Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 201 (2023)

# Enumerating Complexity Revisited

Alexander Shekhovtsov* Nikolay Vereshchagin† Georgii Zakharov‡

**Abstract**

We study the relation between deterministic and randomized *enumerating* Kolmogorov complexity of sets $S \subseteq \mathbb{N}$. Let $\mathrm{D}(S)$ be the length of a shortest input on which a fixed optimal deterministic machine enumerates $S$, and let $\mathrm{R}(S)$ be the probability that a fixed optimal randomized machine enumerates $S$. It is easy to see that $-\log \mathrm{R}(S) \le \mathrm{D}(S) + O(\log \mathrm{D}(S))$ for every $S$; the converse direction is much harder. Solovay proved $\mathrm{D}(S) \le -3\log \mathrm{R}(S) + O(\log\log \frac{1}{\mathrm{R}(S)})$ for all $S$, while for finite sets the sharper bound with the constant 2 is known.

We show that the case of infinite sets carries no extra cost: *any* inequality of the form $\mathrm{D}(F) \le -\alpha \log \mathrm{R}(F) + O(\log\log \frac{1}{\mathrm{R}(F)})$ with a computable $\alpha$, valid for all finite $F$, automatically extends to all sets. More generally, we prove a single dichotomy (Theorem 1) from which such transfer results follow for an arbitrary computable non-decreasing bound, and which also shows that every c.e. set $S$ has a finite subset $F$ such that $F$ has at least as large deterministic complexity as $S$ but randomized one approximately the same or lower. Our result makes qualitative progress on a problem studied by Solovay in 1977 by showing that the search for the optimal constant can be reduced to a purely combinatorial question about a game played with finite sets. In particular, our result yields $\mathrm{D}(S) \le -2\log \mathrm{R}(S) + O(\log\log \frac{1}{\mathrm{R}(S)})$ for every $S$, improving Solovay's constant 3 to 2.

## 1 Introduction

### 1.1 Enumerating complexity

We study Kolmogorov complexity with respect to non-halting Turing machines. Since such machines do not halt, they can output an infinite object, so instead of measuring the complexity of strings we measure the complexity of subsets of $\mathbb{N}$. There are two natural ways to do this. The *deterministic enumerating complexity* $\mathrm{D}(S)$ of a set $S \subseteq \mathbb{N}$ is the length of a shortest input on which a fixed optimal machine enumerates $S$ (and nothing else). The *randomized enumerating complexity* is $-\log_2 \mathrm{R}(S)$, where $\mathrm{R}(S)$ is the probability that a fixed optimal machine with a random input tape enumerates exactly $S$. (All definitions are given in Section 2; log always means $\log_2$.)

Sampling a shortest description immediately gives

$$-\log \mathrm{R}(S) \;\le\; \mathrm{D}(S) + O(\log \mathrm{D}(S)) \tag{1}$$

for every $S \subseteq \mathbb{N}$: a random tape begins with a self-delimiting encoding of a shortest deterministic description of $S$ with probability $2^{-\mathrm{D}(S) - O(\log \mathrm{D}(S))}$ (Lemma 13). The converse direction — extracting a short deterministic description from a randomized one — is the subject of this paper. It can be viewed as a derandomization problem: given a randomized construction

*Columbia University, New York, USA.

†Moscow State University, HSE University, Yandex. This paper was prepared within the framework of the HSE University Basic Research Program (project no. HSE-BR-2025-24).

‡Department of Mathematics and Exeter College, University of Oxford, Oxford, UK. This research was conducted while the author was supported by a Hill Foundation Scholarship.



ISSN 1433-8092

that outputs the desired object with probability $p$, we want a deterministic program of length $O(\log(1/p))$ that reproduces it.

The question goes back to de Leeuw, Moore, Shannon and Shapiro [3], who proved that $S$ is deterministically enumerable if and only if $\mathrm{R}(S) > 0$; their argument, however, gives no quantitative bound on $\mathrm{D}(S)$. Solovay [7] proved the linear bound

$$\mathrm{D}(S) \;\leq\; -3\log \mathrm{R}(S) + O(\log\log \frac{1}{\mathrm{R}(S)}) \qquad \text{for all } S \subseteq \mathbb{N}. \tag{2}$$

It was later observed that the same method gives the better constant 2 for *finite* sets [8]. The natural question — what is the least $\alpha$ for which $\mathrm{D}(S) \leq -\alpha \log \mathrm{R}(S) + O(\log\log \frac{1}{\mathrm{R}(S)})$ holds for all $S$? — has been open since then. By (1) we have $\alpha \geq 1$.

In the halting case, the deterministic complexity equals the randomized complexity up to additive logarithmic terms due to the well-known coding theorems ([4]). These theorems significantly use the fact that a machine eventually halts. If a machine has halted, then the output it has produced is fixed. In contrast, in the non-halting setting that we consider, the output can change indefinitely as there is no fixed finishing time. This difference makes the problem for the non-halting case harder.

### 1.2 Our results

Our main result is a single statement about the interplay of the two complexities, which we state first because everything else follows from it. For $k, l \in \mathbb{N}$ we write $\mathrm{R}(F \mid k, l)$ for the conditional a priori probability of enumerating $F$ given the pair $(k, l)$ (Definition 10).

**Theorem 1** (Main dichotomy)**.** *There is a constant $d$ such that for all $k, l \in \mathbb{N}$ and every $S \subseteq \mathbb{N}$ with $\mathrm{R}(S) \geq 2^{-k}$ at least one of the following holds:*

*(i)* $\mathrm{D}(S) \leq l$*;*

*(ii) there is a* finite $F \subseteq S$ *with* $\mathrm{D}(F) > l$ *and* $\mathrm{R}(F \mid k, l) \geq 2^{-k}/d$*.*

Equivalently: if every finite $F \subseteq S$ with $\mathrm{R}(F \mid k, l) \geq 2^{-k}/d$ satisfies $\mathrm{D}(F) \leq l$, then $\mathrm{R}(S) \geq 2^{-k}$ implies $\mathrm{D}(S) \leq l$. In other words, *the only obstruction to a short deterministic description of $S$ is a finite subset of $S$ that is itself hard to describe deterministically but easy to hit randomly.* It is essential that the two parameters $k$ and $l$ appear as a condition in (ii): the machine we construct receives them as input, so the constant $d$ does not depend on them. This is what makes the theorem loss-free, and it is the point where our argument departs from Solovay's [7].

The transfer of upper bounds from finite to arbitrary sets follows at once.

**Theorem 2** (Transfer)**.** *Let $f \colon \mathbb{N} \to \mathbb{N}$ be a non-decreasing computable function such that*

$$\mathrm{D}(F) < f(k) \quad \textit{for every } k \textit{ and every } \text{finite } F \subseteq \mathbb{N} \textit{ with } \mathrm{R}(F) \geq 2^{-k}.$$

*Then there is a constant $C$ such that*

$$\mathrm{D}(S) < f\big(k + 2\log k + C\big) \quad \textit{for every } k \textit{ and every } \text{infinite } S \subseteq \mathbb{N} \textit{ with } \mathrm{R}(S) \geq 2^{-k}.$$

Specializing $f$ to a linear function we obtain a simple corollary on the transfer from finite to infinite sets, with no loss in the leading constant.

**Corollary 3.** *Let $\alpha \geq 1$ be a computable real such that $\mathrm{D}(F) \leq -\alpha \log \mathrm{R}(F) + O(\log\log \frac{1}{\mathrm{R}(F)})$ for every* finite $F \subseteq \mathbb{N}$*. Then $\mathrm{D}(S) \leq -\alpha \log \mathrm{R}(S) + O(\log\log \frac{1}{\mathrm{R}(S)})$ for every* infinite $S \subseteq \mathbb{N}$ *(with a different constant in the $O$-term).*

Together with the bound for finite sets from [8] this improves Solovay's constant 3 in (2) to 2.

**Corollary 4.** *For every $S \subseteq \mathbb{N}$ we have* $\mathrm{D}(S) \leq -2\log \mathrm{R}(S) + O(\log\log \frac{1}{\mathrm{R}(S)})$.

Finally, Theorem 1 applied to the parameters of $S$ itself says that every c.e. set is faithfully represented by one of its finite subsets.

**Corollary 5.** *For every $S \subseteq \mathbb{N}$ with $\mathrm{R}(S) > 0$ there is a finite $F \subseteq S$ with*

$$\mathrm{D}(F) \;\geq\; \mathrm{D}(S) \qquad \text{and} \qquad -\log \mathrm{R}(F) \;\leq\; -\log \mathrm{R}(S) + O\Big(\log\log \tfrac{1}{\mathrm{R}(S)}\Big).$$

In other words, the deterministic hardness of an arbitrary set is always witnessed by one of its finite subsets, and the randomized machine still hits that subset with essentially the probability it had for $S$. Theorem 1 gives this first in conditional form, namely $\mathrm{R}(F \mid k, l) \geq \mathrm{R}(S)/d$ for $k = \lceil \log(1/\mathrm{R}(S)) \rceil$ and $l = \mathrm{D}(S) - 1$; here the condition can then be removed for free, because $l \leq 2k + O(\log k)$ by Corollary 4, so that specifying the pair $(k, l)$ costs only $O(\log k)$ bits.

### 1.3 Comparison with previous work

Solovay [7] also proceeds by reducing infinite sets to finite ones, but Solovay's reduction loses an additive 1 in the constant. In order to approximate $\mathrm{R}(A)$ for an infinite $A$ one truncates the universe to $[n] = \{0, 1, \ldots, n\}$ and also terminates the machine after $t$ steps. More precisely, define $\mathrm{R}^t_n(A)$ as the probability that the optimal randomized machine enumerates a set $S$, such that $S \cap [n] = A \cap [n]$, if stopped after $t$ steps. In the case $t = \infty$, we consider the set $S$ that the machine enumerates in the limit. It can be shown that $\mathrm{R}^t_n(A) \to \mathrm{R}^\infty_n(A)$ as $t$ goes to infinity, and also that $\mathrm{R}^\infty_n(A) \to \mathrm{R}(A)$ as $n$ goes to infinity. Using these properties, we can try to find $t_n$, such that $\mathrm{R}^{t_n}_n(A)$ is a good approximation of $\mathrm{R}^\infty_n(A)$, and then we would have $\mathrm{R}^{t_n}_n(A) \to \mathrm{R}(A)$ as $n$ goes to infinity, allowing us to estimate $\mathrm{R}(A)$. However, specifying the sequence $(t_n)$ costs an extra $-\log \mathrm{R}(A)$ bits. Therefore, Solovay's result requires an extra $-\log \mathrm{R}(S)$ term:

**Theorem 6** ([7])**.** *Let $\alpha \geq 1$ be such that $\mathrm{D}(F) \leq -\alpha \log \mathrm{R}(F) + O(\log\log \frac{1}{\mathrm{R}(F)})$ for all finite $F \subseteq \mathbb{N}$. Then $\mathrm{D}(S) \leq -(\alpha+1)\log \mathrm{R}(S) + O(\log\log \frac{1}{\mathrm{R}(S)})$ for all $S \subseteq \mathbb{N}$.*

Corollary 3 removes the additive 1. Our proof avoids using the sequence $(t_n)$. Instead, we develop a different technique. For a parameter $k \in \mathbb{N}$, we search for sets $A$ such that $\mathrm{R}(A) \geq 2^{-k}$. We design another randomized machine $N$ with the property that for any finite $A' \subseteq A$, there will be a finite set $C$, such that $A' \subseteq C \subseteq A$ and $N$ enumerates $C$ with probability at least $2^{-k-O(1)}$ at some step of its execution. Essentially, $N$ witnesses that the initial problem can be reduced to the finite sets case.

The problem of finding the optimal $\alpha$ for finite sets is known to be equivalent to a purely combinatorial question about a game. Solovay's method for finite sets rests on a winning strategy in *Martin's game*, which has one parameter $\varepsilon \in (0, 1)$; writing $N(\varepsilon)$ for the least number of sets needed to win it, Solovay proved $N(\varepsilon) = O((1/\varepsilon)^2)$ and Ageev [2] proved the matching lower bound $N(\varepsilon) = \Omega((1/\varepsilon)^2)$. Thus the constant 2 cannot be improved along this route. A related game, *Cats versus Ants*, was introduced in [9]: let $K(\varepsilon)$ be the least number of cats needed to win it. Every strategy for Martin's game yields one for Cats versus Ants, so $K(\varepsilon) \leq N(\varepsilon)$, but Ageev's lower bound does not carry over; only the trivial bound $K(\varepsilon) = \Omega(1/\varepsilon)$ is known. The relevance of the game is that it controls the optimal $\alpha$ for finite sets from both sides: a computable strategy for Alice with $c(1/\varepsilon)^\alpha$ cats bounds $\alpha$ from above, and computable winning strategies for Bob bound it from below (Theorem 20 below). By Corollary 3 any computable bound on finite sets holds for arbitrary ones, so the question "how much do random bits help in non-halting Kolmogorov complexity?" becomes a combinatorial problem about cats and ants, with the answer known to lie between 1 and 2. The game is defined, and the reduction stated precisely, in Section 5.2.

### 1.4 The proof idea

The proof of Theorem 1 is naturally described in terms of cats and ants, and we will use this language throughout Section 3; the reader who prefers a formal description will find one there as well.

We think of the finite subsets of $\mathbb{N}$ as the vertices of a graph, where an enumeration process is a token moving *upward* (a set only grows). We simulate the optimal randomized machine $P$ on all outcomes of the random tape simultaneously. Let us think of the outcome $w \in \{0,1\}^{\mathbb{N}}$ of the random tape as an *ant* of infinitesimal mass, sitting at the finite set enumerated by $P$ on $w$ so far. The total mass of ants whose limit position is $S$ is $\mathrm{R}(S)$. We run, in parallel, the optimal deterministic machine $M$ on all inputs of length at most $l$. Let us think of each such input as a *cat*, sitting at the finite set it has enumerated so far. There are fewer than $2^{l+1}$ cats. Our goal is to show that if the good ants (those converging to $S$) have mass at least $2^{-k}$, then either some cat converges to $S$ — giving $\mathrm{D}(S) \le l$ — or some finite set $F \subseteq S$ has $2^{-k-O(1)}$ mass of ants and no cat in the limit.

To conduct our argument, we define a randomized machine $N$. It is defined in stages $t = 1, 2, \ldots$. At stage $t$ we run the first $t$ steps of $P$ on every outcome of the random tape and the first $t$ steps of $M$ on every input of length at most $l$; the *position* of an ant or of a cat at stage $t$ is the finite set enumerated so far by the corresponding run. Positions therefore only grow with $t$. Let us construct $N$ by referring to outcomes of its random tape as *slow ants*. In other words, each ant $w$ that corresponds to $P$ gets a *slow copy* that corresponds to $N$. The slow copy always enumerates a subset of what the original ant enumerates. Each slow copy is in one of two states at every moment: either *free* or *frozen*. Note that the behaviours of the slow copies exactly define the behaviour of $N$ and vice versa. The machine $N$ also receives $k$ and $l$ as inputs. Write $\mathrm{R}'(S) = \mathrm{R}_N(S)$ for the probability that $N$ enumerates a given set $S$. For each stage $t$, there are two rules that govern the movement of the slow copies.

*Collecting.* If some finite set $F$ has the property that free slow ants of total mass at least $2^{-k-2}$ sit below $F$ (enumerate a subset of $F$) while their fast copies have already passed above $F$ (enumerate a superset of $F$), then all these slow ants are moved to $F$ and frozen there. This action allows us to concentrate slow ants of total mass at least $2^{-k-2}$ in the set $F$.

*Scattering.* If a cat sits exactly at a set $F$ where frozen slow ants sit, they are released: each of them jumps to the current position of its fast copy and becomes free again.

Now look at a good ant. Its slow copy can be scattered infinitely often, or it can be frozen from some moment on, or it can end up free forever. For at least one good ant, the last possibility is excluded by a continuity-of-measure argument: if all good slow ants became free forever, then from some moment on for some natural $n$, a mass of at least $2^{-k-2}$ of them would sit below $S \cap [n]$ with their fast copies above $S \cap [n]$, and the collecting rule would freeze them. So pick a good ant that is scattered infinitely often or frozen from some moment on. In the first case, a single cat scatters our slow ant infinitely often; at each scattering the cat's position coincides with the slow ant's position, which is squeezed between the previous position of the fast ant and $S$; hence that cat enumerates exactly $S$ and $\mathrm{D}(S) \le l$. In the second case, the slow ant is frozen forever at some finite $F \subseteq S$; no cat ever sits at $F$ afterwards, so $\mathrm{D}(F) > l$; and the whole batch that was frozen together with it — of mass at least $2^{-k-2}$ — stays there forever, so $\mathrm{R}'(F \mid k, l) \ge 2^{-k-2}$, whence $\mathrm{R}(F \mid k, l) \ge 2^{-k}/d$ by the optimality of R.

## 2 Preliminaries

Throughout, log is the binary logarithm. Further, $\langle \cdot \rangle$ denotes a fixed self-delimiting encoding with $|\langle y \rangle| \le |y| + 2\log|y| + O(1)$ for strings and $|\langle k \rangle| \le 2\log k + O(1)$ for natural numbers. Write $[n] = \{0, 1, 2, \ldots, n\}$. We identify a natural number with its binary representation.

**Definition 7** (Deterministic enumerating complexity)**.** Let $M$ be a Turing machine with one (read only) input tape and one (write only) output tape which possibly does not halt. The input tape contains a finite string $p \in \{0,1\}^*$ padded with a symbol outside $\{0,1\}$, so the machine can detect where $p$ ends. We say that $M$ *enumerates* $S \subseteq \mathbb{N}$ on the input $p$ if the set of numbers ever written on the output tape is exactly $S$. Put $\mathrm{D}_M(S) = \min\{|p| : M \text{ enumerates } S \text{ on } p\}$ ($= \infty$ if there is no such $p$). There is an *optimal* machine $M^*$, i.e. one with $\mathrm{D}_{M^*}(S) \le \mathrm{D}_M(S) + O(1)$ for every $M$ and all $S$; we fix such an $M^*$ and put $\mathrm{D} := \mathrm{D}_{M^*}$.

**Definition 8** (Conditional deterministic complexity)**.** Let $M$ have two input tapes; a string $p$ is written on the first one and a string $y$ on the second, both padded as in Definition 7. Put $\mathrm{D}_M(S \mid y) = \min\{|p| : M \text{ enumerates } S \text{ on } (p,y)\}$. Again there is an optimal machine, and $\mathrm{D}(S \mid y)$ is defined up to an additive constant.

**Definition 9** (Randomized enumerating complexity)**.** Let $P$ be a machine with one infinite input tape and one output tape. For $w \in \{0,1\}^{\mathbb{N}}$ let $P(w)$ be the set enumerated by $P$ when $w$ is written on the input tape. With respect to the uniform measure on $\{0,1\}^{\mathbb{N}}$ put $\mathrm{R}_P(S) = \mathbf{Pr}_w[P(w) = S]$.[1] There is an optimal machine $P^*$, i.e. one with $\mathrm{R}_{P^*}(S) \ge c_P \cdot \mathrm{R}_P(S)$ for every $P$ and all $S$, with $c_P > 0$ depending only on $P$; we fix such a $P^*$ and put $\mathrm{R} := \mathrm{R}_{P^*}$. The randomized enumerating complexity of $S$ is $-\log \mathrm{R}(S)$.

**Definition 10** (Conditional randomized complexity)**.** Let $P$ have an infinite random input tape and a second input tape carrying a string $y$ (padded as in Definition 7), and put $\mathrm{R}_P(S \mid y) = \mathbf{Pr}_w[P(w,y) = S]$. Again an optimal machine exists and $\mathrm{R}(S \mid y)$ is defined up to a multiplicative constant. For natural numbers $k, l$ we write $\mathrm{R}(S \mid k, l) := \mathrm{R}(S \mid \langle k \rangle l)$.

The existence of optimal machines for all four definitions can be proven by a standard argument, see e.g. [5].

Without loss of generality, let us assume that $\mathrm{D}(S) > 0$ and $\mathrm{R}(S) \le \frac{1}{4}$ for any $S$. Indeed, since the events $\{w : P^*(w) = S\}$ are disjoint, $\sum_S \mathrm{R}(S) \le 1$, so at most four sets have $\mathrm{R}(S) > 1/4$; and only the empty input can witness $\mathrm{D}(S) = 0$, so at most one set has $\mathrm{D}(S) = 0$. All the bounds below are of the form $\mathrm{D}(S) \le \cdots + O(1)$, so for these at most five particular sets they hold after enlarging the additive constant. Thus, whenever we write $\log \mathrm{D}(S)$ or $\log\log \frac{1}{\mathrm{R}(S)}$, we have $\mathrm{D}(S) \ge 1$ and $\mathrm{R}(S) \le 1/4$, so that both expressions are defined and non-negative. All optimal machines above are fixed once and for all, and the constants in all terms $O(m)$ below depend only on the choice of optimal machines.

**Remark 11** (Notation used elsewhere)**.** Solovay [7] and [9], following Chaitin [1], write $I(A)$ for the complexity of the problem "enumerate $A$" (our $\mathrm{D}(A)$), $m(A)$ for the a priori probability of enumerating $A$ (our $\mathrm{R}(A)$), and $H(A) = \lceil -\log m(A) \rceil$ (our $-\log \mathrm{R}(A)$). One difference should be kept in mind: the complexity of [7, 9] is a *prefix* one — there $p$ is required to enumerate $A$ on every infinite extension of $p$ — whereas our D is a *plain* one, since in Definition 7 the machine is told where its input ends. The two are close: $\mathrm{D}(A) \le I(A) + O(1)$ (a machine that is told where its input ends can simulate a prefix machine on it), and $I(A) \le \mathrm{D}(A) + 2\log \mathrm{D}(A) + O(1)$ (encode the input self-delimitingly), exactly as for the plain and the prefix complexity of strings [5, 6]. Since $\mathrm{D}(A) = O\big(\log(1/\mathrm{R}(A))\big)$ by Corollary 4, the two conventions differ by $O\big(\log\log(1/\mathrm{R}(A))\big)$, so every statement in this paper is insensitive to the choice.

The following two facts are standard.

**Lemma 12.** *For all $S \subseteq \mathbb{N}$, all strings $y$ and all $k \in \mathbb{N}$:*

[1] The event $\{w : P(w) = S\}$ is measurable: for each $n$ the event "$n$ is eventually enumerated" is a countable union of cylinders, and $\{w : P(w) = S\}$ is the intersection of these events for $n \in S$ and of their complements for $n \notin S$. The same applies to all sets of tapes considered in Section 3, which are obtained from events depending on finitely many bits by countable unions and intersections.

*(a)* $\mathrm{R}(S \mid y) \;\geq\; c \cdot \mathrm{R}(S)$;

*(b)* $\mathrm{R}(S) \;\geq\; c \cdot 2^{-|\langle y \rangle|}\, \mathrm{R}(S \mid y)$; *in particular* $\mathrm{R}(S) \geq c\, k^{-2}\, \mathrm{R}(S \mid k)$;

*(c) if a string $y$ is computed from a string $x$ by an algorithm $g$, then*

$$\mathrm{R}(S \mid x) \;\geq\; 2^{-m-O(1)} \cdot \mathrm{R}(S \mid y) \qquad \textit{and} \qquad \mathrm{D}(S \mid x) \;\leq\; \mathrm{D}(S \mid y) + m + O(1),$$

*where $m$ is the length of a program computing $g$ and the constants in $O(1)$ do not depend on $g$.*

*Here $c > 0$ is an absolute constant.*

These are the obvious analogues, for enumerating complexity, of well-known properties of Kolmogorov complexity and of the a priori probability, and they are proved in exactly the same way; see [5, 6].

**Lemma 13.** $-\log \mathrm{R}(S) \leq \mathrm{D}(S) + 2 \log \mathrm{D}(S) + O(1)$ *for every* $S \subseteq \mathbb{N}$.

*Proof.* Let $p$ be a shortest input on which the machine $M^*$ of Definition 7 enumerates $S$, so that $|p| = \mathrm{D}(S)$. Consider the randomized machine $P$ that reads from its random tape a self-delimiting code $\langle p \rangle$ and then simulates $M^*$ on the input $p$ (having read $\langle p \rangle$, it knows $p$ completely and in particular knows where $p$ ends). The tapes beginning with $\langle p \rangle$ have measure $2^{-|\langle p \rangle|} \geq 2^{-|p| - 2\log|p| - O(1)}$, and $P$ enumerates $S$ on every such tape, so $\mathrm{R}_P(S) \geq 2^{-\mathrm{D}(S) - 2\log \mathrm{D}(S) - O(1)}$.

It remains to pass from $P$ to the optimal randomized machine $P^*$ of Definition 9, which has nothing to do with $M^*$: by the optimality of $P^*$ we have $\mathrm{R}(S) = \mathrm{R}_{P^*}(S) \geq c_P\, \mathrm{R}_P(S)$ for a constant $c_P > 0$ depending only on $P$, which is the assertion of the lemma. □

# 3 Proof of the main dichotomy

Fix $k, l \in \mathbb{N}$ and let $S \subseteq \mathbb{N}$ satisfy $\mathrm{R}(S) \geq 2^{-k}$. If $S$ is finite the statement is trivial: either $\mathrm{D}(S) \leq l$, or $F := S$ witnesses (ii), because $\mathrm{R}(S \mid k, l) \geq c\, \mathrm{R}(S) \geq c 2^{-k}$ by Lemma 12(a), and it suffices to take $d \geq 1/c$. So assume from now on that $S$ is infinite.

## 3.1 The process

We describe a randomized machine $N$, as in Definition 10, which receives $\langle k \rangle l$ on the second input tape. On its output tape the machine writes its *output so far*. On its work tape it maintains, among other things, bits $\sigma(w) \in \{\mathsf{free}, \mathsf{frozen}\}$.

Besides, the machine $N$ simulates on its work tape the optimal randomized machine $P^*$ from Definition 9 and the optimal deterministic machine $M^*$ from Definition 7 on all inputs $p$ of length at most $l$ in a dovetail fashion. This means the following.

Both machines $P^*, M^*$ work in steps. Let $P^{*\,t}(w)$ be the finite set enumerated by $P^*$ on random input $w$ within $t$ steps. The machine reads at most one bit of its random tape per step, so $P^{*\,t}(w)$ depends only on $w_1 \ldots w_t$ , and, therefore, can be simulated. Likewise for a string $p$ of length at most $l$ let $M^{*\,t}(p)$ denote the set of natural $n$ printed by $M^*$ on the input $p$ within $t$ steps. The set $M^{*\,t}(p)$ does not decrease as $t$ increases and its union over all $t$ is $M^*(p)$, the set enumerated by $M^*$ on $p$.

The machine $N$ works in stages $t = 0, 1, \ldots$. On stage $t$ it simulates $t$ steps of $P^*$ on all $w$ and $t$ steps of $M^*$ on all inputs $p$ of length at most $l$. As already pointed out, the action of $P^*$ within $t$ steps depends only on the length-$t$ prefix of $w$, so these simulations can be performed in finite time. We call $w$, viewed as a point of the probability space, a *fast ant* sitting at $P^{*\,t}(w)$ and call $p$ a *cat* sitting at $M^{*\,t}(p)$. There are fewer than $2^{l+1}$ cats.

In a similar way we will denote by $N^t(w)$ the output of $N$ by the end of stage $t$ on random input $w$. This set $N^t(w)$ is for us to define. It must be finite and must not decrease as $t$ increases.

The pair $(N^t(w), \sigma_t(w))$ will be viewed as the position and state of the *slow ant* $w$. Initially $N^0(w) = \varnothing$ and $\sigma_0(w) = \mathsf{free}$ for all $w$. At each stage $t = 1, 2, \ldots$ the machine $N$ advances the simulations of $P^*, M^*$ by one step, and then performs the following two operations in this order. Neither operation depends on the random input of $N$.

At the beginning of stage $t$ set $B(w) := N^{t-1}(w)$ and $\sigma(w) := \sigma_{t-1}(w)$ for all $w$. These are the current position and state of the slow ant $w$; both operations below change them, and their values at the end of the stage will be taken as $N^t(w)$ and $\sigma_t(w)$.

**Scattering.** For every finite set $F$ such that $M^{*\,t}(p) = F$ for some cat $p$: for every $w$ such that $\sigma(w) = \mathsf{frozen}$ and $N^{t-1}(w) = F$, set $B(w) := P^{*\,t}(w)$ and $\sigma(w) := \mathsf{free}$. (If a cat sits exactly at a set $F$ where the frozen slow ant $w$ sits, the ant $w$ is released: it jumps to the current position of its fast copy and becomes free again.)

**Collecting.** Do the following loop. While there exists a finite set $F$ with

$$\mathbf{Pr}\big[\, w :\ \sigma(w) = \mathsf{free},\ B(w) \subseteq F \subseteq P^{*\,t}(w) \,\big] \ \geq\ 2^{-k-2}, \tag{3}$$

pick such an $F$ and, for every $w$ counted in (3), set $B(w) := F$ and $\sigma(w) := \mathsf{frozen}$. It might happen that there are several such $F$; it does not matter which one is picked. (If some finite set $F$ has the property that free slow ants of total mass at least $2^{-k-2}$ sit below $F$ while their fast copies have already passed above $F$, then all these slow ants are moved to $F$ and frozen there.)

Note that this loop can be iterated only finitely many times. Indeed, each iteration turns a set of free slow ants of mass at least $2^{-k-2}$ into frozen ones, and no ant is unfrozen during the loop. As the total mass is 1, there are at most $2^{k+2}$ iterations. Also note that neither operation depends on $N$'s own random input: the work of $N$ is almost independent of it, and the only thing that does depend on it is the output of $N$.

Once both operations have been done, we set $\sigma_t(w) := \sigma(w)$; then the machine $N$ reads the $t$-th bit of its own random input $w$ and prints on its output tape all numbers from $B(w) \setminus N^{t-1}(w)$. Obviously we have $N^{t-1}(w) \subseteq B(w)$, so the set enumerated by the end of the stage is $N^t(w) = B(w)$. Note that both $B(w)$ and $\sigma(w)$ depend only on length-$t$ prefix of $w$, thus there is no need to read more random bits.

Thus $N$ enumerates, on the random tape $w$, the set $N(w) = \bigcup_t N^t(w)$. We record the elementary properties of the process.

**Lemma 14.** *(a) $N^t(w) \subseteq P^{*\,t}(w)$ for all $t, w$, and $N^t(w)$ is non-decreasing in $t$; hence $N$ is a legitimate enumerator and $N(w) \subseteq P^*(w)$.*

*(b) For every $t$, the functions $w \mapsto (P^{*\,t}(w), N^t(w), \sigma_t(w))$ depend only on $w_1 \ldots w_t$. In particular all the probabilities in (3) are computable rationals with denominator $2^t$, and the process is computable given $k$ and $l$.*

*Proof.* (a) At a stage $t$ the position of a slow ant either stays the same or is changed by one of the two operations, and we check the claim in all three cases. If the ant is not moved at all, its position remains $N^{t-1}(w) \subseteq P^{*\,(t-1)}(w) \subseteq P^{*\,t}(w)$ by the induction hypothesis and the monotonicity of $P^{*\,t}(w)$ in $t$. Scattering puts the ant at $P^{*\,t}(w)$ itself, and collecting puts it at a set $F$ with $B(w) \subseteq F \subseteq P^{*\,t}(w)$. So in each case the new position contains the old one and is contained in $P^{*\,t}(w)$, and the operations of a single stage can only be applied one after another, each of them preserving both properties.

(b) Induction on $t$. The set $P^{*\,t}(w)$ depends only on the first $t$ bits of $w$ because $P^*$ reads at most one bit per step. The conditions in the two operations are stated in terms of $P^{*\,t}$,

$B$, $\sigma$, $N^{t-1}$ and the (deterministic) positions of the cats, and both operations act identically on all $w$ satisfying the same condition. Hence the property is preserved. Consequently every set occurring in (3) is a finite union of cylinders of rank $t$. Note also that $F$ has to satisfy $F \subseteq P^{*\,t}(w)$ for at least one $w$, so it is a subset of the finite set $\bigcup_{|u|=t} P^{*\,t}(u)$, and the loop searches through finitely many candidates. □

**Remark 15.** Formally, a machine cannot store the uncountable objects $N(\cdot)$, $\sigma(\cdot)$. By Lemma 14(b) it does not have to: at stage $t$ it keeps the finite table of the values of $(B, \sigma)$ on the $2^t$ cylinders of rank $t$, and when passing from $t$ to $t+1$ it copies the value at a cylinder $u$ to the two cylinders $u0$ and $u1$. On the random input $w$ actually supplied to $N$, the machine prints the elements of $N(w)$ as they appear.

Call an ant $w$ (fast or slow) *good* if $P^*(w) = S$. By assumption the good fast ants have mass $\mathrm{R}(S) \geq 2^{-k}$. Note that a slow ant is scattered only if it is frozen, and it is frozen only by collecting, which requires it to be free; so the states alternate,

$$\mathsf{free} \underset{\text{scattering}}{\overset{\text{collecting}}{\rightleftharpoons}} \mathsf{frozen}$$

(collecting freezes a free ant, scattering frees a frozen one). Within a single stage $t$ an ant can change its state twice, from frozen to free by scattering and back to frozen by collecting, but no more: the two operations are performed once each and in this order. In particular, an ant with $\sigma_{t-1}(w) = \mathsf{free}$ is not touched by scattering, so during stage $t$ its state either does not change at all or changes once, from free to frozen; we use this in the proof of Lemma 16.

Each slow ant satisfies exactly one of:

(A) it is scattered (and collected) infinitely often;

(B) from some moment on it is frozen forever;

(C) from some moment on it is free forever.

Indeed, if $w$ is scattered finitely often then after the last scattering it is free, and it can be frozen at most once more (a second freezing would require an intervening scattering); so it stays free forever or is frozen forever. In case (C) the slow ant never moves after the last scattering, because both operations that move it change its state.

### 3.2 Not all good ants are eventually free

**Lemma 16.** *Not all good slow ants are in case (C).*

*Proof.* Assume they all are. Then each good slow ant $w$ has a limit position $N(w)$, a finite subset of $S$ by Lemma 14(a), which it reaches at some finite moment and never leaves, while being free from that moment on.

For $n \in \mathbb{N}$ let $G_n := \{w \text{ good} : N(w) \subseteq S \cap [n]\}$ . As shown above, the limit position $N(w)$ of every good slow ant is a finite subset of $S$ and hence is contained in $S \cap [n]$ for some $n$; that is, the union of the sequence $(G_n)$ is equal to the set of all good ants, of mass at least $2^{-k}$. Since the sequence $(G_n)$ is non-decreasing, by continuity of measure there is an $n$ with

$$\mathbf{Pr}[G_n] \;\geq\; 2^{-k-1}.$$

Fix this $n$ and put $F := S \cap [n]$. For $t \in \mathbb{N}$ let

$$E_t := \{w \in G_n : \text{ for all } t' \geq t,\ P^{*\,t'}(w) \supseteq F,\ \sigma_{t'}(w) = \mathsf{free} \text{ and } N^{t'}(w) = N(w)\}.$$

For every $w \in G_n$ we have $P^{*\,t'}(w) \uparrow S \supseteq F$ and, by our assumption, $w$ is eventually free; hence $w \in E_t$ for all large $t$. Thus $(E_t)$ is non-decreasing with union $G_n$, and by continuity of measure there is a $t_0$ with

$$\mathbf{Pr}[E_{t_0}] \;\geq\; 2^{-k-2}.$$

Consider the collecting loop at any stage $t \geq t_0$ and look at its final state, when no $F'$ satisfies (3). By the definition of $E_{t_0}$, every $w \in E_{t_0}$ is free at the end of stage $t$ and sits at $N(w)$. The loop only freezes ants and never frees them, so an ant free at the end of the loop was free throughout it and was never moved by it; note that this argument runs backwards from the end of the stage, so it applies to $t = t_0$ as well, although $\sigma_{t_0}(w) = \mathsf{free}$ is only known at the end of stage $t_0$. Thus at the end of the loop every $w \in E_{t_0}$ is free and its current position satisfies $B(w) = N(w) \subseteq S \cap [n] = F \subseteq P^{*\,t}(w)$. Hence $F = S \cap [n]$ satisfies (3) with mass at least $\mathbf{Pr}[E_{t_0}] \geq 2^{-k-2}$, so the loop could not have stopped — a contradiction. □

### 3.3 A good ant scattered infinitely often

**Lemma 17.** *If some good slow ant is in case (A), then* $\mathrm{D}(S) \leq l$.

*Proof.* Let $w_0$ be such an ant and let $t_1 < t_2 < \cdots$ be the stages at which it is scattered. The scattering at stage $t$ requires a cat sitting at the position the slow ant had at the end of stage $t-1$; as there are fewer than $2^{l+1}$ cats, by the pigeonhole principle there is a single cat $p$ (thus $|p| \leq l$) and an infinite subsequence, which we again denote $t_1 < t_2 < \cdots$, such that

$$M^{*\,t_i}(p) \;=\; N^{t_i-1}(w_0) \;=:\; F_i \qquad (i = 1, 2, \dots).$$

Immediately after the scattering at stage $t_i$ the slow ant sits at $P^{*\,t_i}(w_0)$, and its position never decreases, so

$$F_{i+1} \;=\; N^{t_{i+1}-1}(w_0) \;\supseteq\; P^{*\,t_i}(w_0).$$

Since $w_0$ is good, $P^{*\,t}(w_0) \uparrow S$, and therefore $\bigcup_i F_i \supseteq \bigcup_i P^{*\,t_i}(w_0) = S$. On the other hand $F_i = N^{t_i-1}(w_0) \subseteq P^{*\,t_i-1}(w_0) \subseteq S$ by Lemma 14(a). Hence

$$\bigcup_i F_i \;=\; S.$$

It remains to note that the cat $p$ enumerates exactly $S$. Indeed, $M^*(p) = \bigcup_t M^{*\,t}(p) \supseteq \bigcup_i F_i = S$; conversely, for every $t$ there is $t_i > t$, and by monotonicity $M^{*\,t}(p) \subseteq M^{*\,t_i}(p) = F_i \subseteq S$, so $M^*(p) \subseteq S$. Thus $M^*(p) = S$ and $\mathrm{D}(S) \leq |p| \leq l$. □

### 3.4 A good ant frozen forever

**Lemma 18.** *If some good slow ant is in case (B), then there is a finite* $F \subseteq S$ *with* $\mathrm{D}(F) > l$ *and* $\mathrm{R}_N(F \mid k, l) \geq 2^{-k-2}$.

*Proof.* Let $w_0$ be such an ant, let $t_0$ be the stage of its last freezing and let $F := N^{t_0}(w_0)$ be its position from $t_0$ on. Then $F$ is a finite subset of $S$ by Lemma 14(a).

First, $\mathrm{D}(F) > l$. Indeed, assume that $M^*(p) = F$ for some $p$ with $|p| \leq l$; then $p$ is one of the cats. As $F$ is finite and $M^{*\,t}(p)$ is non-decreasing with union $F$, we have $M^{*\,t}(p) = F$ for all sufficiently large $t$ — so the cat $p$ sits at $F$ at some stage $t \geq t_0 + 1$. The scattering rule at stage $t$ would release $w_0$, which sits at $F$ at the end of stage $t-1$, a contradiction.

Second, consider the collecting operation which froze $w_0$ at stage $t_0$. It froze at $F$ the whole set $W := \{w : \sigma(w) = \mathsf{free},\ B(w) \subseteq F \subseteq P^{*\,t_0}(w)\}$, where $B$ and $\sigma$ are the current values immediately before that operation; the mass of $W$ is at least $2^{-k-2}$ by (3). Every $w \in W$ sits at $F$ in the frozen state after stage $t_0$, and can be released only by a cat sitting at $F$. That never happens: a cat sitting at $F$ at a stage $t \geq t_0 + 1$ would, by the scattering rule, release $w_0$

as well, contradicting the choice of $w_0$. (Note that this does not follow from $\mathrm{D}(F) > l$ alone: a cat may visit $F$ in passing, with $M^{*t}(p) = F$ at some stage and $M^*(p) \supsetneq F$ in the limit.) Hence $N^t(w) = F$ for all $t \geq t_0$ and all $w \in W$, so $N(w) = F$ and $N$ enumerates $F$ on every $w \in W$:

$$\mathrm{R}_N(F \mid k, l) \;\geq\; \mathbf{Pr}[W] \;\geq\; 2^{-k-2}. \qquad \square$$

### 3.5 Putting the pieces together

*Proof of Theorem 1.* The case of finite $S$ was settled at the beginning of the section, so let $S$ be infinite with $\mathrm{R}(S) \geq 2^{-k}$ and run the process of Section 3.1 with the parameters $k, l$. By Lemma 16 some good slow ant is in case (A) or in case (B).

If some good slow ant is in case (A), then $\mathrm{D}(S) \leq l$ by Lemma 17 and alternative (i) holds.

Otherwise some good slow ant is in case (B), and Lemma 18 provides a finite $F \subseteq S$ with $\mathrm{D}(F) > l$ and $\mathrm{R}_N(F \mid k, l) \geq 2^{-k-2}$. The machine $N$ is one fixed machine which receives $k$ and $l$ on its second input tape; hence by optimality of $\mathrm{R}(\cdot \mid \cdot)$ there is an absolute constant $c_N > 0$ with $\mathrm{R}(F \mid k, l) \geq c_N \, \mathrm{R}_N(F \mid k, l) \geq c_N 2^{-k-2}$. With $d := \max(4/c_N,\, 1/c)$ — the second bound covering the case of finite $S$ settled at the beginning of the section — we get $\mathrm{R}(F \mid k, l) \geq 2^{-k}/d$, which is alternative (ii). $\square$

## 4 Transfer from finite sets to all sets

*Proof of Theorem 2.* Let $f$ be non-decreasing and computable and assume $\mathrm{D}(F) < f(k)$ for all $k$ and all finite $F$ with $\mathrm{R}(F) \geq 2^{-k}$. Let $C$ be a constant to be chosen at the end of the proof, and given $k$ put

$$l \;:=\; f(k + 2\log k + C) - 1.$$

(We ignore rounding: $k + 2\log k + C$ is to be read as $k + \lceil 2\log k \rceil + C$.) Note that $l$ is computed from $k$ by an algorithm $g$, because $f$ is computable and $C$ is a constant; let $m$ be the length of a program computing $g$.

Let $S$ be a set with $\mathrm{R}(S) \geq 2^{-k}$; we must show $\mathrm{D}(S) < f(k + 2\log k + C)$, i.e. $\mathrm{D}(S) \leq l$. Suppose not, and apply Theorem 1 to $S$ with the parameters $k$ and $l$. Alternative (i) fails, so there is a finite $F \subseteq S$ with

$$\mathrm{D}(F) > l \qquad \text{and} \qquad \mathrm{R}(F \mid k, l) \;\geq\; 2^{-k}/d.$$

We now remove the condition. Since $l = g(k)$, Lemma 12(c) gives $\mathrm{R}(F \mid k) \geq 2^{-m-O(1)} \, \mathrm{R}(F \mid k, l)$, and then Lemma 12(b) gives

$$\mathrm{R}(F) \;\geq\; c\, k^{-2} \, \mathrm{R}(F \mid k) \;\geq\; c\, 2^{-m-O(1)} \, k^{-2} \, 2^{-k}/d \;\geq\; 2^{-(k + 2\log k + c_1)},$$

where $c_1 = m + O(1)$; it depends only on $f$ and $C$ (through the algorithm $g$), but not on $k$, $l$, $S$ or $F$. By the hypothesis of the theorem applied to the finite set $F$ with the parameter $k + 2\log k + c_1$,

$$\mathrm{D}(F) \;<\; f(k + 2\log k + c_1) \;\leq\; f(k + 2\log k + C) \;=\; l + 1,$$

provided $C \geq c_1$, where we used that $f$ is non-decreasing. This contradicts $\mathrm{D}(F) > l$.

It remains to check that $C$ can indeed be chosen with $C \geq c_1$. We have $c_1 = m + O(1)$, and $m$ — the length of a program computing $k \mapsto f(k + 2\log k + C) - 1$ — is $a_f + O(\log C)$, where $a_f$ depends on $f$ alone and the term $O(\log C)$ accounts for writing the constant $C$ inside the program. Hence $c_1 \leq a'_f + b \log C$ for an absolute constant $b$, and any sufficiently large $C$ satisfies $C \geq a'_f + b \log C$. $\square$

*Proof of Corollary 3.* Let $\beta, \gamma$ be constants with $\mathrm{D}(F) \le \alpha \log \frac{1}{\mathrm{R}(F)} + \beta \log\log \frac{1}{\mathrm{R}(F)} + \gamma$ for all finite $F$, and put

$$f(k) \;:=\; \lceil \alpha k \rceil + \lceil \beta \log k \rceil + \gamma + 1.$$

This is a non-decreasing computable function (a fixed function, even though we may not know $\alpha$, $\beta$ and $\gamma$), and for a finite $F$ with $\mathrm{R}(F) \ge 2^{-k}$ we have $\log \frac{1}{\mathrm{R}(F)} \le k$, hence $\mathrm{D}(F) < f(k)$. By Theorem 2, every $S$ with $\mathrm{R}(S) \ge 2^{-k}$ satisfies

$$\mathrm{D}(S) \;<\; f(k + 2\log k + c) \;=\; \alpha k + O(\log k).$$

Applying this with $k := \lceil \log(1/\mathrm{R}(S)) \rceil$ gives $\mathrm{D}(S) \le \alpha \log \frac{1}{\mathrm{R}(S)} + O\big(\log\log \frac{1}{\mathrm{R}(S)}\big)$. □

*Proof of Corollary 4.* By [8], $\mathrm{D}(F) \le 2\log \frac{1}{\mathrm{R}(F)} + O(\log\log \frac{1}{\mathrm{R}(F)})$ for every finite $F$ (the bound there is stated for the prefix version of D, which is an upper bound for ours by Remark 11). Apply Corollary 3 with $\alpha = 2$. □

*Proof of Corollary 5.* Let $\mathrm{R}(S) > 0$; then $\mathrm{D}(S) < \infty$ (this is the theorem of [3], and also follows from Corollary 4). If $\mathrm{D}(S) = 0$ — which happens for at most one set — we may take $F = \varnothing$: then $\mathrm{D}(F) \ge 0 = \mathrm{D}(S)$ and $-\log \mathrm{R}(F) = O(1)$, since $\mathrm{R}(\varnothing) > 0$. So assume $\mathrm{D}(S) \ge 1$. Put $l := \mathrm{D}(S) - 1$ and $k := \lceil \log(1/\mathrm{R}(S)) \rceil$, so that $2^{-k} \le \mathrm{R}(S) < 2^{-k+1}$. Apply Theorem 1 with these $k, l$. Alternative (i) would say $\mathrm{D}(S) \le \mathrm{D}(S) - 1$, so alternative (ii) holds and yields a finite $F \subseteq S$ with $\mathrm{D}(F) > \mathrm{D}(S) - 1$, i.e. $\mathrm{D}(F) \ge \mathrm{D}(S)$, and $\mathrm{R}(F \mid k, l) \ge 2^{-k}/d_0 > \mathrm{R}(S)/(2d_0)$, where $d_0$ is the constant of Theorem 1.

It remains to remove the condition. By Corollary 4 we have $l < \mathrm{D}(S) \le 2k + O(\log k)$, so the pair $(k, l)$ has a self-delimiting encoding of length $2\log k + 2\log l + O(1) = O(\log k)$, and Lemma 12(b) gives

$$\mathrm{R}(F) \;\ge\; c\, 2^{-O(\log k)}\, \mathrm{R}(F \mid k, l) \;\ge\; \mathrm{R}(S)/k^{O(1)}.$$

Since $k = \lceil \log(1/\mathrm{R}(S)) \rceil$, this is the second inequality of the corollary. □

# 5 Concluding remarks

## 5.1 What the new proof avoids

Solovay's reduction and its subsequent refinements approximate $\mathrm{R}(A)$ for an infinite $A$ by truncated quantities $\mathrm{R}^t_n(A)$ — the probability that after $t$ steps the machine has enumerated $A \cap [n]$ and nothing outside $A$ — and must include in the deterministic description a sequence of times $(t_n)$ at which these approximations are reliable. That sequence costs $-\log \mathrm{R}(A)$ extra bits, which is exactly the loss in Theorem 6. In the proof above no such sequence is needed: the freezing mechanism makes the construction wait for the probability mass instead of being told when to trust it, and the price is reduced to the two numbers $k$ and $l$, supplied as a condition. Removing that condition costs only $O(\log k)$ bits by Lemma 12: in the proof of Theorem 2 because there $l$ is computable from $k$, and in Corollary 5 because there $l = O(k)$. It is also worth noting that the argument never needs to know $\mathrm{R}(A)$, nor to enumerate the sets $A$ under consideration.

## 5.2 The Cats versus Ants game

For completeness we recall the game of [9] and the exact form of the reduction, in the notation of the present paper.

**Definition 19** (Ants versus cats)**.** The game is specified by a rational $\varepsilon \in (0,1)$ with $1/\varepsilon$ integer and by a number $k$ of cats. The nodes are the finite subsets of $\mathbb{N}$, with an arc from $A$ to $B$ whenever $A \subsetneq B$. The ants form an infinitely divisible unit mass; the $k$ cats are indivisible. At the start all the ants and all the cats are at the node $\varnothing$. Bob moves the ants, Alice moves the cats, and they move alternately, Bob first. In his turn Bob may perform finitely many actions of the form "move any portion of the ants from a node $A$ to a node $B$ reachable from $A$"; in her turn Alice may perform finitely many actions of the form "move a cat from a node $A$ to a node $B$ reachable from $A$". The game lasts forever, and *Alice wins* if after each of her moves every node carrying at least $\varepsilon$ of the ants contains a cat.

Let $K(\varepsilon)$ be the least $k$ for which Alice has a computable winning strategy in the $(\varepsilon, k)$-game, the strategy receiving $\varepsilon$ as input.

Note that a position of the game is a finite object: only finitely many nodes carry ants or cats at any moment, and the ant mass on each of them is a rational number, since each move redistributes finitely many portions. The infinite binary strings of Section 3, which index the ants there, play no role here.

Bob wins whenever $k < 1/\varepsilon$: on his first move he puts a portion $\varepsilon$ of the ants in each of the nodes $\{1\}, \dots, \{1/\varepsilon\}$, and Alice has too few cats. Hence $K(\varepsilon) \geq 1/\varepsilon$; in fact $1/\varepsilon$ cats do not suffice either, as Bob wins the $(1/4, 4)$-game [9]. On the other hand Alice wins with $O(1/\varepsilon^2)$ cats, by a computable strategy adapted from Martin's strategy [7, 9]; so

$$1/\varepsilon \;<\; K(\varepsilon) \;=\; O\big(1/\varepsilon^2\big).$$

What happens for $k \approx (1/\varepsilon)^\alpha$ with $1 < \alpha < 2$ is unknown. The game is tied to our problem as follows.

**Theorem 20** ([9])**.** *Let $\alpha \geq 1$ be rational.*

*(1) If for some constant $c$ Alice has a computable strategy that, given $\varepsilon$, wins the $(\varepsilon, k)$-game with some $k \leq c\,(1/\varepsilon)^\alpha$ for all $\varepsilon$, then $\mathrm{D}(F) \leq \alpha \log \frac{1}{\mathrm{R}(F)} + O\big(\log\log \frac{1}{\mathrm{R}(F)}\big)$ for every finite $F \subseteq \mathbb{N}$.*

*(2) Conversely, if for some $\delta > 0$ and arbitrarily small $\varepsilon$ Bob has a computable winning strategy in the $(\varepsilon, \delta(1/\varepsilon)^\alpha)$-game, then $\mathrm{D}(F) \geq \alpha \log \frac{1}{\mathrm{R}(F)} - O(\log \mathrm{D}(F))$ for infinitely many finite $F \subseteq \mathbb{N}$.*

Thus the game bounds the optimal constant for finite sets from both sides: by (1) every $\alpha$ for which Alice wins computably is an upper bound, and by (2) every $\alpha$ for which Bob wins computably is a lower bound. Note that if Bob has a winning strategy, then he also has a computable winning strategy. This follows from the fact that his strategy wins in a finite number of moves (if Alice can prolong the game forever, then this is not a valid winning strategy for Bob). This argument requires that in every position Alice have only finitely many essentially different moves, and this is indeed the case. Let $U$ be the union of all nodes carrying ants after Bob's move. A cat at a node $B \not\subseteq U$ covers no node carrying ants, and every node reachable from $B$ is also reachable from $B \cap U$. Hence moving a cat to $B \cap U$ instead of $B$ never hurts Alice, and we may assume that she moves cats only to subsets of $U$ (then all cats always sit at subsets of $U$, so such moves are legal). Thus Alice has only finitely many moves in each position, and by König's lemma every winning strategy for Bob wins within a bounded number of moves, that is, it is a finite object. Note that Bob has infinitely many moves, but they are finite objects (finitely many portions with rational masses) and can be effectively enumerated; König's lemma is applied to the tree of plays consistent with a fixed strategy of Bob, which branches only at Alice's moves. By bruteforcing all strategies that win in a finite time, Bob can find the winning one. Therefore the requirement of computability in Theorem 20(2) can be dropped: it suffices that for some $\delta > 0$ and arbitrarily small $\varepsilon$ Bob has a winning strategy

in the $(\varepsilon, \delta(1/\varepsilon)^\alpha)$-game. However, we do not know if every winning strategy of Alice can be made computable. Since the game is determined (the winning condition for Alice is closed), this is the only obstacle: if Alice, whenever she can win with $c\,(1/\varepsilon)^\alpha$ cats for all $\varepsilon$, could also do this by a single computable strategy receiving $\varepsilon$ as input, the two bounds would meet. As it stands, we cannot conclude that they meet.

Our transfer theorem shows that the game governs *all* sets, not only finite ones:

**Corollary 21.** *If Alice has a computable strategy as in Theorem 20(1) for a rational $\alpha \geq 1$, then $\mathrm{D}(S) \leq \alpha \log \frac{1}{\mathrm{R}(S)} + O\big(\log\log \frac{1}{\mathrm{R}(S)}\big)$ for every $S \subseteq \mathbb{N}$, finite or infinite.*

*Proof.* Combine Theorem 20(1) with Corollary 3. □

For $\alpha = 2$ this is a second route to Corollary 4: Alice's strategy with $O(1/\varepsilon^2)$ cats gives the bound for finite sets, and Theorem 2 carries it over to all sets. In particular, any improvement of Alice's strategy would immediately improve the bound for arbitrary sets.

### 5.3 The remaining question

By Corollary 3 the optimal constant $\alpha$ is the same for finite and for arbitrary sets, and by Theorem 20 it is bounded above by every exponent for which Alice has computable winning strategies and below by every exponent for which Bob has winning strategies. We know $1 \leq \alpha \leq 2$. The upper bound 2 comes from Martin's game, where the analogous quantity satisfies $N(\varepsilon) = \Theta((1/\varepsilon)^2)$ [7, 2], so no improvement can come from that game; for Cats versus Ants only the trivial lower bound is known. Determining $\alpha$ — in particular deciding whether $\alpha = 1$, i.e. whether randomness helps only by an additive logarithmic term, as it does in the halting case — remains open.

## 6 AI disclosure

In preparing this paper the authors used a large language model (Claude, Anthropic) to speed up the writing: it was used to draft and polish the English prose, to help with LaTeX formatting, and to check the text for consistency of notation and for typos. All results of the paper — the definitions, theorems and their proofs — were obtained by the authors themselves without any use of generative models. All text produced with the assistance of the model was checked by the authors, who take full responsibility for its correctness.

## References


[1] G. J. Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM* 22(3):329–340, 1975.

[2] M. Ageev. Martin's game: a lower bound for the number of sets. *Theoretical Computer Science* 289(1):871–876, 2002.

[3] K. de Leeuw, E. F. Moore, C. E. Shannon, N. Shapiro. Computability by probabilistic machines. In *Automata Studies*, pages 183–212, Princeton University Press, 1956.

[4] L. A. Levin. Laws of information conservation (non-growth) and aspects of the foundation of probability theory. *Problems of Information Transmission* 10(3):206–210, 1974.

[5] M. Li, P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer, 3rd edition, 2008.

[6] A. Shen, V. A. Uspensky, N. Vereshchagin. *Kolmogorov Complexity and Algorithmic Randomness.* Mathematical Surveys and Monographs, vol. 220, American Mathematical Society, 2017.

[7] R. M. Solovay. On random r.e. sets. In *Non-Classical Logics, Model Theory and Computability*, volume 89 of *Studies in Logic and the Foundations of Mathematics*, pages 283–307, North-Holland, 1977.

[8] N. K. Vereshchagin. Kolmogorov complexity of enumerating finite sets. *Information Processing Letters* 103(1):34–39, 2007.

[9] N. K. Vereshchagin. Kolmogorov complexity and games. *Bulletin of the EATCS* (94), 2008.



ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il