output tape after  $\mathcal{M}$  has entered a terminal state, or  $\bot$  if  $\mathcal{M}$  does not terminate on input x. In particular, if  $\mathcal{M}$  halts with a string  $y \in \{0,1\}^n$  then it must have run for at least n steps.

Throughout the paper let  $\mathcal{U}$  denote a prefix-free universal Turing machine (UTM). For any string  $\Pi \in \{0,1\}^*$  let  $\mathfrak{t}_{\mathcal{U}}(\Pi)$  be the (minimum) number of steps after which  $\mathcal{U}$  halts on input  $\Pi$ . We assume that  $\mathcal{U}$  simulates any given TM with (multiplicative) logarithmic overhead [HS66]. That is, there exists some universal constant  $c_{\mathcal{U}} \in \mathbb{N}$  such that if the TM encoded by  $\Pi$  halts in t steps on input  $\varepsilon$ , then  $\mathcal{U}$  halts on input  $\Pi$  in  $\mathfrak{t}_{\mathcal{U}}(\Pi) \leq c_{\mathcal{U}} t \log_2(t)$  steps. Let  $\mathfrak{t} : \mathbb{N} \to \mathbb{N}$  be a time bound. Let  $\mathsf{DTM}[\mathfrak{t}]$  be the set of deterministic TMs that halt within  $\mathfrak{t}(n)$  steps on inputs of length  $n \in \mathbb{N}$ . For any TM  $\mathcal{M}$  let  $L_{\mathcal{M}} \coloneqq \{x \in \{0,1\}^* \mid \mathcal{M}(x) = 1\}$  be its (characteristic) language. Throughout, we require a time bound  $\mathfrak{t}$  to be time-constructible, i.e., there exists a TM  $\mathcal{M}_{\mathfrak{t}} \in \mathsf{DTM}[\mathcal{O}(\mathfrak{t})]$  that computes  $\mathfrak{t}$ . Let

$$\mathsf{DTIME}[\mathfrak{t}] \coloneqq \left\{ L \subseteq \{0,1\}^* \mid \exists \mathcal{M} \in \mathsf{DTM}[\mathfrak{t}] : L = L_{\mathcal{M}} \right\}$$
(12)

be the class of languages decided by some DTM in time t. Let  $\mathsf{DTIME}[\mathcal{O}(\mathfrak{t})] \coloneqq \bigcup_{d \in \mathbb{N}} \mathsf{DTIME}[d \cdot \mathfrak{t}]$  be the class of languages decided by some DTM in time  $\mathcal{O}(\mathfrak{t})$ . In the following let C be some class of languages that is closed under intersection. Let

$$\operatorname{Heur}_{\gamma_{\mathsf{fp}},\gamma_{\mathsf{fn}}}\mathsf{C} \coloneqq \left\{ L \subseteq \left\{0,1\right\}^{*} \middle| \exists L' \in \mathsf{C} : \left| \begin{array}{c} \left(L \setminus L'\right) \cap \left\{0,1\right\}^{\lambda} \\ \left(L' \setminus L\right) \cap \left\{0,1\right\}^{\lambda} \\ \leq_{\mathsf{abf}} \gamma_{\mathsf{fn}}(\lambda)2^{\lambda} \\ \leq_{\mathsf{abf}} \gamma_{\mathsf{fn}}(\lambda)2^{\lambda} \end{array} \right\}$$
(13)

be the class of languages with a C-heuristic with false-positive error at most  $\gamma_{fp}$  and false-negative error at most  $\gamma_{fn}$ .

*Complexity Measures.* The most basic notion of Kolmogorov complexity is the length of the smallest program (witness program) that produces a given string w.r.t. some UTM.

**Definition 1** (Solomonoff–Kolmogorov–Chaitin complexity [Sol60; Kol63; Cha69]). Let  $\mathcal{U}$  be a (prefix-free) UTM. For any string  $x \in \{0,1\}^*$  we say

$$\mathbf{K}_{\mathcal{U}}(x) \coloneqq \min\{|\Pi| \mid \Pi \in \{0,1\}^* : \mathcal{U}(\Pi) = x\}$$

$$\tag{14}$$

is the (prefix-free) Kolmogorov complexity.<sup>10</sup>

While a powerful notion, it is not computable, hence Levin [Lev84] came up with an alternative definition which charges an additional logarithmic term for the runtime of the witness program that produces the given string.

**Definition 2 (Levin–Kolmogorov complexity [Lev84; Tra84]).** Let  $\mathcal{U}$  be a (prefix-free) UTM. For any string  $x \in \{0,1\}^*$  we say

$$\operatorname{Kt}_{\mathcal{U}}(x) \coloneqq \min\{|\Pi| + \lceil \log_2(t) \rceil \mid \Pi \in \{0,1\}^*, t \in \mathbb{N} : \mathcal{U}(\Pi) = x \wedge \mathfrak{t}_{\mathcal{U}}(\Pi) \le t\}$$
(15)

<sup>&</sup>lt;sup>10</sup> For brevity and in accord with the literature [LV08] we simply use the term "Kolmogorov complexity".

is the (prefix-free global) Levin–Kolmogorov complexity. Let

N

$$\mathsf{IKtP}_{\mathcal{U}} \coloneqq \{(y,k) \in \{0,1\}^m \times [m] \mid m \in \mathbb{N} : \mathrm{Kt}(y) \le k\}$$

$$\tag{16}$$

be the decisional minimum Kt-problem. This version is called "global compression" because the witness program outputs the entire string y.

For reference, we also define the "local compression" version where the witness program outputs each bit of the string y separately. For any string  $x \in \{0,1\}^*$  we say

$$\ddot{\mathrm{Kt}}_{\mathcal{U}}(x) \coloneqq \min\left\{ |\Pi| + \lceil \log_2(t) \rceil \middle| \begin{array}{c} \Pi \in \{0,1\}^*, t \in \mathbb{N} : \forall i \in \{1,...,|x|\} : \\ \mathcal{U}(\Pi,i) = x_i \wedge \mathfrak{t}_{\mathcal{U}}(\Pi,i) \le t \end{array} \right\}$$
(17)

is the (prefix-free local) Levin-Kolmogorov complexity.

We mainly focus on the global version and discuss various definitional subtleties in Section 3.

**Fact 1** (Relation between K and Kt). For any string  $x \in \{0,1\}^*$  it holds that  $Kt(x) \ge K(x) + \lceil \log_2(|x|) \rceil$ .

*Proof.* This is because even the shortest (global witness) program for x must run for at least |x| steps.<sup>11</sup>

**Definition 3 (1-/Martin-Löf-randomness ([DH10] referring to [Mar66; Lev74; Cha75])).** Let  $\mathcal{U}$  be a (prefix-free) UTM. An infinite sequence of bits  $w = (w_i)_{i \in \mathbb{N}}$  is called 1-random, iff there exists some constant  $\hat{c}_{\mathcal{U},w} \in \mathbb{N}$  such that for each  $n \in \mathbb{N}$  it holds that  $K(w_1||...||w_n) \ge n - \hat{c}_{\mathcal{U},w}$ .

Analogously, an infinite sequence of bits  $w = (w_i)_{i \in \mathbb{N}}$  is called 1-Kt-random, iff there exists some constant  $\hat{c}_{\mathcal{U},w} \in \mathbb{N}$  such that for each  $n \in \mathbb{N}$  it holds that  $\operatorname{Kt}(w_1||...||w_n) \ge n + \lceil \log_2(n) \rceil - \hat{c}_{\mathcal{U},w}$ .

Going forward we fix some arbitrary UTM  $\mathcal{U}$  but omit it in our notation and simply write K, Kt, MKtP, etc. By Fact 1 K-randomness implies Kt-randomness.

**Fact 2** (Chaitin's  $\Omega$  constant is 1-random [Cha75]). Let  $\Omega_i$  be the *i*-th bit of Chaitin's constant [Cha75] in binary representation. Then the sequence  $\Omega = (\Omega_i)_{i \in \mathbb{N}}$  is 1-random and thus 1-Kt-random with constant  $\hat{c}_{\Omega}$ .

#### 5 Formal Results

**Lemma 2.** The algorithm TRAVERSE<sub> $\hat{c}_{Q,CKt}$ </sub> in Fig. 4 visits infinitely many different strings  $(z_i)_{i\in\mathbb{N}}$ .

Proof. First note that the lengths of  $z_i$  are non-decreasing, i.e.,  $|z_{i+1}| \ge |z_i|$ . Suppose for contradiction that there exists some maximal length  $\hat{\ell} \in \mathbb{N}$  such that all strings  $|z_i| \le \hat{\ell}$  for all  $i \in \mathbb{N}$ . By inspection it is apparent that from some point onward  $\operatorname{TRAVERSE}_{\hat{c}_{\Omega,C_{\mathrm{Kt}}}}$  cycles through all strings of length  $\hat{\ell}$ . Because the prefixes of Chaitin's constant  $\Omega = (\Omega_i)_{i\in\mathbb{N}}$  are a 1-Kt-random sequence, for each length  $\ell \in \mathbb{N}$  the string  $\Omega_1 ||...|| \Omega_{\ell} \in \{0,1\}^{\ell}$  has complexity  $\operatorname{Kt}(\Omega_1 ||...|| \Omega_{\ell}) \ge \operatorname{K}(\Omega_1 ||...|| \Omega_{\ell}) \ge \ell + \lceil \log_2(\ell) \rceil - \hat{c}_{\Omega}$ . Thus, once  $\operatorname{TRAVERSE}_{\hat{c}_{\Omega,C_{\mathrm{Kt}}}}$  visits the string  $z_i = \Omega_1 ||...|| \Omega_{\hat{\ell}}$  the next string is  $z_{i+1} = \Omega_1 ||...|| \Omega_{\hat{\ell}} ||0$  due to line 3. This contradicts  $\hat{\ell} + 1 = |z_i| + 1 \le \hat{\ell}$ .

Now, we prove our main result.

**Theorem 1.** The Levin–Kolmogorov complexity cannot be decided in deterministic linear time in the worst-case, i.e.,  $\mathsf{MKtP} \notin \mathsf{DTIME}[\mathcal{O}(n)]$ .

<sup>&</sup>lt;sup>11</sup> Here, we presume the global compression version of Kt, and add  $\lfloor \log_2(|x|) \rfloor$ .

*Proof.* The intuition of this proof is already outlined in Section 3. The high-level idea is to assume that Kt can be computed quickly, and then construct a sufficiently fast TM that produces a highly complex string. This then contradicts the definition of a complex string needing a large or slow program to compute. Key properties of our constructed TM is that it finds a complex string sufficiently fast and that the TM is aware of its own runtime. For the latter property we use a counter variable in our TM to upper bound its runtime by counting the number of visited strings of a given length. This counter needs to be larger than the actual runtime of the TM (Claim 1) so it is guaranteed to output a string larger than its own critical threshold. On the other hand, the counter must not be too large (Claim 3), for otherwise it would not output critical strings that would actually suffice for a contradiction.

Suppose for contradiction MKtP  $\in$  DTIME[ $\mathcal{O}(n)$ ], then there exists some  $c_{\mathrm{Kt}} \in \mathbb{N}$  such that MKtP  $\in$  DTIME[ $2^{c_{\mathrm{Kt}}}n$ ], i.e., there exists some TM  $\Pi_{\mathrm{Kt}}$  that decides  $\mathrm{Kt}(z) \leq k$  in time at most  $\mathfrak{t}(n + \lceil \log_2(n) \rceil) \leq \mathfrak{t}(2n) \coloneqq 2^{c_{\mathrm{Kt}}+1}n$  on any instance  $(z,k) \in \{0,1\}^n \times [n]$ . Later, we will choose a sufficiently large  $c_{\mathrm{Kt}}$ . Fix the constant  $\hat{c}_{\Omega}$  from Fact 2. For any  $c_{\mathrm{Kt}}$  let  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  be the smallest TM implementing the TRAVERSE $_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  algorithm from Fig. 4. There exists some universal<sup>12</sup> constant  $c_{\mathrm{fix}} \in \mathbb{N}$  such that for any integer  $c_{\mathrm{Kt}} \in \mathbb{N}$  the TM  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  has size  $|\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}| \leq c_{\mathrm{fix}} + 2\lfloor \log_2(c_{\mathrm{Kt}}) + 1\rfloor$  by storing  $c_{\mathrm{Kt}}$  prefix-free. In particular, for any  $c_{\mathrm{Kt}} \geq 2(c_{\mathrm{fix}} + c_{\mathcal{U}}) + 8$  the TM's size is bounded by  $|\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}| \leq c_{\mathrm{Kt}} - c_{\mathcal{U}}$ 

(recall that  $c_{\mathcal{U}}$  is the universal simulation constant). We derive a contradiction through a series of claims about the TM  $\mathcal{M}_{\hat{c}_{\Omega}, c_{\mathrm{Kt}}}$ . The TM  $\mathcal{M}_{\hat{c}_{\Omega}, c_{\mathrm{Kt}}}$  visits the sequence  $(z_i)_{i \in \mathbb{N}}$  of strings. Let  $Z \coloneqq \{z_i \mid i \in \mathbb{N}\}$ . For each length  $\ell \in \mathbb{N}$  let  $Z_{\ell} \coloneqq Z \cap \{0, 1\}^{\ell} = \{\hat{z}_{\ell}, ..., \check{z}_{\ell}\}$  where  $\hat{z}_{\ell}$  and  $\check{z}_{\ell}$  are the lexicographically first resp. last string in  $Z_{\ell}$ . Our first claim establishes that—whenever  $\mathcal{M}_{\hat{c}_{\Omega}, c_{\mathrm{Kt}}}$  checks whether to output a visited string in line 10—its

use the variable  $t_{\ell}$  to effectively bound its own critical threshold. **Claim 1** (Time counter lower bound). For any length  $\ell$  let  $\tilde{t}_{\ell}$  be the number of steps that  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  takes to reach line 10 with length  $\ell$ . It holds that  $t_{\ell} \geq \tilde{t}_{\ell}$ .

variable  $t_{\ell}$  is larger than the number of steps that  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  took so far. This means that  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  can

*Proof.* First, under the assumption  $\mathsf{MKtP} \in \mathsf{DTIME}[2^{c_{\mathsf{Kt}}}n]$  we argue that  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathsf{Kt}}}$  takes at most  $c_{\mathsf{Kt}}^3 \ell^2$  steps to execute line 9. Our first goal is to bound the time needed to execute line 9. First, we recursively bound the variable  $t_\ell$  by

$$t_{\ell} \coloneqq t_{\ell-1} + (\operatorname{int}(z_i) - \operatorname{int}(\hat{z}_{\ell}) + 2)2^{c_{\mathrm{Kt}} + 1}\ell + 2^{2c_{\mathrm{Kt}} + \ell - \lceil \log_2(\ell) \rceil + 1}$$
(18)

$$= t_{\ell-1} + 2^{c_{\mathrm{Kt}}+2} |Z_{\ell}| \ell + 2^{2c_{\mathrm{Kt}}+\ell - \lceil \log_2(\ell) \rceil + 1}$$
(19)

$$\leq t_{\ell-1} + 2^{4\ell + 4c_{\rm Kt}} \tag{20}$$

for sufficiently large  $c_{\text{Kt}}$ . Resolving this recursive upper bound for sufficiently large  $c_{\text{Kt}}$  it follows that  $t_{\ell} \leq 2^{4\ell+4c_{\text{Kt}}} + t_0$  where  $t_0 \coloneqq 0$ .

Now, the value  $t_{\ell}$  can be computed by simple arithmetic (addition, multiplication) and bit shifting operations taking at most quadratic time in the maximal bit length  $\mathcal{O}(\ell)$  of the operands  $\ell$ ,  $t_{\ell}$  int $(\hat{z}_{\ell})$ , int $(z_i) = \operatorname{int}(\check{z}_{\ell})$  and  $c_{\mathrm{Kt}}$ . That means there is some  $c' \in \mathbb{N}$  (independent of  $c_{\mathrm{Kt}}$ ) such that  $t_{\ell}$  can be computed in time  $c' \log_2(t_{\ell})^2 \leq c'(4\ell + 4c_{\mathrm{Kt}})^2 \leq c_{\mathrm{Kt}}^3 \ell^2$  for sufficiently large  $c_{\mathrm{Kt}}$ .

Taking a step back we observe that the TM  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  takes at most  $\widetilde{\Delta}_{\ell} \coloneqq \widetilde{t}_{\ell} - \widetilde{t}_{\ell-1}$  actual steps to iterate over the strings  $Z_{\ell}$  of length  $\ell$  (lines 6 through 13). We see through

$$\widetilde{\Delta}_{\ell} \coloneqq \widetilde{t}_{\ell} - \widetilde{t}_{\ell-1} \tag{21}$$

$$\leq \underbrace{|Z_{\ell}|(2^{c_{Kt}+1}\ell+4\ell)}_{\text{steps for } Z_{\ell}\setminus\{\tilde{z}_{\ell}\}\text{ in lines 6 and 15}} + \underbrace{2^{c_{Kt}+1}\ell+4\ell+c_{Kt}^{3}\ell^{2}+2^{c_{Kt}+1}\ell+c_{Kt}\ell+4\ell}_{\text{steps for } \tilde{z}_{\ell}\text{ in lines 6-13}}$$
(22)

$$\leq (2^{c_{\rm Kt}+1}+4)|Z_{\ell}|\ell + (2^{c_{\rm Kt}+2}+8+c_{\rm Kt}^3+c_{\rm Kt})\ell^2$$
(23)

 $\overline{12}$  Independent of  $c_{\rm Kt}$ .

 $\text{TRAVERSE}_{\hat{c}_{\Omega}, c_{\text{Kt}}}$ 

 $1: z_1 \coloneqq 0 \in \{0,1\}^*$  $2: \quad t_0 := 0 \in \mathbb{N}$  $3: \quad \ell\coloneqq 1\in\mathbb{N}$ 4:  $\hat{z}_1 \coloneqq 1 \in \{0,1\}^*$ 5: for  $i \in \mathbb{N}_{>1}$ if  $\operatorname{Kt}(z_i) \ge \ell + \lceil \log_2(\ell) \rceil - \hat{c}_{\Omega} // \text{ in } 2^{c_{\operatorname{Kt}}+1} \ell \text{ steps}$ 6:  $z_{i+1} := z_i || 0 \in \{0, 1\}^{\ell+1}$  // in 4 $\ell$  steps 7: $\hat{z}_{\ell+1} \coloneqq z_{i+1} \in \{0,1\}^{\ell+1}$  // store the starting node of length  $\ell+1$ 8:  $\# \ \ \, \inf c_{\mathrm{Kt}}^3 \ell^2 \ \, \text{steps} \ \, / \ \, \text{add time spend} \\ \ \ \, \text{on length} \ \, \ell \ \, \text{plus safety margin for} \\ \ \ \, \text{increasing the counter itself}$  $t_{\ell} := t_{\ell-1} + (\operatorname{int}(z_i) - \operatorname{int}(\hat{z}_{\ell}) + 1)2^{c_{\mathrm{Kt}}+2}\ell + 2^{2c_{\mathrm{Kt}}+\ell - \lceil \log_2(\ell) \rceil + 1}$ 9:  $\text{if } \operatorname{Kt}(z_i) \geqq c_{\operatorname{Kt}} + \lceil \log_2(t_\ell \log_2(t_\ell)) \rceil \quad /\!\!/ \begin{array}{c} \inf c_{\operatorname{Kt}} \ell + 2^{c_{\operatorname{Kt}}+1} \ell \text{ steps for computing } c_{\operatorname{Kt}} + \frac{1}{\lceil \log_2(t_\ell \log_2(t_\ell)) \rceil} \text{ and deciding MKtP} \end{array}$ 10:return  $z_i$ 11: endif 12: $\ell \coloneqq \ell + 1 \quad // \text{ in } 4\ell \text{ steps}$ 13:else 14:  $z_{i+1} \coloneqq \operatorname{next}(z_i) \in \{0,1\}^{\ell} \quad /\!\!/ \text{ in } 4\ell \text{ steps}$ 15:endif 16:17: endfor

Fig. 4: Our search algorithm with runtime bounds under the assumption  $\mathsf{MKtP} \in \mathsf{DTIME}[2^{c_{\mathsf{Kt}}}n]$ . The parameters  $\hat{c}_{\Omega}, c_{\mathsf{Kt}} \in \mathbb{N}$  are hardcoded. It might not be obvious why line 9 can be executed in  $c_{\mathsf{Kt}}^3 \ell^2$  steps, the reason is fleshed out in the proof of Claim 1.

$$\leq 2^{c_{\mathrm{Kt}}+2} |Z_{\ell}| \ell + 2^{2c_{\mathrm{Kt}}+\ell-\lceil \log_2(\ell) \rceil+1}$$

$$= \Delta_{\ell}$$
(24)
(25)

that the variable  $t_{\ell}$  grows more quickly than  $\tilde{t}_{\ell}$  and since  $t_0 = 0 = \tilde{t}_0$ , it follows that  $t_{\ell} \geq \tilde{t}_{\ell}$  for any  $\ell \in \mathbb{N}$ .

Claim 2 (Non-termination). The TM  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  never halts, thus TRAVERSE $_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  never halts.

*Proof.* If  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  halted and produced a string  $\hat{z} \in \{0,1\}^{\hat{\ell}}$  within  $\tilde{t}_{\hat{\ell}}$  steps, then by definition of the (prefix-free global) Levin–Kolmogorov complexity

$$\operatorname{Kt}(\hat{z}) \leq |\mathcal{M}_{\hat{c}_{\Omega}, c_{\mathrm{Kt}}}| + \lceil \log_2(\mathfrak{t}_{\mathcal{U}}(\mathcal{M}_{\hat{c}_{\Omega}, c_{\mathrm{Kt}}})) \rceil$$
(26)

$$\leq |\mathcal{M}_{\hat{c}_{\Omega}, c_{\mathrm{Kt}}}| + \left\lceil \log_2 \left( c_{\mathcal{U}} \tilde{t}_{\hat{\ell}} \log_2(\tilde{t}_{\hat{\ell}}) \right) \right\rceil \tag{27}$$

$$\leq c_{\rm Kt} - c_{\mathcal{U}} + \left\lceil \log_2 \left( c_{\mathcal{U}} \tilde{t}_{\hat{\ell}} \log_2(\tilde{t}_{\hat{\ell}}) \right) \right\rceil \tag{28}$$

$$\leq c_{\rm Kt} + \left\lceil \log_2(\tilde{t}_{\hat{\ell}} \log_2(\tilde{t}_{\hat{\ell}})) \right\rceil \tag{29}$$

$$\leq c_{\rm Kt} + \left\lceil \log_2(t_{\hat{\ell}} \log_2(t_{\hat{\ell}})) \right\rceil \tag{30}$$

by the fact that  $|\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}| \leq c_{\mathrm{Kt}} - c_{\mathcal{U}}$  and Claim 1. However, the only way  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  returns a string is in line 11, thus the condition in line 10 must be fulfilled, namely  $\mathrm{Kt}(\hat{z}) \geq c_{\mathrm{Kt}} + \lceil \log_2(t_{\hat{\ell}} \log_2(t_{\hat{\ell}})) \rceil$ . This contradicts Eq. (26). Consequently, under the hypothesis  $\mathsf{MKtP} \in \mathsf{DTIME}[2^{c_{\mathsf{Kt}}}n]$  the TM  $\mathcal{M}_{\hat{c}_{\Omega}, c_{\mathsf{Kt}}}$  never halts.

Because of Claim 2 the TM  $\mathcal{M}_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  visits the same sequence of strings  $(z_i)_{i\in\mathbb{N}}$  as TRAVERSE $_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$ . For any length  $\ell$  let  $i_{\ell} := \sum_{j=1}^{\ell} |Z_j|$  be number of visited string of length at most  $\ell$ .

Claim 3 (Time counter upper bound). For any length  $\ell$  it holds that  $t_{\ell} \leq 2^{2c_{\mathrm{Kt}}+4}(i_{\ell}\ell+2^{\ell}/\ell)$ .

*Proof.* Using Eqs. (18) and (25) we can bound the telescope sum

$$t_{\ell} = t_0 + \sum_{j=1}^{\ell} \Delta_j \tag{31}$$

$$= \sum_{j=1}^{\ell} \left( 2^{c_{\mathrm{Kt}}+2} |Z_j| j + 2^{2c_{\mathrm{Kt}}+j-\lceil \log_2(j) \rceil+1} \right)$$
 Eq. (25)

$$\leq 2^{c_{\mathrm{Kt}}+2} \ell \left( \sum_{j=1}^{\ell} |Z_j| \right) + 2^{2c_{\mathrm{Kt}}+3+\ell}/\ell$$
(33)

$$\leq 2^{2c_{\rm Kt}+4} (i_\ell \ell + 2^\ell / \ell)$$
 (35)

Now we have upper bounded the counter variable  $t_{\ell}$  in terms of the number  $i_{\ell}$  of visited strings of length at most  $\ell$ . It remains to argue that for infinitely many  $\ell$  the value  $i_{\ell}$  is sufficiently small, to reach a contradiction. To this end, we will reexpress  $i_{\ell}$  in a different form. Let  $S_{\ell} \subset \{0, 1\}^{\ell}$  be the lexicographical successors of  $Z_{\ell}$  (the right neighbors of  $Z_{\ell}$ ). Now, note that because TRAVERSE<sub> $\hat{c}\Omega, c_{\text{Kt}}$ </sub> doesn't wrap around (staying to the left of Chaitin's constant), it holds that  $Z_{\ell+1} \cup S_{\ell+1} = (\{z_{i_{\ell}}\} \cup S_{\ell})||\{0, 1\}$  and thus  $|Z_{\ell+1}|+|S_{\ell+1}| = 2|S_{\ell}|+2$ . Let  $\gamma_{\ell} := |Z_{\ell}|/|Z_{\ell} \cup S_{\ell}|$  be the fraction of strings of length  $\ell$  that TRAVERSE<sub> $\hat{c}\Omega, c_{\text{Kt}}$ </sub> actually visits to the strings that it could potentially visit. Using this expression for  $\gamma_{\ell}$  we can rewrite the previous equality as a recursive formula for  $|S_{\ell}|$  (depending on  $\gamma_{\ell}$ ), i.e.,

$$2|S_{\ell}| + 2 = |Z_{\ell+1}| + |S_{\ell+1}| \tag{36}$$

$$= (|Z_{\ell+1}| + |S_{\ell+1}|)\gamma_{\ell+1} + |S_{\ell+1}|$$
(37)

$$= 2(|S_{\ell}| + 1)\gamma_{\ell+1} + |S_{\ell+1}| \tag{38}$$

$$\Rightarrow |S_{\ell+1}| = 2(|S_{\ell}| + 1)(1 - \gamma_{\ell+1})$$
(39)

By solving this recursion with  $|S_1| \coloneqq 1$  we can express the number of successor strings as

$$|S_{\ell}| = \sum_{\kappa=1}^{\ell} 2^{\kappa} \prod_{i=\ell-\kappa+1}^{\ell} (1-\gamma_i) .$$
(40)

In turn, we can use the definition of  $\gamma_{\ell}$  to express the number of visited strings of length exactly  $\ell$  as

$$|Z_{\ell}| = 2(|S_{\ell-1}| + 1)\gamma_{\ell} = \gamma_{\ell} \sum_{\kappa=1}^{\ell} 2^{\kappa} \prod_{i=\ell-\kappa+1}^{\ell-1} (1-\gamma_i) .$$
(41)

Lastly, we can sum over all lengths to obtain

$$i_{\ell} \coloneqq \sum_{j=1}^{\ell} |Z_j| \tag{42}$$

$$=\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} \prod_{i=j-\kappa+1}^{j} (1-\gamma_i)$$
(43)

$$=\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} e^{\sum_{i=j-\kappa+1}^{j} \ln(1-\gamma_i)}$$
(44)

$$\leq \sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} e^{-\sum_{i=j-\kappa+1}^{j} \gamma_i} \qquad \qquad \boxed{\ln(1-x) \leq -x}$$
(45)

$$=\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} e^{\sigma_{j-\kappa} - \sigma_j}$$

$$\tag{46}$$

where  $\sigma_{\ell} \coloneqq \sum_{i=1}^{\ell} \gamma_i$ . This expression is bounded by Lemma 1.

Conclusion. Using Lemma 1 let  $\hat{\ell} \geq e^{2^{3c_{\mathrm{Kt}}+\hat{c}_{\Omega}+6}}$  be an arbitrarily large integer such that  $i_{\hat{\ell}} \leq 2^{\hat{\ell}}/\hat{\ell}\ln(\hat{\ell})$ . Let  $z_{i_{\hat{\ell}}} \in \{0,1\}^{\hat{\ell}}$  be the last string of length  $\hat{\ell}$  visited by TRAVERSE $_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$ . Because  $z_{i_{\hat{\ell}}}$  is the last string of length  $\hat{\ell}$  the condition  $\mathrm{Kt}(z_{i_{\hat{\ell}}}) \geq |z_{i_{\hat{\ell}}}| + \lceil \log_2(|z_{i_{\hat{\ell}}}|) \rceil - \hat{c}_{\Omega} = \hat{\ell} + \lceil \log_2(\hat{\ell}) \rceil - \hat{c}_{\Omega}$  in line 6 in TRAVERSE $_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  must be fulfilled. Moreover, because TRAVERSE $_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  never halts—according to Claim 2—the violated return condition in line 10 in TRAVERSE $_{\hat{c}_{\Omega},c_{\mathrm{Kt}}}$  dictates  $\mathrm{Kt}(z_{i_{\hat{\ell}}}) \leq c_{\mathrm{Kt}} + \lceil \log_2(t_{\hat{\ell}}\log_2(t_{\hat{\ell}})) \rceil$ . Thus we arrive at the contradiction

$$\hat{\ell} + \left| \log_2(\hat{\ell}) \right| - \hat{c}_{\Omega} \lneq \operatorname{Kt}(z_{i_{\hat{\ell}}})$$

$$\tag{47}$$

$$\leq c_{\rm Kt} + \left\lceil \log_2(t_{\hat{\ell}} \log_2(t_{\hat{\ell}})) \right\rceil \tag{48}$$

$$\leq c_{\rm Kt} + \left\lceil \log_2(t_{\hat{\ell}}) \right\rceil + \log_2(\hat{\ell}) \tag{49}$$

$$\leq c_{\mathrm{Kt}} + \left[ (2c_{\mathrm{Kt}} + 4) + \log_2 \left( i_{\hat{\ell}} \hat{\ell} + 2^{\hat{\ell}} / \hat{\ell} \right) \right] + \log_2 \left( \hat{\ell} \right) \qquad \qquad \boxed{\mathrm{Claim } 3} \tag{50}$$

$$\leq c_{\rm Kt} + \left| (2c_{\rm Kt} + 4) + \log_2 \left( 2^{\hat{\ell}} / \ln(\hat{\ell}) + 2^{\hat{\ell}} / \hat{\ell} \right) \right| + \log_2 \left( \hat{\ell} \right) \qquad \text{Lemma 1} \quad (51)$$

$$= c_{\mathrm{Kt}} + 2c_{\mathrm{Kt}} + 6 + \hat{\ell} - \log_2 \ln\left(\hat{\ell}\right) + \log_2\left(\hat{\ell}\right)$$

$$(53)$$

$$\leq c_{\rm Kt} + 2c_{\rm Kt} + 6 + \hat{\ell} - (3c_{\rm Kt} + \hat{c}_{\Omega} + 6) + \log_2(\hat{\ell}) \qquad \qquad \hat{\ell} \geq e^{2^{3c_{\rm Kt} + \hat{c}_{\Omega} + 6}} \tag{54}$$

$$=\hat{\ell} + \log_2\left(\hat{\ell}\right) - \hat{c}_{\Omega} . \tag{55}$$

The proof of Theorem 1 relativizes. By adapting it we can show analogous results for various definitions of Kt complexity.

**Corollary 3.** Assume a computational model with constant universal simulation overhead, e.g. randomaccess machines. The (global compression) Levin–Kolmogorov complexity cannot be decided in deterministic quadratic time in the worst-case, i.e.,  $\mathsf{MKtP} \notin \mathsf{DTIME}[\mathcal{O}(n^2)]$ . The (local compression) Levin– Kolmogorov complexity cannot be decided in deterministic linear time in the worst-case, i.e.,  $\mathsf{MKtP} \notin \mathsf{DTIME}[\mathcal{O}(n)]$ .

Next, we prove our conditional lower bounds.

**Theorem 2.** For each time bound  $\mathfrak{t}(n) \geq n$  at least one of the following is true:

- 1. MKtP  $\notin$  DTIME[ $\mathfrak{t}$ ],
- 2. MKtP  $\notin \operatorname{Heur}_{\gamma_{\mathsf{fp}},\gamma_{\mathsf{fn}}}\mathsf{DTIME}[\mathcal{O}(n)]$  with no false positive error  $\gamma_{\mathsf{fp}}(n) \coloneqq 0$  and false negative error  $\gamma_{\mathsf{fn}}(n) \coloneqq \frac{1}{2n\mathfrak{t}(2n)} \frac{2}{2^n}$ ,

 $\mathrm{TRAVERSE}'_{\hat{c}_{\Omega}, c_{\mathrm{Kt}}, \Pi_{\mathfrak{t}}, \Pi_{\mathrm{Kt}}, \widetilde{\Pi}_{\mathrm{Kt}}}$ 

1:  $z_1 \coloneqq 0 \in \{0, 1\}^*$ 2:  $t_0 \coloneqq 0 \in \mathbb{N}$  $3: \quad \ell \coloneqq 1 \in \mathbb{N}$ 4:  $\hat{z}_1 \coloneqq 1 \in \{0,1\}^*$ for  $i \in \mathbb{N}_{>1}$ 5: $b \coloneqq \widetilde{\Pi}_{\mathrm{Kt}}(z_i, \ell + \lceil \log_2(\ell) \rceil - \hat{c}_{\Omega})$  // in  $2^{c'+1}\ell$  steps / quick error-prone check 6: if b = 17: $b \coloneqq \Pi_{\mathrm{Kt}}(z_i, \ell + \lceil \log_2(\ell) \rceil - \hat{c}_{\Omega})$  // in  $\mathfrak{t}(2\ell)$  steps / slower exact check 8: fi 9: if b = 0 // assert  $\operatorname{Kt}(z_i) \ge \ell + \lceil \log_2(\ell) \rceil - \hat{c}_{\Omega}$ 10:  $z_{i+1} \coloneqq z_i || 0 \in \{0,1\}^{\ell+1} \quad /\!\!/ \text{ in } 4\ell \text{ steps}$ 11:  $\hat{z}_{\ell+1} \coloneqq z_{i+1} \in \{0,1\}^{\ell+1}$  # store the starting node of length  $\ell+1$ 12: $t'_{\ell} \coloneqq t_{\ell-1} + (\operatorname{int}(z_i) - \operatorname{int}(\hat{z}_{\ell}) + 2)2^{c_{\mathrm{Kt}}+1}\ell + 2^{2c_{\mathrm{Kt}}+\ell - \lceil \log_2(\ell) \rceil + 1} \quad \text{ } \# \underset{\ell \text{ plus safety margin for increasing the spend on length}}{\ell}$ 13:counter itself  $\mathbf{if} \quad \Pi_{\mathrm{Kt}}(z_i, c_{\mathrm{Kt}} + \lceil \log_2(t_\ell \log_2(t_\ell)) \rceil) = 0 \quad /\!\!/ \text{ in } c_{\mathrm{Kt}}\ell + \mathfrak{t}(2\ell) \text{ steps / assert } \mathrm{Kt}(z_i) \geq c_{\mathrm{Kt}} + \lceil \log_2(t_\ell \log_2(t_\ell)) \rceil$ 14: return  $z_i$ 15:endif 16 :  $\ell := \ell + 1 \quad /\!\!/ \text{ in } 4\ell \text{ steps}$ 17:else 18: $z_{i+1} \coloneqq \operatorname{next}(z_i) \in \{0,1\}^{\ell} \quad /\!\!/ \text{ in } 4\ell$ 19:endif 20 : 21: endfor

Fig. 5: Our search algorithm with runtime bounds under the assumption MKtP  $\notin$  DTIME[t] and MKtP  $\notin$  Heur<sub> $\gamma_{f_p}, 0$ </sub>DTIME[ $\mathcal{O}(n)$ ] where t is assumed to be time-constructible. The parameters  $\hat{c}_{\Omega}, c_{Kt} \in \mathbb{N}$ , the TM  $\Pi_t$  computing t, the t-time TM  $\Pi_{Kt}$  and the linear-time TM  $\widetilde{\Pi}_{Kt}$  are hardcoded. Changes to Fig. 4 are marked in gray.

*Proof.* This proof is a slight modification of the proof of Theorem 1, thus we only include the relevant changes. For contradiction assume  $\mathsf{MKtP} \in \mathsf{DTIME}[\mathfrak{t}]$  (by a TM  $\Pi_{\mathrm{Kt}}$ ) and  $\mathsf{MKtP} \in \mathsf{Heur}_{0,\gamma_{\mathrm{fn}}}\mathsf{DTIME}[\mathcal{O}(n)]$  with false negative error probability  $\gamma_{\mathrm{fn}}(n) \coloneqq 1/2n\mathfrak{t}(2n) - 2/2^n$  (by a TM  $\widetilde{\Pi}_{\mathrm{Kt}}$ ). See Fig. 5 for the modified traversal algorithm  $\mathrm{TRAVERSE}'_{\hat{c}_{\Omega},c_{\mathrm{Kt}},\Pi_{\mathfrak{t}},\Pi_{\mathrm{Kt}},\widetilde{\Pi}_{\mathrm{Kt}}}$ . Let  $\mathcal{M}'_{\hat{c}_{\Omega},c_{\mathrm{Kt}},\Pi_{\mathfrak{t}},\Pi_{\mathrm{Kt}}}$  be a TM implementing the modified  $\mathrm{TRAVERSE}'_{\hat{c}_{\Omega},c_{\mathrm{Kt}},\Pi_{\mathfrak{t}},\Pi_{\mathrm{Kt}},\widetilde{\Pi}_{\mathrm{Kt}}}$ . Clearly, if the analog of Claim 1 holds, then  $\mathcal{M}'_{\hat{c}_{\Omega},c_{\mathrm{Kt}},\Pi_{\mathfrak{t}},\Pi_{\mathrm{Kt}},\widetilde{\Pi}_{\mathrm{Kt}}}$  does not terminate for the same reason as in Claim 2 (note there that the check in line 14 is an errorless

check). Because the definition of the counter variable  $t_{\ell}$  is identical to  $\text{TRAVERSE}_{\hat{c}_{\Omega}, c_{\text{Kt}}, \Pi_{\mathfrak{t}}, \Pi_{\text{Kt}}, \widetilde{\Pi}_{\text{Kt}}}$  the analog of Claim 3 also holds.

It remains to argue the analog of Claim 1. As before we observe that the TM  $\mathcal{M}'_{\hat{c}_{\Omega},c_{\mathrm{Kt}},\Pi_{\mathfrak{t}},\Pi_{\mathrm{Kt}},\Pi_{$ 

$$\widetilde{\Delta}_{\ell} \coloneqq \widetilde{t}_{\ell} - \widetilde{t}_{\ell-1} \tag{56}$$

$$\leq |\mathcal{Z}_{\ell}| (2^{c_{\mathrm{Kt}}+1}\ell + 4\ell) + 2^{\ell} c_{\ell} (\ell) t (2\ell) + 2^{c_{\mathrm{Kt}}+1}\ell + t (2\ell) + 4\ell + c^{3} \ell^{2} + c_{\mathrm{Tr}} \ell + t (2\ell) + 4\ell \tag{57}$$

$$\leq \underbrace{|Z_{\ell}|(2^{c_{\mathrm{Kt}}+1}\ell+4\ell)+2^{\ell}\gamma_{\mathrm{fn}}(\ell)\mathfrak{t}(2\ell)}_{\text{steps for }Z_{\ell}\setminus\{\check{z}_{\ell}\}\text{ in lines }6,8,19} + \underbrace{2^{c_{\mathrm{Kt}}+1}\ell+\mathfrak{t}(2\ell)+4\ell+c_{\mathrm{Kt}}^{3}\ell^{2}+c_{\mathrm{Kt}}\ell+\mathfrak{t}(2\ell)+4\ell}_{\text{steps for }\check{z}_{\ell}\text{ in lines }6-17} \tag{57}$$

$$\leq \left(2^{c_{\rm Kt}+1}+4\right)|Z_{\ell}|\ell + \left(2^{\ell}\gamma_{\rm fn}(\ell)+2\right)\mathfrak{t}(2\ell) + \left(2^{c_{\rm Kt}+1}+8+c_{\rm Kt}^{3}+c_{\rm Kt}\right)\ell^{2}$$
(58)

$$\leq 2^{c_{\mathrm{Kt}}+2} |Z_{\ell}| \ell + (2^{\ell} \gamma_{\mathrm{fn}}(\ell) + 2) \mathfrak{t}(2\ell) + 2^{2c_{\mathrm{Kt}}+\ell - |\log_2(\ell)|}$$
(59)

$$\leq 2^{c_{\rm Kt}+2} |Z_{\ell}| \ell + 2^{2c_{\rm Kt}+\ell-\lceil \log_2(\ell)\rceil+1}$$
(60)

$$=t_{\ell}-t_{\ell-1} \tag{61}$$

$$=: \Delta_{\ell} \tag{62}$$

that the variable  $t_{\ell}$  grows more quickly than  $\tilde{t}_{\ell}$  and since  $t_0 = 0 = \tilde{t}_0$ , it follows that  $t_{\ell} \ge \tilde{t}_{\ell}$  for any  $\ell \in \mathbb{N}$  which establishes Claim 3. The concluding part of the proof works exactly as in the proof of Theorem 1.

The reason why our proof can tolerate the additional runtime cost caused by the exact Kt solver  $\Pi_{\text{Kt}}$  is because the safety margin that we add to the counter in line 13 is more than we actually need for Theorem 1.

### 6 Proof of Lemma 1

**Lemma 1 (Infinitely-often bound).** For any sequence  $(\gamma_j)_{j \in \mathbb{N}}$  with  $\gamma_j \in [0,1]$  and  $\sigma_\ell \coloneqq \sum_{i=1}^{\ell} \gamma_i$  it holds that infinitely often  $\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa} e^{\sigma_{j-\kappa} - \sigma_j} \leq_{io} 2^{\ell} / \ell \ln(\ell)$ .

*Proof.* Our proof of this claim is quite technical and somewhat tedious although it fundamentally only requires analytic Riemann integration bounds (see Appendix A). A high-level intuition for our bound may best be explained by looking at the double sum

$$\mathfrak{s}(\ell) \coloneqq \sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
(63)

where  $\sigma_{\ell} \coloneqq \sum_{i=1}^{\ell} \gamma_i$ . We don't know the exact values of  $\gamma_j \in [0, 1]$  but we see that the summands of the outer sum depend on  $\gamma_j$  in two ways. The faster  $\gamma_j$  grows the faster the outer summands grow because the *j*-th summand depends linearly on  $\gamma_j$ . On the other hand, the faster  $\gamma_j$  grows the faster  $\sigma_j$  grows and thus the slower the inner summands grow because of the  $e^{\sigma_j}$  term in the denominator of the  $\kappa$ -th inner summand. So, there is a "sweet spot" for the asymptotic growth rate of  $\gamma_j$  that maximizes the growth rate of  $\mathfrak{s}$ . The maximal growth rate is close to  $\Theta(\sum_{j=1}^{\ell} \frac{1}{j} \sum_{\kappa=1}^{j} 2^{\kappa}(1-\frac{\kappa}{j})^{\epsilon}) = \Theta(2^{\ell}/\ell^{1+\epsilon})$  for small  $\epsilon > 0$  and  $\gamma_j = \epsilon/j$ , thus  $\sigma_j \approx \epsilon \ln(j)$ . Thus we cannot hope to prove  $\mathfrak{s}(\ell) \in \mathcal{O}(2^{\ell}/\ell^{1+\epsilon})$  without further restrictions on  $\gamma_j$ . However, we can prove a weaker bound  $\mathfrak{s}(\ell) \leq_{\mathrm{io}} \mathcal{O}(2^{\ell}/\ell \ln(\ell))$ . The way we prove this bound is by establishing increasingly stronger lower bounds for the sum  $\sigma_{\ell}$ . The first bound will be of the rough form  $\sigma_{\ell} \in \Omega(\ln \ln(\ell))$ , the second one  $\sigma_{\ell} \in \Omega(\ln(\ell)^{1/17})$  and the third one  $\sigma_{\ell} \in \Omega(\ln(\ell)^3)$ . The last bound then yields a contradiction to the counter assumption  $2^{\ell}/\ell \ln(\ell) \leq_{\mathrm{abf}} \mathfrak{s}(\ell)$ .

Let us proceed with the formal proof. In long equations we highlight changes relative to the previous line with a gray background and give explanations in framed boxes. We use the convention that for any b < a the sum  $\sum_{i=a}^{b} \mathfrak{f}(i) \coloneqq 0$ . Suppose for contradiction

$$\frac{2^{\ell}}{\ell \ln(\ell)} \leq_{\mathsf{abf}} \mathfrak{s}(\ell) \coloneqq \sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}} , \qquad (64)$$

then there exists some  $\ell_1 \in \mathbb{N}$  such that for all  $\ell \geq \ell_1$ 

$$\frac{1}{\ell \ln(\ell)} \le \sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa-\ell}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
(65)

$$\leq \sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa-\ell} \tag{66}$$

$$\leq \sum_{j=1}^{\ell} 2^{j+1-\ell} \gamma_j \tag{67}$$

where Eq. (66) trivially uses  $\sigma_j \geq \sigma_{j-k}$  and Eq. (67) uses  $\sum_{\kappa=1}^{j} 2^{\kappa} = 2^{j+1} - 2$ . For convenience, we define a helper variable  $\delta_{\ell} \coloneqq \max(0, \lceil \ln \ln(\ell+1)/8 - \ln \ln(\ell_1)/4 \rceil) \leq \ell$ . Note that  $\delta_{\ell} \geq \log_2 \ln(\ell)/16$  for  $\ell \geq \ell_1$  if  $\ell_1$  is sufficiently large (which is without loss of generality). Using Riemann integration on the sum of Eq. (65) from  $\ell_1$  to  $\ell$  yields

 $=\frac{1}{2}$ 

=

$$\frac{1}{4}\ln\ln(\ell+1) - \frac{1}{4}\ln\ln(\ell_1) = \frac{1}{2}\int_{\ell_1}^{\ell+1} \frac{1}{2x\ln(x)} dx$$
(68)

$$\leq \frac{1}{2} \sum_{i=\ell_1}^{\ell} \frac{1}{2i \ln(i)}$$
 Fact 3 (69)

$$\sum_{i=1}^{\ell} 2^{-i} \sum_{j=1}^{i} 2^{j} \gamma_{j} \tag{72}$$

$$= \frac{1}{2} \sum_{j=1}^{\ell} 2^{j} \gamma_{j} \sum_{i=j}^{\ell} 2^{-i}$$
 Lemma 3  
sum switching (73)

$$\leq \frac{1}{2} \sum_{j=1}^{\ell} 2^j \gamma_j \sum_{i=j}^{\infty} 2^{-i} \qquad \qquad \boxed{\ell < \infty \text{ and } 2^{-i} \ge 0} \qquad (74)$$

$$=\sum_{j=1}^{\ell} \gamma_j \qquad \qquad \sum_{i=j}^{\infty} 2^{-i} = 2^{1-j} \qquad (75)$$
$$= \sigma_\ell \qquad \qquad (76)$$

$$\leq \sigma_{\ell} - \sigma_{\delta_{\ell}} + \delta_{\ell} \quad . \tag{77}$$

Reordering the terms yields

$$\sigma_{\ell} - \sigma_{\delta_{\ell}} \ge \ln \ln(\ell + 1)/8 \ge \ln \ln(\ell)/16 \tag{78}$$

for all  $\ell \geq \ell_1$ . To get this bound we started Eq. (65) off with the trivial bound  $\sigma_j \geq \sigma_{j-\kappa}$ . Now, we can use our new nontrivial bound for  $\sigma_j$  repeat the previous procedure and obtain an even better bound.

Plugging Eq. (78) back into a weighted sum of Eq. (65) gives the better bound on  $\sigma_{\ell}$  for  $\ell \geq \ell_1$ , i.e.,

$$\ln(\ell+1) - \ln(\ell_1)$$

$$= \int_{\ell_1}^{\ell+1} \frac{1}{x} dx$$
(80)

$$\leq \sum_{i=\ell_1}^{\ell} \frac{1}{i}$$
 (81)

$$=\sum_{i=\ell_1}^{\ell} \frac{\ln(i)}{i\ln(i)}$$
(82)

$$\leq \sum_{i=\ell_1}^{\ell} \ln(i) \sum_{j=1}^{i} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa-i}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
(83)

$$=\sum_{i=\ell_1}^{\ell} \frac{\ln(i)}{2^i} \sum_{j=1}^{i} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
(84)

$$\leq \sum_{i=\ell_1}^{\ell} \frac{\ln(i)}{2^i} \sum_{j=1}^{i} \gamma_j \left( 2^{\ell_1} + \sum_{\kappa=\ell_1}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}} \right)$$
$$= \sum_{i=\ell_1}^{\ell} \frac{\ln(i)}{2^i} \left( 2^{\ell_1} \sigma_i + \sum_{j=1}^{i} \gamma_j \sum_{\kappa=\ell_1}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}} \right)$$

$$\leq 2^{\ell_1} \sum_{i=\ell_1}^{\ell} \frac{i \ln(i)}{2^i} + \sum_{i=\ell_1}^{\ell} \frac{\ln(i)}{2^i} \sum_{j=\ell_1}^{i} \gamma_j \sum_{\kappa=\ell_1}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$

$$\gamma_j \le 1 \implies \sigma_i \le i \tag{89}$$

(87)

(88)

$$\boxed{\sum_{i=1}^{\infty} \frac{i \ln(i)}{2^i} \le 2} \tag{90}$$

$$\sigma_j \ge \sigma_{j-\kappa} \tag{92}$$

 $\sum_{\kappa=\ell_1}^{j-1-\delta_j} 2^{\kappa} \le 2^{j-\delta_j}$ (93)

$$\begin{array}{c} \kappa \ge j - \delta_j \\ \Longrightarrow \delta_j \ge j - \kappa \end{array}$$

$$(94)$$

$$\begin{aligned} \sigma_j - \sigma_{\delta_j} &\geq \ln \ln(j) / 16 \\ \forall j &\geq \ell_1 \text{ by Eq. (78)} \end{aligned}$$
 (95)

$$\delta_j \ge \log_2 \ln(j)/16 \tag{96}$$

 $\leq 2^{\ell_1+1} + \sum_{j=\ell_*}^{\ell} \frac{\ln(i)}{2^i} \sum_{j=\ell_*}^{i} \gamma_j \left( \sum_{\kappa=\ell_*}^{j-1-\delta_j} 2^{\kappa} + \sum_{\kappa=j-\delta_*}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}} \right)$ 

 $\leq 2^{\ell_1+1} + \sum_{i=\ell}^{\ell} \frac{\ln(i)}{2^i} \sum_{i=\ell}^{i} \gamma_j \left( 2^{j-\delta_j} + \sum_{i=-i,\kappa}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}} \right)$ 

 $\leq 2^{\ell_1+1} + \sum_{i=\ell}^{\ell} \frac{\ln(i)}{2^i} \sum_{i=\ell}^{i} \gamma_j \left( 2^{j-\delta_j} + \sum_{i=\ell}^{j} \frac{2^{\kappa}}{\sigma_j - \sigma_{\delta_j}} \right)$ 

 $\leq 2^{\ell_1+1} + \sum_{i=\ell_1}^{\ell} \frac{\ln(i)}{2^i} \sum_{j=\ell_1}^{i} \gamma_j \left( 2^{j-\delta_j} + \sum_{\kappa=j-\delta_j}^{j} \frac{2^{\kappa}}{\ln(j)^{1/16}} \right)$ 

 $\leq 2^{\ell_1+1} + \sum_{i=\ell}^{\ell} \frac{\ln(i)}{2^i} \sum_{i=\ell}^{i} \gamma_j \left( \frac{2^j}{\ln(j)^{1/16}} + \sum_{i=\ell,\kappa}^{j} \frac{2^{\kappa}}{\ln(j)^{1/16}} \right)$ 

 $\leq 2^{\ell_1+1} + \sum_{i=\ell_1}^{\ell} \frac{\ln(i)}{2^i} \sum_{j=\ell_1}^{i} \gamma_j \left( \frac{2^{j+1}}{\ln(j)^{1/16}} + \sum_{\kappa=j-\delta_j}^{j} \frac{2^{\kappa}}{\ln(j)^{1/16}} \right)$ 

 $= 2^{\ell_1+1} + \sum_{i=\ell_1}^{\ell} \frac{\ln(i)}{2^i} \sum_{j=\ell_1}^{i} \gamma_j \sum_{\kappa=j-\delta_j}^{j+1} \frac{2^{\kappa}}{\ln(j)^{1/16}}$ 

$$= 2^{\ell_1+1} + 16 \sum_{j=\ell_1}^{\ell} \gamma_j \ln(j)^{15/16}$$

$$\leq 2^{\ell_1+1} + 16 \sum_{j=\ell_1}^{\ell} \gamma_j \ln(\ell)^{15/16}$$

$$\ln(x)^{15/16}$$
is non-decreasing (104)

 $\leq 2^{\ell_1+1} + 16 \sigma_\ell \ln(\ell)^{15/16}$ . (105)

Let  $\delta'_{\ell} \coloneqq \lceil \log_2(e) \ln(\ell)^{1/17} \rceil$ . Thus there exists some sufficiently large  $\ell_2 \in \mathbb{N}$  s.t. for all  $\ell \ge \ell_2$  it holds that σ

$$\sigma_{\ell} - \sigma_{\delta'_{\ell}} \ge \sigma_{\ell} - \delta'_{\ell} \tag{106}$$

$$\geq \left(\ln(\ell+1) - \ln(\ell_1) - 2^{\ell_1+1}\right) / \left(16\ln(\ell)^{15/16}\right) - \log_2(e)\ln(\ell)^{1/17} - 1$$
(107)

$$\geq \ln(\ell)^{1/17}$$
 (108)

Now, we repeat the previous strategy for a third time to reach the final sufficient bound  $\sigma_{\ell} \in \Omega(\ln(\ell)^3)$ . Plugging Eq. (106) back into a weighted sum of Eq. (65) gives the better bound on  $\sigma_{\ell}$  for  $\ell \geq \ell_2$ 

$$2(\ell+1)^{1/2} - 2\ell_2^{1/2} \tag{109}$$

$$\int^{\ell+1} x^{1/2} \, .$$

$$= \int_{\ell_2} \frac{1}{x} dx \tag{110}$$

$$\leq \sum_{i=\ell_2}^{l} \frac{1}{i} \tag{111}$$

$$=\sum_{i=\ell_2}^{\ell} \frac{i^{1/2} \ln(i)}{i \ln(i)}$$
(112)

$$\leq \sum_{i=\ell_2}^{\ell} \frac{i^{1/2} \ln(i)}{2^i} \sum_{j=1}^{i} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
Eq. (65)
  
(113)

$$\leq \sum_{i=\ell_{2}}^{\ell} \frac{i^{1/2} \ln(i)}{2^{i}} \sum_{j=1}^{i} \gamma_{j} \left( \sum_{\kappa=1}^{\ell_{2}-1} \frac{2^{\kappa}}{e^{\sigma_{j}-\sigma_{j-\kappa}}} + \sum_{\kappa=\ell_{2}}^{j} \frac{2^{\kappa}}{e^{\sigma_{j}-\sigma_{j-\kappa}}} \right)$$

$$\leq \sum_{i=\ell_{2}}^{\ell} \frac{i^{1/2} \ln(i)}{2^{i}} \sum_{j=1}^{i} \gamma_{j} \left( 2^{\ell_{2}} + \sum_{\kappa=\ell_{2}}^{j} \frac{2^{\kappa}}{e^{\sigma_{j}-\sigma_{j-\kappa}}} \right)$$

$$\equiv \sum_{i=\ell_{2}}^{\ell} \frac{i^{1/2} \ln(i)}{2^{i}} \left( 2^{\ell_{2}} \sigma_{i} + \sum_{j=1}^{i} \gamma_{j} \sum_{\kappa=\ell_{2}}^{j} \frac{2^{\kappa}}{e^{\sigma_{j}-\sigma_{j-\kappa}}} \right)$$

$$\leq 2^{\ell_{2}} \sum_{i=\ell_{2}}^{\ell} \frac{i^{3/2} \ln(i)}{2^{i}} + \sum_{i=\ell_{2}}^{\ell} \frac{i^{1/2} \ln(i)}{2^{i}} \sum_{j=\ell_{2}}^{i} \gamma_{j} \sum_{\kappa=\ell_{2}}^{j} \frac{2^{\kappa}}{e^{\sigma_{j}-\sigma_{j-\kappa}}}$$

$$(118)$$

$$\leq 2^{\ell_{2}+1} + \sum_{i=\ell_{2}}^{\ell} \frac{i^{1/2} \ln(i)}{2^{i}} \sum_{j=\ell_{2}}^{i} \gamma_{j} \sum_{\kappa=\ell_{2}}^{j} \frac{2^{\kappa}}{e^{\sigma_{j}-\sigma_{j-\kappa}}}$$

$$\leq 2^{\ell_2+1} + \sum_{i=\ell_2}^{\ell} \frac{i^{1/2} \ln(i)}{2^i} \sum_{j=\ell_2}^{i} \gamma_j \left( \sum_{\kappa=\ell_2}^{j-1-\delta'_j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}} + \sum_{\kappa=j-\delta'_j}^{j} \frac{2^{\kappa}}{e^{\sigma_j - \sigma_{j-\kappa}}} \right)$$
 [split sum] (120)

$$\leq 2^{\ell_2+1} + \sum_{i=\ell_2}^{\ell} \frac{i^{1/2} \ln(i)}{2^i} \sum_{j=\ell_2}^{i} \gamma_j \left( \sum_{\kappa=\ell_2}^{j-1-\delta'_j} 2^{\kappa} + \sum_{\kappa=j-\delta'_j}^{j} \frac{2^{\kappa}}{e^{\sigma_j-\sigma_{j-\kappa}}} \right)$$

$$(121)$$

(122)

(123)

(124)

(125)

(126)

(127)

 $\begin{aligned} \kappa &\geq j - \delta'_j \\ \implies \delta'_j &\geq j - \kappa \end{aligned}$ 

 $\delta_j' \ge \log_2(e) \ln(j)^{1/17}$ 

$$\leq 2^{\ell_2+1} + \sum_{i=\ell_2}^{\ell} \frac{i^{1/2} \ln(i)}{2^i} \sum_{j=\ell_2}^{i} \gamma_j \left( 2^{j-\delta'_j} + \sum_{\kappa=j-\delta'_j}^{j} \frac{2^{\kappa}}{e^{j}} \right)$$

$$\leq 2^{\ell_2+1} + \sum_{i=\ell_2}^{\ell} \frac{i^{1/2} \ln(i)}{2^i} \sum_{j=\ell_2}^{i} \gamma_j \left( \frac{2^j}{e^{\ln(j)^{1/17}}} + \sum_{\kappa=j-\delta'_j}^{j} \frac{2^{\kappa}}{e^{\ln(j)^{1/17}}} \right)$$

$$\leq 2^{\ell_2+1} + \sum_{i=\ell_2}^{\ell} \frac{i^{1/2} \ln(i)}{2^i} \sum_{j=\ell_2}^{i} \gamma_j \left( \frac{2^{j+1}}{e^{\ln(j)^{1/17}}} + \sum_{\kappa=j-\delta'_j}^{j} \frac{2^{\kappa}}{e^{\ln(j)^{1/17}}} \right)$$

$$= 2^{\ell_2+1} + \sum_{i=\ell_2} \frac{i^{1/2} \ln(i)}{2^i} \sum_{j=\ell_2} \gamma_j \sum_{\kappa=j-\delta'_j}^{i} \frac{2^{\kappa}}{e^{\ln(j)^{1/17}}}$$
(1)  
$$\leq 2^{\ell_2+1} + \sum_{i=\ell_2}^{\ell} \frac{i^{1/2} \ln(i)}{2^i} \sum_{j=\ell_2}^{i} \gamma_j \frac{2^{j+2}}{e^{\ln(j)^{1/17}}}$$
(1)

$$= 2^{\ell_2 + 1} + 4 \sum_{j=\ell_2}^{\ell} \gamma_j \cdot \frac{2^j}{e^{\ln(j)^{1/17}}} \sum_{i=j}^{\ell} \frac{i^{1/2} \ln(i)}{2^i}$$
(128)  
Lemma 3  
sum switching  
(129)

$$= 2^{\ell_2 + 1} + 16 \sum_{j=\ell_2}^{\ell} \gamma_j \frac{j^{1/2} \ln(j)}{e^{\ln(j)^{1/17}}}$$
(132)

$$\leq 2^{\ell_2+1} + 16 \,\sigma_\ell \,\frac{\ell^{1/2}\ln(\ell)}{e^{\ln(\ell)^{1/17}}} \,. \tag{134}$$

Let  $\delta_{\ell}'' := \lceil \log_2(e) \ln(\ell)^3 \rceil$ . Thus there exists some sufficiently large  $\ell_3 \in \mathbb{N}$  s.t. for all  $\ell \ge \ell_3$  it holds that  $\lim_{\ell \to 0} (\ell)^{1/17}$ 

$$\sigma_{\ell} - \sigma_{\delta_{\ell}''} \ge \sigma_{\ell} - \delta_{\ell}'' \ge \left( (\ell+1)^{1/2} - (\ell_2)^{1/2} - 2^{\ell_2 + 1} \right) \frac{e^{\ln(\ell)^{1/17}}}{16\ell^{1/2}\ln(\ell)} - \log_2(e)\ln(\ell)^3 - 1 \ge \ln(\ell)^3 .$$
(135)

Finally, we can use our last bound to obtain a contradiction. Plugging Eq. (135) into Eq. (65) yields

$$\frac{1}{\ell \ln(\ell)} \le \sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa-\ell}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
(136)

$$\leq \sum_{j=1}^{\ell_3-1} \gamma_j \sum_{\kappa=1}^j \frac{2^{\kappa-\ell}}{e^{\sigma_j - \sigma_{j-\kappa}}} + \sum_{j=\ell_3}^\ell \gamma_j \sum_{\kappa=1}^j \frac{2^{\kappa-\ell}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
 [split sum] (137)

$$\leq \sum_{j=1}^{\ell_3-1} \gamma_j 2^{j+1-\ell} + \sum_{j=\ell_3}^{\ell} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa-\ell}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
(139)

$$\leq 2^{\ell_3+1-\ell} + \sum_{j=\ell_3}^{\ell} \gamma_j \sum_{\kappa=1}^{j} \frac{2^{\kappa-\ell}}{e^{\sigma_j - \sigma_{j-\kappa}}}$$
(140)

$$\leq 2^{\ell_3+1-\ell} + \sum_{j=\ell_3}^{\ell} \gamma_j \left( \sum_{\kappa=1}^{j-1-\delta_j''} \frac{2^{\kappa-\ell}}{e^{\sigma_j - \sigma_{j-\kappa}}} + \sum_{\kappa=j-\delta_j''}^{j} \frac{2^{\kappa-\ell}}{e^{\sigma_j - \sigma_{j-\kappa}}} \right)$$
 [split sum] (141)

$$\leq 2^{\ell_3 + 1 - \ell} + \frac{2^{\ell_2}}{e^{\ln(\ell)^3}} + \frac{1}{e^{\ln(\ell)^3}}$$

$$\leq 2^{\ell_3 + 1 - \ell} + \frac{2\ell^2}{e^{\ln(\ell)^3}}$$

$$(150)$$

$$\ell \geq 2$$

$$(151)$$

or equivalently the contradiction

$$1 \le \frac{\ell \ln(\ell)}{2^{\ell - \ell_3 - 1}} + \frac{2\ell^3 \ln(\ell)}{e^{\ln(\ell)^3}} \to 0$$
(152)

for  $\ell \to \infty$ .

To the valiant reader that has retraced the full proof of Lemma 1 we want to put the proposition To the valual reader that has retraced the full proof of Lemma 1 we want to put the proportion that the proof can be carried out so long as the right-hand side of the lemma has the form  $2^{\ell}/\prod_{i=0}^{k} \ln^{(i)}(\ell)$ for some fixed  $k \in \mathbb{N}$  where  $\ln^{(i)}$  is the *i*-th times iterated logarithm. Towards this, we assume a slight simplification of the form  $\sum_{j=1}^{\ell} \gamma_j \sum_{\kappa=1}^{j} 2^{\kappa}/e^{\sigma_j-\sigma_{j-\kappa}} \approx \sum_{j=1}^{\ell} 2^j \gamma_j/e^{\sigma_j} \leq_{io} 2^{\ell}/\prod_{i=0}^{k} \ln^{(i)}(\ell)$ . We sketch a proof by induction where we go from a bound  $\sigma_{\ell} \in \Omega(\ln^{(k+1)}(\ell))$  to  $\sigma_{\ell} \in \Omega(\ln^{(k)}(\ell))$ . Starting out with the counter assumption  $\sum_{j=1}^{\ell} 2^j \gamma_j/e^{\sigma_j} \geq_{abf} 2^{\ell}/\prod_{i=0}^{k} \ln^{(i)}(\ell)$  we find that the first repe-tition of Eq. (68) is of the form  $\Theta(\ln^{(k+1)}(\ell)) = \Theta(\int 1/\prod_{i=0}^{k} \ln^{(i)}(\ell) d\ell) \leq \Theta(\sigma_{\ell})$ . Inserting this bound into the counter assumption gives

the counter assumption gives

$$\sum_{j=1}^{\ell} 2^{j} \gamma_{j} / e^{\ln^{(k+1)}(j) \cdot \Theta(1)} = \sum_{j=1}^{\ell} 2^{j} \gamma_{j} / \ln^{(k)}(j)^{\Theta(1)} \ge 2^{\ell} / \prod_{i=0}^{k} \ln^{(i)}(\ell)$$
(153)

$$\Longrightarrow \sigma_{\ell} \ge \ln^{(k)}(j)^{\Theta(1)} / \prod_{i=0}^{k} \ln^{(i)}(\ell)$$
(154)

The second repetition of Eq. (68) takes the form

$$\Theta\left(\ln^{(k)}(\ell)\right) = \Theta\left(\int \frac{\ln^{(k)}(\ell)}{\prod_{i=0}^{k} \ln^{(i)}(\ell)} d\ell\right)$$
(155)

$$\leq \Theta\left(\sum_{\ell'=1}^{\ell} \ln^{(k)}(\ell') \sum_{j=1}^{\ell'} 2^{j-\ell} \gamma_j / \ln^{(k)}(j)^{\Theta(1)}\right)$$
(156)

$$\leq \Theta\left(\ln^{(k)}(\ell)^{1-\Theta(1)}\sigma_{\ell}\right) \tag{157}$$

$$\implies \sigma_{\ell} \in \Omega\left(\ln^{(k)}(\ell)^{\Theta(1)}\right) \tag{158}$$

which is already a better bound than from the first repetition, although not quite  $\Theta(\sigma_{\ell}) \ge \Theta(\ln^{(k)}(\ell))$ . The third repetition of Eq. (68) takes the form

$$\Theta\left(\ln^{(k)}(\ell)\right) = \Theta\left(\int \frac{\ln^{(k)}(\ell)}{\prod_{i=0}^{k} \ln^{(i)}(\ell)} d\ell\right)$$
(159)

$$\leq \Theta\left(\sum_{\ell'=1}^{\ell} \ln^{(k)}(\ell') \sum_{j=1}^{\ell'} 2^{j-\ell} \gamma_j / e^{\ln^{(k)}(j)^{\Theta(1)}}\right)$$
(160)

$$\leq \Theta\left(\ln^{(k)}(\ell)/e^{\ln^{(k)}(j)^{\Theta(1)}} \cdot \sigma_{\ell}\right)$$
(161)

$$\implies \sigma_{\ell} \in \Omega\left(e^{\ln^{(k)}(\ell)^{\Theta(1)}}\right) \ge \Theta\left(\ln^{(k)}(\ell)\right)$$
(162)

which concludes the induction step.

### 7 Acknowledgments

The author would like to thank the anonymous reviewers for their helpful comments. Moreover, the author expresses his gratitude to Rafael Pass for suggesting the problem of an unconditional lower bound for Kt, Yanyi Liu for many helpful discussions about meta-complexity, Akın Ünal for checking the proof of a previous version of Lemma 1, and Chris Brzuska for a helpful discussion about the density of 1-random prefixes.

### References

- [AB18] B. Aydinlioğlu and E. Bach. Affine relativization: unifying the algebrization and relativization barriers. ACM Trans. Comput. Theory, 10(1), January 2018.
- [ABK<sup>+</sup>02] E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. In 43rd FOCS, pages 669–678. IEEE Computer Society Press, November 2002.
- [ABK<sup>+</sup>06] E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. SIAM Journal on Computing, 35(6):1467–1493, 2006. eprint: https://doi.org/10.1137/ 050628994.
- [ACM<sup>+</sup>21] E. Allender, M. Cheraghchi, D. Myrisiotis, H. Tirumala, and I. Volkovich. One-Way Functions and a Conditional Variant of MKTP. In M. Bojańczy and C. Chekuri, editors, 41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021), volume 213 of Leibniz International Proceedings in Informatics (LIPIcs), 7:1–7:19, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.
- [AHT23] E. Allender, S. Hirahara, and H. Tirumala. Kolmogorov complexity characterizes statistical zero knowledge. In Y. T. Kalai, editor, *ITCS 2023*, volume 251, 3:1–3:19. LIPIcs, January 2023.
- [AIV21] E. Allender, R. Ilango, and N. Vafa. The non-hardness of approximating circuit size. Theory of Computing Systems, 65(3):559–578, April 2021.
- [AKR<sup>+</sup>11] E. Allender, M. Koucký, D. Ronneburger, and S. Roy. The pervasive reach of resource-bounded kolmogorov complexity in computational complexity theory. *Journal of Computer and System Sciences*, 77(1):14–40, 2011. Celebrating Karp's Kyoto Prize.
- [All01] E. Allender. When worlds collide: derandomization, lower bounds, and kolmogorov complexity. In R. Hariharan, V. Vinay, and M. Mukund, editors, FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg, 2001.
- [All21] E. Allender. Vaughan jones, kolmogorov complexity, and the new complexity landscape around circuit minimization. *New Zealand Journal of Mathematics*, 52:585–604, September 2021.

[AW08] S. Aaronson and A. Wigderson. Algebrization: a new barrier in complexity theory. In R. E. Ladner and C. Dwork, editors, 40th ACM STOC, pages 731–740. ACM Press, May 2008.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the  $\mathcal{P} = :\mathcal{NP}$  question. SIAM Journal on Computing, 4(4):431–442, 1975. eprint: https://doi.org/10.1137/0204037.

- [BLM<sup>+</sup>23] M. Ball, Y. Liu, N. Mazor, and R. Pass. Kolmogorov comes to cryptomania: on interactive kolmogorov complexity and key-agreement. In 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS), pages 458–483, 2023.
- [BM82] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo random bits. In 23rd FOCS, pages 112–117. IEEE Computer Society Press, November 1982.
- [Cha66] G. J. Chaitin. On the length of programs for computing finite binary sequences. J. ACM, 13(4):547– 569, October 1966.
- [Cha69] G. J. Chaitin. On the simplicity and speed of programs for computing infinite sets of natural numbers. J. ACM, 16(3):407–422, July 1969.
- [Cha75] G. J. Chaitin. A theory of program size formally identical to information theory. J. ACM, 22(3):329– 340, July 1975.
- [DH10] R. Downey and D. Hirschfeldt. Algorithmic Randomness and Complexity. 2010.
- [GGM84] O. Goldreich, S. Goldwasser, and S. Micali. On the cryptographic applications of random functions. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 276–288. Springer, Berlin, Heidelberg, August 1984.
- [GGS<sup>+</sup>13] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, 45th ACM STOC, pages 467–476. ACM Press, June 2013.
- [GII<sup>+</sup>19] A. Golovnev, R. Ilango, R. Impagliazzo, V. Kabanets, A. Kolokolova, and A. Tal. AC<sup>0</sup>[p] lower bounds against MCSP via the coin problem. In C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, editors, *ICALP 2019*, volume 132 of *LIPIcs*, 66:1–66:15. Schloss Dagstuhl, July 2019.
- [GKL<sup>+</sup>22] H. Goldberg, V. Kabanets, Z. Lu, and I. C. Oliveira. Probabilistic Kolmogorov Complexity with Applications to Average-Case Complexity. In S. Lovett, editor, 37th Computational Complexity Conference (CCC 2022), volume 234 of Leibniz International Proceedings in Informatics (LIPIcs), 16:1–16:60, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.
- [GM84] S. Goldwasser and S. Micali. Probabilistic encryption. Journal of Computer and System Sciences, 28(2):270–299, 1984.
- [Har83] J. Hartmanis. Generalized Kolmogorov complexity and the structure of feasible computations (preliminary report). In 24th FOCS, pages 439–445. IEEE Computer Society Press, November 1983.
- [HIL<sup>+</sup>99] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. SIAM Journal on Computing, 28(4):1364–1396, 1999.
- [Hir18] S. Hirahara. Non-black-box worst-case to average-case reductions within NP. In M. Thorup, editor, 59th FOCS, pages 247–258. IEEE Computer Society Press, October 2018.
- [Hir20a] S. Hirahara. Characterizing average-case complexity of PH by worst-case meta-complexity. In 61st FOCS, pages 50–60. IEEE Computer Society Press, November 2020.
- [Hir20b] S. Hirahara. Unexpected hardness results for Kolmogorov complexity under uniform reductions. In K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath, and J. Chuzhoy, editors, 52nd ACM STOC, pages 1038–1051. ACM Press, June 2020.
- [Hir20c] S. Hirahara. Unexpected power of random strings. In T. Vidick, editor, ITCS 2020, volume 151, 41:1–41:13. LIPIcs, January 2020.
- [Hir22a] S. Hirahara. NP-hardness of learning programs and partial MCSP. In *63rd FOCS*, pages 968–979. IEEE Computer Society Press, October 2022.
- [Hir22b] S. Hirahara. Symmetry of Information from Meta-Complexity. In S. Lovett, editor, 37th Computational Complexity Conference (CCC 2022), volume 234 of Leibniz International Proceedings in Informatics (LIPIcs), 26:1–26:41, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.
- [HIR23] Y. Huang, R. Ilango, and H. Ren. Np-hardness of approximating meta-complexity: a cryptographic approach. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, pages 1067–1075, New York, NY, USA. Association for Computing Machinery, 2023.

[Hir23] S. Hirahara. Capturing one-way functions via np-hardness of meta-complexity. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, pages 1027–1038, New York, NY, USA. Association for Computing Machinery, 2023.

[HN22] S. Hirahara and M. Nanashima. On worst-case learning in relativized heuristica. In 62nd FOCS, pages 751–758. IEEE Computer Society Press, February 2022.

- [HS66] F. C. Hennie and R. E. Stearns. Two-tape simulation of multitape turing machines. J. ACM, 13(4):533–546, October 1966.
- [IKK09] R. Impagliazzo, V. Kabanets, and A. Kolokolova. An axiomatic approach to algebrization. In M. Mitzenmacher, editor, 41st ACM STOC, pages 695–704. ACM Press, May 2009.
- [Ila20a] R. Ilango. Approaching MCSP from above and below: hardness for a conditional variant and AC<sup>0</sup>[p].
   In T. Vidick, editor, *ITCS 2020*, volume 151, 34:1–34:26. LIPIcs, January 2020.
- [Ila20b] R. Ilango. Constant depth formula and partial function versions of MCSP are hard. In 61st FOCS, pages 424–433. IEEE Computer Society Press, November 2020.
- [Imp95] R. Impagliazzo. A personal view of average-case complexity. In Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference, pages 134–147, 1995.
- [KC00] V. Kabanets and J. Cai. Circuit minimization problem. In 32nd ACM STOC, pages 73–79. ACM Press, May 2000.
- [Ko86] K.-I. Ko. On the notion of infinite pseudorandom sequences. *Theoretical Computer Science*, 48:9–33, 1986.
- [Kol63] A. N. Kolmogorov. On tables of random numbers. Sankhyā: The Indian Journal of Statistics, Series A (1961-2002), 25(4):369–376, 1963.
- [Kol65] A. Kolmogorov. Three approaches to the quantitative definition of information. *Problemy Peredachi* Informatsii, 1:3–11, 1965.
- [Lev74] L. A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974.
- [Lev84] L. A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, 61(1):15–37, 1984.
- [LO22] Z. Lu and I. C. Oliveira. Theory and applications of probabilistic kolmogorov complexity, 2022. arXiv: 2205.14718 [cs.CC].
- [LOS21] Z. Lu, I. C. Oliveira, and R. Santhanam. Pseudodeterministic algorithms and the structure of probabilistic time. In S. Khuller and V. V. Williams, editors, 53rd ACM STOC, pages 303–316. ACM Press, June 2021.
- [LOZ22] Z. Lu, I. C. Oliveira, and M. Zimand. Optimal coding theorems in time-bounded kolmogorov complexity. In M. Bojanczyk, E. Merelli, and D. P. Woodruff, editors, *ICALP 2022*, volume 229 of *LIPIcs*, 92:1–92:14. Schloss Dagstuhl, July 2022.
- [LP20] Y. Liu and R. Pass. On one-way functions and kolmogorov complexity. In 61st FOCS, pages 1243– 1254. IEEE Computer Society Press, November 2020.
- [LP21a] Y. Liu and R. Pass. Cryptography from sublinear-time average-case hardness of time-bounded kolmogorov complexity. In S. Khuller and V. V. Williams, editors, 53rd ACM STOC, pages 722–735. ACM Press, June 2021.
- [LP21b] Y. Liu and R. Pass. On the possibility of basing cryptography on  $\mathsf{EXP} \neq \mathsf{BPP}$ . In T. Malkin and C. Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 11–40, Virtual Event. Springer, Cham, August 2021.
- [LP23a] Y. Liu and R. Pass. On one-way functions and sparse languages. In G. N. Rothblum and H. Wee, editors, TCC 2023, Part I, volume 14369 of LNCS, pages 219–237. Springer, Cham, November 2023.
- [LP23b] Y. Liu and R. Pass. On one-way functions and the worst-case hardness of time-bounded kolmogorov complexity. Cryptology ePrint Archive, Paper 2023/1086, 2023. https://eprint.iacr.org/2023/1086.
- [LP23c] Y. Liu and R. Pass. One-way functions and the hardness of (probabilistic) time-bounded Kolmogorov complexity w.r.t. samplable distributions. In H. Handschuh and A. Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 645–673. Springer, Cham, August 2023.
- [LP24] Y. Liu and R. Pass. A direct PRF construction from Kolmogorov complexity. In M. Joye and G. Leander, editors, EUROCRYPT 2024, Part IV, volume 14654 of LNCS, pages 375–406. Springer, Cham, May 2024.

[LV08]	M. Li and P. Vitányi. An Introduction to Kolmogorov Complexity and Its Applications. Springer New York, New York, NY, 2008.
[Mar66]	P. Martin-Löf. The definition of random sequences. Information and Control. 9(6):602–619, 1966.
[MP24a]	N. Mazor and R. Pass. Gap MCSP Is Not (Levin) NP-Complete in Obfustopia. In R. Santhanam, editor, 39th Computational Complexity Conference (CCC 2024), volume 300 of Leibniz International Proceedings in Informatics (LIPIcs), 36:1–36:21, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.
[MP24b]	N. Mazor and R. Pass. The Non-Uniform Perebor Conjecture for Time-Bounded Kolmogorov Com- plexity Is False. In V. Guruswami, editor, 15th Innovations in Theoretical Computer Science Con- ference (ITCS 2024), volume 287 of Leibniz International Proceedings in Informatics (LIPIcs), 80:1– 80:20, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.
[Nao91]	M. Naor. Bit commitment using pseudorandomness. <i>Journal of Cryptology</i> , 4(2):151–158, January 1991.
[Oli19]	I. C. Oliveira. Randomness and intractability in Kolmogorov complexity. In C. Baier, I. Chatzigian- nakis, P. Flocchini, and S. Leonardi, editors, <i>ICALP 2019</i> , volume 132 of <i>LIPIcs</i> , 32:1–32:14. Schloss Dagstuhl, July 2019.
[OPS19]	I. C. Oliveira, J. Pich, and R. Santhanam. Hardness magnification near state-of-the-art lower bounds. In A. Shpilka, editor, 34th Computational Complexity Conference (CCC 2019), volume 137 of Leibniz International Proceedings in Informatics (LIPIcs), 27:1–27:29, Dagstuhl, Germany. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.
[Rom90]	J. Rompel. One-way functions are necessary and sufficient for secure signatures. In 22nd ACM STOC, pages 387–394. ACM Press, May 1990.
[RR97]	A. A. Razborov and S. Rudich. Natural proofs. <i>Journal of Computer and System Sciences</i> , 55(1):24–35, 1997.
[RS21]	H. Ren and R. Santhanam. Hardness of kt characterizes parallel cryptography. In <i>Proceedings of the 36th Computational Complexity Conference</i> , CCC '21, Dagstuhl, DEU. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2021.
[RS22]	H. Ren and R. Santhanam. A Relativization Perspective on Meta-Complexity. In P. Berenbrink and B. Monmege, editors, 39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022), volume 219 of Leibniz International Proceedings in Informatics (LIPIcs), 54:1–54:13, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.
[Sip83]	M. Sipser. A complexity theoretic approach to randomness. In 15th ACM STOC, pages 330–335. ACM Press, April 1983.
[Sol60]	R. Solomonoff. A Preliminary Report on a General Theory of Inductive Inference. AFOSR TN-60- 1459. United States Air Force, Office of Scientific Research, 1960.
[Sol64a]	R. Solomonoff. A formal theory of inductive inference. part i. <i>Information and Control</i> , 7(1):1–22, 1964.
[Sol64b]	R. Solomonoff. A formal theory of inductive inference. part ii. <i>Information and Control</i> , 7(2):224–254, 1964.
[Tra84]	B. Trakhtenbrot. A survey of russian approaches to perebor (brute-force searches) algorithms. Annals of the History of Computing, 6(4):384–400, 1984.
[Wil13]	R. Williams. Improving exhaustive search implies superpolynomial lower bounds. SIAM Journal on Computing, 42(3):1218–1244, 2013. eprint: https://doi.org/10.1137/10080703X.

[Wil14] R. Williams. Nonuniform acc circuit lower bounds. J. ACM, 61(1), January 2014.

# A Technical Lemmas

**Lemma 3 (Sum-switching).** Let  $a, b, n, c \in \mathbb{N}$  be integers. Let  $\mathfrak{f}, \mathfrak{g} : \mathbb{N} \to \mathbb{R}$  be functions. Then

$$\sum_{i=a}^{n} \mathfrak{f}(i) \sum_{j=b}^{i-c} \mathfrak{g}(j) = \sum_{i=a}^{n} \sum_{j=b}^{i-c} \mathfrak{f}(i) \mathfrak{g}(j) = \sum_{j=b}^{n-c} \mathfrak{g}(j) \sum_{i=\max(j+c,a)}^{n} \mathfrak{f}(i) .$$
(163)

**Fact 3** (Riemann integration). Let  $a, n \in \mathbb{N}$  be integers. Let  $\mathfrak{f} : \mathbb{R} \to \mathbb{R}$  be a monotonically decreasing integrable function. Then

$$\int_{a}^{b+1} \mathfrak{f}(x) \mathrm{d}x \le \sum_{i=a}^{b} \mathfrak{f}(i) \le \int_{a-1}^{b} \mathfrak{f}(x) \mathrm{d}x \ . \tag{164}$$

Lemma 4 (Riemann bound). Let  $\nu \in [0,1]$  and  $j \in \mathbb{N}_{\geq 4}$ .

$$\sum_{i=j}^{\infty} \frac{i^{\nu} \ln(i)}{2^{i}} \le 4 \cdot \frac{j^{\nu} \ln(j)}{2^{j}}$$
(165)

Proof.

$$-\frac{\partial}{\partial x}\frac{x^{\nu}\ln(x)}{2^{x}} = \frac{x^{\nu}\ln(x)\ln(2)}{2^{x}} - \frac{x^{\nu-1}(\nu\ln(x)-1)}{2^{x}} \ge \frac{x^{\nu}\ln(x)}{2^{x+1}}$$
(166)

for all  $x \ge 0$ . Because  $\frac{x^{\nu} \ln(x)}{2^x}$  is monotonically decreasing for  $x \ge 3$  it follows

$$\sum_{i=j}^{\infty} \frac{i^{\nu} \ln(i)}{2^{i}} \le \int_{j-1}^{\infty} \frac{x^{\nu} \ln(x)}{2^{x}} \mathrm{d}x$$
(167)

$$=2\int_{j-1}^{\infty} \frac{x^{\nu}\ln(x)}{2^{x+1}} \mathrm{d}x$$
(168)

$$\leq 2 \int_{j-1}^{\infty} \left( -\frac{\partial}{\partial x} \frac{x^{\nu} \ln(x)}{2^{x}} \right) \mathrm{d}x \tag{169}$$

$$=2\left[\frac{x^{\nu}\ln(x)}{2^{x}}\right]_{\infty}^{j-1} \tag{170}$$

$$=2\frac{\ln(j-1)}{2^{j-1}}$$
(171)

$$\leq 4 \frac{\ln(j)}{2^j} . \tag{172}$$

Density of 1-random strings For any (finite or infinite) string  $w \in \{0,1\}^* \cup 2^{\omega}$  and any length  $\ell$  let  $w \restriction \ell := w_1 ||...||w_\ell$  be the  $\ell$ -bit prefix of w. For each complexity deficiency  $d \in \mathbb{N}$  let  $W_d := \{w \in \{0,1\}^* \mid \forall \ell \in \{0,...,|w|\} : \mathrm{K}(w \restriction \ell) \geq \ell - d\}$  be the (finite) strings that are "d-prefix-random". Analogously, let  $\mathfrak{W}_d := \{w \in 2^{\omega} \mid \forall \ell \in \mathbb{N} : \mathrm{K}(w \restriction \ell) \geq \ell - d\}$  be the (infinite) strings that are "d-prefix-random".

**Lemma 5.** Prefixes of 1-random strings have arbitrary (constant) density on each length. More technially, for each  $\hat{\epsilon} \in \mathbb{R}_{\geq 0}$  there exists some  $\hat{d} \in \mathbb{N}$  such that  $|W_{\hat{d}} \cap \{0,1\}^{\ell}| \geq 2^{\ell}(1-\hat{\epsilon})$ .

*Proof.* For each string  $\sigma \in \{0,1\}^*$  let  $[[\sigma]] := \sigma || 2^{\omega}$  be its infinite extension. Let  $\mu$  be the uniform measure on the Cantor space  $2^{\omega}$  where for each  $\sigma \in \{0,1\}^*$  its probability is  $\mu([[\sigma]]) := 2^{-|\sigma|}$ . Let  $\mathfrak{W} := \bigcup_{d \in \mathbb{N}} \mathfrak{W}_d$  be the set of 1-random strings.

We know from [Mar66] (Corollary 6.2.6 in [DH10]) that  $\mu(\mathfrak{W}) = 1$ . Suppose for contradiction there existed some  $\hat{\epsilon} \in \mathbb{R}_{\geq 0}$  such that for each  $\hat{d} \in \mathbb{N}$  it holds that  $\mu(\mathfrak{W}_{\hat{d}}) \leq 1 - \hat{\epsilon}$ . Then, we would find the following contradiction  $1 = \mu(\mathfrak{W}) = \lim_{\hat{d} \to \infty} \mu(\mathfrak{W}_{\hat{d}}) \leq 1 - \hat{\epsilon} \leq 1$ . Consequently, for each  $\hat{\epsilon} \in \mathbb{R}_{\geq 0}$  there exists some  $\hat{d} \in \mathbb{N}$  such that  $\mu(\mathfrak{W}_{\hat{d}}) \geq 1 - \hat{\epsilon}$ .

Let  $R_{\ell} \coloneqq W_{\hat{d}} \cap \{0,1\}^{\ell}$  be the  $\hat{d}$ -prefix-random strings of length  $\ell$ . Note that  $\mathfrak{W}_{\hat{d}} \subseteq \bigcup_{\sigma \in R_{\ell}} [[\sigma]]$ . Hence  $1 - \hat{\epsilon} \leq \mu(\mathfrak{W}_{\hat{d}}) \leq \sum_{\sigma \in R_{\ell}} \mu([[\sigma]]) \leq \sum_{\sigma \in R_{\ell}} 2^{-\ell}$ . Thus, the  $\hat{d}$ -prefix-random strings have (at least) constant density in each length, i.e.,  $|R_{\ell}| \geq 2^{\ell}(1 - \hat{\epsilon})$ .