Kolmogorov Complexity, Complexity Cores, and the Distribution of Hardness★

David W. Juedes and Jack H. Lutz

Department of Computer Science
Iowa State University
Ames, IA 50011, USA
juedes@iastate.edu, lutz@iastate.edu

Abstract. Problems that are complete for exponential space are provably intractable and known to be exceedingly complex in several technical respects. However, every problem decidable in exponential space is efficiently reducible to every complete problem, so each complete problem must have a highly organized structure. The authors have recently exploited this fact to prove that complete problems are, in two respects, *unusually simple* for problems in expontential space. Specifically, every complete problem must have ususually small complexity cores and unusually low space-bounded Kolmogorov complexity. It follows that the complete problems form a negligibly small subclass of the problems decidable in exponential space. This paper explains the main ideas of this work.

1 Introduction

It is well understood that an object that is complex in one sense may be simple in another. In this paper we show that every decision problem that is complex in one standard, complexity-theoretic sense *must be* unusually simple in two other such senses.

Throughout this paper, the terms "problem," "decision problem," and "language" are synonyms and refer to a set $A \subseteq \{0,1\}^*$, i.e., a set of binary strings. The three notions of complexity considered are completeness (or hardness) for a complexity class, space-bounded Kolmogorov complexity, and the existence of large complexity cores. (All terms are defined and discussed in §§2-6 below, so this paper is essentially self-contained.) In a certain setting, we prove that every problem that is complete for a complexity class must have unusually low space-bounded Kolmogorov complexity and unusually small complexity cores. Thus complexity in one sense *implies* simplicity in another.

To be specific, we work with the complexity class ESPACE = DSPACE(2^{linear}). There are two related reasons for this choice. First, ESPACE

★ This research was supported in part by National Science Foundation Grants CCR-8809238 and CCR-9157382 and in part by DIMACS, where the second author was a visitor while part of this work was carried out.

has a rich, well-behaved structure that is well enough understood that we can prove absolute results, unblemished by oracles or unproven hypotheses. In particular, much is known about the distribution of Kolmogorov complexities in ESPACE [Lut92a, §4 below], while very little is known at lower complexity levels. Second, the structure of ESPACE is closely related to the structure of polynomial complexity classes. For example, Hartmanis and Yesha [HY84] have shown that

$$\mathrm{E} \subsetneqq \mathrm{ESPACE} \Longleftrightarrow \mathrm{P} \subsetneqq \mathrm{P/Poly} \cap \mathrm{PSPACE}.$$

This, together with the first reason, suggests that the separation of P from PSPACE might best be achieved by separating E from ESPACE. We thus seek a detailed, quantitative account of the structure of ESPACE.

For simplicity of exposition, we work with polynomial time, many-one reducibility ("$\leq_m^P$-reducibility"), introduced by Karp[Kar72]. Problems that are $\leq_m^P$-complete for ESPACE have been exhibited by Meyer and Stockmeyer [MS72], Stockmeyer and Chandra[SC89], and others. Such problems are correctly regarded as exceedingly complex. They are provably intractable in terms of computational time and space. They have exponential circuit-size complexity [Kan82], weakly exponential space-bounded Kolmogorov complexity [Huy86], and dense complexity cores [OS86, Huy87]. Problems that are $\leq_m^P$-hard for ESPACE have all these properties and need not even be recursive.

Notwithstanding these lower bounds on the complexity of $\leq_m^P$-hard problems for ESPACE, we will prove in §6 below that such problems are *unusually simple* in two respects. The word "unusually" here requires some explanation.

Suppose that we choose a language $A \subseteq \{0,1\}^*$ probabilistically, according to a random experiment in which an independent toss of a fair coin is used to decide membership of each string $x \in \{0,1\}^*$ in A. For a set X of languages, let $\Pr(X) = \Pr_A[A \in X]$ denote the probability that $A \in X$ (" the probability that event X occurs") in this experiment, provided that this probability exists. (All sets X of languages considered in this paper are Lebesque measurable, so that $\Pr(X)$ is well-defined. Thus we will not concern ourselves with issues of measurability.) If the event X has the property that $\Pr(X) = 1$, then we say that *almost every* language $A \subseteq \{0,1\}^*$ is in X. In such a case, the complement X^c of X has probability $\Pr(X^c) = 0$, so it is *unusual* for a language A to be in X^c. In particular, a language A is *unusually simple* in the sense of a given complexity measure if there is a lower complexity bound that holds for almost all languages but does not hold for A.

This probabilistic notion of "almost every" and "unusual" is intuitive and suggestive of our intent, but is not strong enough for our purposes. As we have noted, we seek to understand the structure of ESPACE. Accordingly, we will prove in §6 below that $\leq_m^P$-hard problems for ESPACE are *unusually simple for problems in* ESPACE in two specific senses. This means that, in each of these senses, there is a lower complexity bound that holds for almost every language in ESPACE but does not hold for languages that are $\leq_m^P$-hard for ESPACE. This immediately yields a quantitative result on the distribution of $\leq_m^P$-complete problems in ESPACE: Almost every language in ESPACE fails to be $\leq_m^P$-complete.

But what does it mean for "almost every language in ESPACE" to have some property? Naively, we would like to say that almost every language is ESPACE is in some set X if, in the above random experiment, $\Pr(X|\mathrm{ESPACE}) = \Pr_A[A \in X | A \in \mathrm{ESPACE}] = 1$. The problem here is that ESPACE is a countable set of languages, so $\Pr_A[A \in \mathrm{ESPACE}] = 0$, so the conditional probability $\Pr(X|\mathrm{ESPACE})$ is not defined. We thus turn to *resource-bounded measure*, a complexity-theoretic generalization of Lebesque measure developed by Lutz[Lut92a, Lut92b]. Suppose we are given a *resource bound*, e.g., the set pspace, consisting of all functions computable in polynomial space. Then resource-bounded measure theory defines the pspace-*measure* $\mu_{\mathrm{pspace}}(X)$ of a set X of languages (provided that X is pspace-measurable). In all cases, $0 \leq \mu_{\mathrm{pspace}}(X) \leq 1$. If $\mu_{\mathrm{pspace}}(X) = 0$ or $\mu_{\mathrm{pspace}}(X) = 1$, then $\Pr(X) = 0$ or $\Pr(X) = 1$, respectively, but the pspace-measure conditions are much stronger than this: It is shown in [Lut92a, Lut92b] that, if $\mu_{\mathrm{pspace}}(X) = 0$, then $X \cap \mathrm{ESPACE}$ is a *negligibly small* subset of ESPACE. In fact, pspace-measure induces a natural, internal, measure structure on ESPACE. In this structure, a set X of languages has *measure 0 in* ESPACE, and we write $\mu(X|\mathrm{ESPACE}) = 0$, if $\mu_{\mathrm{pspace}}(X \cap \mathrm{ESPACE}) = 0$. A set X has *measure 1 in* ESPACE, and we write $\mu(X|\mathrm{ESPACE}) = 1$, if $\mu(X^c|\mathrm{ESPACE}) = 0$. Finally, we say that *almost every* language in ESPACE is in some set X of languages if $\mu(X|\mathrm{ESPACE}) = 1$. In §3 below we summarize those aspects of resource-bounded measure that are used in this paper.

Kolmogorov complexity, discussed in several papers in this volume, was introduced by Solomonoff[Sol64], Kolmogorov[Kol65], and Chaitin[Cha66]. Resource-bounded Kolmogorov complexity has been investigated extensively [Kol65, Har83, Sip83, Lev84, Lon86, BB86, Huy86, Ko86, AR88, All89, AW90, Lut90, Lut92a, etc.]. In this paper we work with the *space-bounded Kolmogorov complexity* of languages. Roughly speaking, for $A \subseteq \{0,1\}^*$, $n \in \mathbf{N}$, and a space bound t, the space-bounded Kolmogorov complexity $KS^t(A_{=n})$ is the length of the shortest program that prints the 2^n-bit characteristic string of $A_{=n} = A \cap \{0,1\}^n$, using at most t units of workspace. This quantity $KS^t(A_{=n})$ is frequently interpreted as the "amount of information" that is contained in $A_{=n}$ and is "accessible" by computation using $\leq t$ space. In §4 below, we review the precise formulation of this definition (and the analoguous definition of $KS^t(A_{\leq n})$) and some of its properties. After surveying some recent complexity-theoretic applications of an almost-everywhere lower bound on $KS^t(A_{\leq n})$[Lut92a], we prove a new almost everywhere lower bound result (Theorem 6/Corollary 7) showing that for all $c \in \mathbf{N}$ and $\epsilon > 0$, almost every language $A \in \mathrm{ESPACE}$ has space-bounded Kolmogorov complexity

$$KS^{2^{cn}}(A_{=n}) > 2^n - n^\epsilon \text{ a.e.}$$

(This improves the $2^n - 2^{\epsilon n}$ lower bound of [Lut92a].) It should be noted that the proof of this result is the only direct use of resource-bounded measure in this paper. All the measure-theoretic results in §5-6 are proven by appeal to this almost everywhere lower bound on space-bounded Kolmogorov complexity.

In §5 , we review the fundamental notion of a *complexity core*, introduced by Lynch[Lyn75] and investigated by many others [Du85, ESY85, Orp86, OS86, BD87, Huy87, RO87, BDR88, DB89, Ye90, etc.]. Intuitively, a complexity core for a language A is a fixed set K of inputs such that *every* machine whose decisions are consistent with A fails to decide efficiently on almost all elements of K. The meanings of "efficiently" and "almost all" are parameters of this definition that may be varied according to the context. In §5, in order to better understand ESPACE, we work with DSPACE(2^{cn})-complexity cores (for fixed constants c). In Theorem 9 we prove that any upper bound on the densities of DSPACE(2^{cn})-complexity cores for a language A implies a corresponding upper bound on the space-bounded Kolmogorov complexity of A. The quantitative details imply that almost every language in ESPACE has co-sparse complexity cores.

In §6, we apply these results to our main topic, which is the complexity and distribution of $\leq_m^P$-hard problems for ESPACE. It is well-known that such problems are not feasibly decidable and must obey certain lower bounds on their complexities. As noted above, Huynh[Huy86] has proven that every $\leq_m^P$-hard for ESPACE has weakly exponential (i.e., $> 2^{n^\epsilon}$ for some $\epsilon > 0$) space-bounded Kolmogorov complexity; and Orponen and Schöning[OS86] have (essentially) proven that every $\leq_m^P$-hard language for ESPACE has a dense DSPACE(2^{cn})-complexity core. Intuitively, such results are not surprising, as we do not expect hard problems to be simple. However, in §6, we prove that these hard problems *must* be simple in that they obey *upper* bounds on their complexities. In Theorem 13 we prove that every DSPACE(2^n)-complexity core of every $\leq_m^P$-hard language for ESPACE must have a dense complement. Note that this upper bound is the "mirror image" of the Orponen-Schöning lower bound cited above: Every hard problem has a dense core, but this core's complement must also be dense. In Theorem 14 we use Theorems 9 and 13 to prove that every $\leq_m^P$-hard language for ESPACE has space-bounded Kolmogorov complexity that is less than 2^n by a weakly exponential amount. Again, note that this upper bound is the "mirror image" of the Huynh lower bound cited above.

We have seen that almost every language in ESPACE has co-sparse complexity cores and essentially maximal Kolmogorov complexity. Thus our upper bounds imply that the $\leq_m^P$-complete problems have *unusually low* space-bounded Kolmogorov complexity and *unusually small* complexity cores for problems in ESPACE. It follows that the $\leq_m^P$-complete problems form a measure 0 subset of ESPACE.

In order to simplify the exposition of the main ideas and to highlight the role played by Kolmogorov complexity, we do not state our results in the strongest possible form in this volume. The interested reader may wish to consult the technical paper [JL92] for a more thorough treatment of these issues. For example, it is shown in [JL92] that $\leq_m^P$-hard problems for E have unusually small complexity cores, whence the $\leq_m^P$-complete problems for E form a measure 0 subset of E. (Note added in proof: Recently, Mayordomo[May91] has independently proven that the $\leq_m^P$-complete problems for E form a measure 0 subset of E. Mayordomo's proof exploits the Berman [Ber76] result that every $\leq_m^P$-complete problem for E has an infinite subset in P.)

2 Preliminaries

Most of our notation and terminology is standard. We deal with *strings*, *languages*, *functions*, and *classes*. Strings are finite sequences of characters over the alphabet $\{0,1\}$; we write $\{0,1\}^*$ for the set of all strings. Languages are sets of strings. Functions usually map $\{0,1\}^*$ into $\{0,1\}^*$. A class is either a set of languages or a set of functions.

When a property $\phi(n)$ of the natural numbers is true for all but finitely many $n \in \mathbf{N}$, we say that $\phi(n)$ holds *almost everywhere (a.e.)* . Similarly, $\phi(n)$ holds *infinitely often (i.o.)*, if $\phi(n)$ is true for infinitely many $n \in \mathbf{N}$. We write $[\![\phi]\!]$ for the Boolean value of a condition ϕ. That is, $[\![\phi]\!] = 1$ if ϕ is true, 0 if ϕ is false.

If $x \in \{0,1\}^*$ is a string, we write $|x|$ for the *length* of x. If $A \subseteq \{0,1\}^*$ is a language, then we write A^c, $A_{\leq n}$, and $A_{=n}$ for $\{0,1\}^* \setminus A$, $A \cap \{0,1\}^{\leq n}$, and $A \cap \{0,1\}^n$ respectively. The sequence of strings over $\{0,1\}$, $s_0 = \lambda, s_1 = 0, s_2 = 1, s_3 = 00, \ldots$, is referred to as the standard lexicographic enumeration of $\{0,1\}^*$. The *characteristic string* of $A_{\leq n}$ is the N-bit string

$$\chi_{A_{\leq n}} = [\![s_0 \in A]\!][\![s_1 \in A]\!] \ldots [\![s_{N-1} \in A]\!],$$

where $N = |\{0,1\}^{\leq n}| = 2^{n+1} - 1$.

We use the string pairing function $\langle x, y\rangle = bd(x)01y$, where $bd(x)$ is x with each bit doubled (e.g., $bd(1101) = 11110011$). Note that $|\langle x,y\rangle| = 2|x| + |y| + 2$ for all $x, y \in \{0,1\}^*$. For each $g : \{0,1\}^* \to \{0,1\}^*$ and $k \in \mathbf{N}$, we also define the function $g_k : \{0,1\}^* \to \{0,1\}^*$ by $g_k(x) = g(\langle 0^k, x\rangle)$ for all $x \in \{0,1\}^*$.

If A is a finite set, we denote its cardinality by $|A|$. A language D is *dense* if there exists some constant $\epsilon > 0$ such that $|D_{\leq n}| > 2^{n^\epsilon}$ a.e. A language S is *sparse* if there exists a polynomial p such that $|S_{\leq n}| \leq p(n)$ a.e.. A language S is *co-sparse* if S^c is sparse.

All *machines* here are deterministic Turing machines. A machine M is an *acceptor* if M on input x either accepts, rejects or does not halt. The language accepted by a machine M is denoted by $L(M)$. A machine M is a *transducer* defining the function f_M if M on input x outputs $f_M(x)$. The functions $time_M(x)$ and $space_M(x)$ represent the number of steps and tape cells, respectively, that the machine M uses on input x. Some of our machines take inputs of the form (x,n), where $x \in \{0,1\}^*$ and $n \in \mathbf{N}$. These machines are assumed to have two input tapes, one for x and the other for the standard binary representation $\beta(n) \in \{0,1\}^*$ of n.

The following standard time- and space-bounded uniform complexity classes are used in this paper.

$$\begin{aligned}
\mathrm{DTIME}(t(n)) &= \{L(M) \mid (\exists c)(\forall x) time_M(x) \leq c \cdot t(|x|) + c\} \\
\mathrm{DTIMEF}(t(n)) &= \{f_M \mid (\exists c)(\forall x) time_M(x) \leq c \cdot t(|x|) + c\} \\
\mathrm{DSPACE}(s(n)) &= \{L(M) \mid (\exists c)(\forall x) space_M(x) \leq c \cdot s(|x|) + c\} \\
\mathrm{DSPACEF}(s(n)) &= \{f_M \mid (\exists c)(\forall x) space_M(x) \leq c \cdot s(|x|) + c\} \\
\mathrm{P} &= \bigcup_{i=1}^{\infty} \mathrm{DTIME}(n^i),
\end{aligned}$$

$$\mathrm{PSPACE} = \bigcup_{i=1}^{\infty} \mathrm{DSPACE}(n^i),$$

$$\mathrm{PF} = \bigcup_{i=1}^{\infty} \mathrm{DTIMEF}(n^i),$$

$$\mathrm{E} = \bigcup_{c=1}^{\infty} \mathrm{DTIME}(2^{cn}), \text{ and}$$

$$\mathrm{ESPACE} = \bigcup_{c=1}^{\infty} \mathrm{DSPACE}(2^{cn}).$$

The nonuniform complexity class P/Poly, mentioned in §1, is defined in terms of machines with advice. An *advice function* is a function $h : \mathbf{N} \to \{0,1\}^*$. A language A is in P/Poly if and only if there exist $B \in \mathrm{P}$, a polynomial p, and an advice function h such that $|h(k)| \leq p(k)$ and $x \in A \iff \langle x, h(|x|)\rangle \in B$ for all $k \in \mathbf{N}$ and $x \in \{0,1\}^*$. It is well-known [KL80] that P/Poly consists exactly of those languages that are computed by polynomial-size Boolean circuits.

If A and B are languages, then a *polynomial time, many-one reduction* (briefly $\leq^P_m$*-reduction*) *of* A *to* B is a function $f \in \mathrm{PF}$ such that $A = f^{-1}(B) = \{x \mid f(x) \in B\}$. A $\leq^P_m$*-reduction of* A is a function $f \in \mathrm{PF}$ that is a $\leq^P_m$-reduction of A to some language B. Note that f is a $\leq^P_m$-reduction of A if and only if f is $\leq^P_m$-reduction of A to $f(A) = \{f(x) \mid x \in A\}$. We say that A is *polynomial time, many-one reducible* (briefly, $\leq^P_m$*-reducible*) to B, and we write $A \leq^P_m B$, if there exists a $\leq^P_m$-reduction f of A to B. In this case, we also say that $A \leq^P_m B$ *via* f.

A language H is $\leq^P_m$*-hard* for a class $\mathcal{C}$ of languages if $A \leq^P_m H$ for all $A \in \mathcal{C}$. A language C is $\leq^P_m$*-complete* for $\mathcal{C}$ if $C \in \mathcal{C}$ and C is $\leq^P_m$-hard for $\mathcal{C}$. If $\mathcal{C} = \mathrm{NP}$, this is the usual notion of NP-completeness[GJ79]. In this paper we are especially concerned with languages that are $\leq^P_m$-hard or $\leq^P_m$-complete for ESPACE.

3 Resource-Bounded Measure

In this section we very briefly give some fundamentals of resource-bounded measure, where the resource bound is polynomial space. (This is the resource bound that endows ESPACE with measure structure.) For more details, examples, motivation, and proofs, see [Lut92a, Lut92b].

The *characteristic sequence* of a language $A \subseteq \{0,1\}^*$ is the binary sequence $\chi_A \in \{0,1\}^\infty$ defined by $\chi_A[i] = [\![s_i \in A]\!]$ for all $i \in \mathbf{N}$. (Recall from §2, that $s_0, s_i, s_2, \ldots$ is the standard enumeration of $\{0,1\}^*$.) For $x \in \{0,1\}^*$ and $A \subseteq \{0,1\}^*$, we say that x is a *prefix*, or *partial specification*, of A if x is a prefix of χ_A, i.e., if there exists $y \in \{0,1\}^\infty$ such that $\chi_A = xy$. In this case, we write $x \sqsubseteq A$. The *cylinder specified by* a string $x \in \{0,1\}^*$ is

$$C_x = \{A \subseteq \{0,1\}^* | x \sqsubseteq A\}.$$

We let $\mathbf{D} = \{m2^{-n} | m, n \in \mathbf{N}\}$ be the set of *nonnegative dyadic rationals*. Many functions in this paper take their values in $\mathbf{D}$ or in $[0, \infty)$, the set of nonnegative real numbers. In fact, with the exception of some functions that map into $[0, \infty)$, all our functions are of the form $f : X \to Y$, where each of the sets X, Y is $\mathbf{N}$, $\{0,1\}^*$, $\mathbf{D}$, or some cartesian product of these sets. Formally, in order to have uniform criteria for their computational complexity, we regard all such functions as mapping $\{0,1\}^*$ into $\{0,1\}^*$. For example, a function $f : \mathbf{N}^2 \times \{0,1\}^* \to \mathbf{N} \times \mathbf{D}$ is formally interpreted as a function $\tilde{f} : \{0,1\}^* \to \{0,1\}^*$. Under this interpretation, $f(i, j, w) = (k, q)$ means that $\tilde{f}(\langle 0^i, \langle 0^j, w\rangle\rangle) = \langle 0^k, \langle u, v\rangle\rangle$, where u and v are the binary representations of the integer and fractional parts of q, respectively. Moreover, we only care about the values of $\tilde{f}$ for arguments of the form $\langle 0^i, \langle 0^j, w\rangle\rangle$, and we insist that these values have the form $\langle 0^k, \langle u, v\rangle\rangle$ for such arguments.

For a function $f : \mathbf{N} \times X \to Y$ and $k \in \mathbf{N}$, we define the function $f_k : X \to Y$ by $f_k(x) = f(k, x) = f(\langle 0^k, x\rangle)$. We then regard f as a "uniform enumeration" of the functions $f_0, f_1, f_2, \ldots$. For a function $f : \mathbf{N}^n \times X \to Y$ $(n \geq 2)$, we write $f_{k,l} = (f_k)_l$, etc.

We work with the resource bound

$$\text{pspace} = \{f : \{0,1\}^* \to \{0,1\}^* \mid f \text{ is computable in polynomial space}\}.$$

(The length $|f(x)|$ of the output *is* included as part of the space used in computing f.)

Resource-bounded measure was originally developed in terms of "modulated covering by cylinders" [Lut90]. Though the main results of this paper are true, the underlying development was technically flawed. This situation is remedied in [Lut92a, Lut92b], where resource-bounded measure is reformulated in terms of density functions. We review relevant aspects of the latter formulation here.

A *density function* is a function $d : \{0,1\}^* \to [0, \infty)$ satisfying

$$d(x) \geq \frac{d(x0) + d(x1)}{2}$$

for all $x \in \{0,1\}^*$. The *global value* of a density function d is $d(\lambda)$. An n-dimensional *density system (n-DS)* is a function $d : \mathbf{N}^n \times \{0,1\}^* \to [0, \infty)$ such that $d_{\mathbf{k}}$ is a density function for every $\mathbf{k} \in \mathbf{N}^n$. It is sometimes convenient to regard a density function as a 0-DS.

A *computation* of an n-DS d is a function $\hat{d} : \mathbf{N}^{n+1} \times \{0,1\}^* \to \mathbf{D}$ such that

$$\left|\hat{d}_{\mathbf{k},r}(x) - d_{\mathbf{k}}(x)\right| \leq 2^{-r} \tag{1}$$

for all $\mathbf{k} \in \mathbf{N}^n$, $r \in \mathbf{N}$, and $x \in \{0,1\}^*$. A pspace-*computation* of an n-DS d is a computation $\hat{d}$ such that $\hat{d} \in$ pspace. An n-DS is pspace-*computable* if there exists a pspace-computation $\hat{d}$ of d.

The *set covered by* a density function d is

$$S[d] = \bigcup_{d(x) \geq 1} C_x.$$

A density function d *covers* a set X of languages if $X \subseteq S[d]$. A *null cover* of a set X of languages is a 1-DS d such that, for all $k \in \mathbf{N}$, d_k covers X with global value $d_k(\lambda) \leq 2^{-k}$. It is easy to show [Lut92b] that a set X of languages has classical Lebesgue measure 0 (*i.e.*, probability 0 in the coin-tossing random experiment) if and only if there exists a null cover of X. In this paper we are interested in the situation where the null cover d is pspace-computable.

Definition 1. Let X be a set of languages and let X^c denote the complement of X.

(1) A pspace-*null cover* of X is a null cover of X that is pspace-computable.
(2) X has pspace-*measure 0*, and we write $\mu_{\text{pspace}}(X) = 0$, if there exists a pspace-null cover of X.
(3) X has pspace-*measure 1*, and we write $\mu_{\text{pspace}}(X) = 1$, if $\mu_{\text{pspace}}(X^c) = 0$.
(4) X has *measure 0 in* ESPACE, and we write $\mu(X \mid \text{ESPACE}) = 0$, if $\mu_{\text{pspace}}(X \cap \text{ESPACE}) = 0$.
(5) X has *measure 1 in* ESPACE, and we write $\mu(X \mid \text{ESPACE}) = 1$, if $\mu(X^c \mid \text{ESPACE}) = 0$. In this case, we say that X contains *almost every* language in ESPACE.

It is shown in [Lut92a, Lut92b] that these definitions endow ESPACE with internal measure-theoretic structure. Specifically, if $\mathcal{I}$ is either the collection $\mathcal{I}_{\text{pspace}}$ of all pspace-measure 0 sets or the collection $\mathcal{I}_{\text{ESPACE}}$ of all sets of measure 0 in ESPACE, then $\mathcal{I}$ is a "pspace-ideal," *i.e.*, is closed under subsets, finite unions, and "pspace-unions" (countable unions that can be generated in polynomial space). More importantly, it is shown that the ideal $\mathcal{I}_{\text{ESPACE}}$ is a *proper* ideal, *i.e.*, that ESPACE does *not* have measure 0 in ESPACE.

Our proof of Theorem 6 below does not proceed directly from the above definitions. Instead we use a sufficient condition, proved in [Lut92a], for a set to have pspace-measure 0. To state this condition we need a polynomial notion of convergence for infinite series. All our series here consist of nonnegative terms. A *modulus* for a series $\sum_{n=0}^{\infty} a_n$ is a function $m : \mathbf{N} \to \mathbf{N}$ such that

$$\sum_{n=m(j)}^{\infty} a_n \leq 2^{-j}$$

for all $j \in \mathbf{N}$. A series is p-*convergent* if it has a modulus that is a polynomial.

The following sufficient condition for a set to have pspace-measure 0 is a special case (for pspace) of a resource-bounded generalization of the classical first Borel-Cantelli lemma.

Lemma 2. *(Lutz[Lut92a]). If d is a pspace-computable 1-DS such that the series $\sum_{n=0}^{\infty} d_n(\lambda)$ is p-convergent, then*

$$\mu_{\text{pspace}}\Big(\bigcap_{t=0}^{\infty} \bigcup_{n=t}^{\infty} S[d_n]\Big) = \mu_{\text{pspace}}(\{A \mid A \in S[d_n] \ i.o.\}) = 0.$$

4 Space-Bounded Kolmogorov Complexity

In this section we present the basic facts about space-bounded Kolmogorov complexity that are used in this paper.

Some terminology and notation will be useful. For a fixed machine M and "program" $\pi \in \{0,1\}^*$ for M, we say that "$M(\pi, n) = w$ in $\leq s$ space" if M, on input (π, n), outputs the string $w \in \{0,1\}^*$ and halts without using more than s cells of workspace. We are especially interested in situations where the output is of the form $\chi_{A_{=n}}$ or of the form $\chi_{A_{\leq n}}$, i.e., the 2^n-bit characteristic string of $A_{=n}$ or the $(2^{n+1}-1)$-bit characteristic string of $A_{\leq n}$, for some language A.

Given a machine M, a space bound $s : \mathbf{N} \to \mathbf{N}$, a language $A \subseteq \{0,1\}^*$, and a natural number n, the $s(n)$-*space-bounded Kolmogorov complexity* of $A_{=n}$ relative to M is

$$KS_M^{s(n)}(A_{=n}) = min\{|\pi| \Big| M(\pi, n) = \chi_{A_{=n}} \text{ in} \leq s(n) \text{ space } \}.$$

Similarly, the $s(n)$-*space-bounded Kolmogorov complexity* of $A_{\leq n}$ relative to M is

$$KS_M^{s(n)}(A_{\leq n}) = min\{|\pi| \Big| M(\pi, n) = \chi_{A_{\leq n}} \text{ in} \leq s(n) \text{ space } \}.$$

Well-known simulation techniques show that there is a machine U that is *optimal* in the sense that for each machine M there is a constant c such that for all s, A and n, we have

$$KS_U^{c \cdot s(n)+c}(A_{=n}) \leq KS_M^{s(n)}(A_{=n}) + c$$

and

$$KS_U^{c \cdot s(n)+c}(A_{\leq n}) \leq KS_M^{s(n)}(A_{\leq n}) + c.$$

As is standard in this subject, we fix an optimal machine U and omit it from the notation.

We now recall the following almost-everywhere lower bound result.

Theorem 3. *(Lutz[Lut92a]). Let $c \in \mathbf{N}$ and $\epsilon > 0$.*

(a) If

$$X = \{A \subseteq \{0,1\}^* | KS^{2^{cn}}(A_{=n}) > 2^n - 2^{\epsilon n} \ a.e.\},$$

then $\mu_{\text{pspace}}(X) = \mu(X|\text{ESPACE}) = 1$.

(b) If

$$Y = \{A \subseteq \{0,1\}^* | KS^{2^{cn}}(A_{\leq n}) > 2^{n+1} - 2^{\epsilon n} \ a.e.\},$$

then $\mu_{\text{pspace}}(Y) = \mu(Y|\text{ESPACE}) = 1$.

Informally, Theorem 3 says that $KS(A_{=n})$ and $KS(A_{\leq n})$ are very high for almost all n, for all almost all $A \in \text{ESPACE}$. This lower bound has been useful in a variety of applications in complexity theory, especially in contexts involving Boolean circuits.

Example 1. The *circuit-size complexity* of a language $A \subseteq \{0,1\}^*$ is the function $CS_A : \mathbf{N} \to \mathbf{N}$ defined as follows: For each $n \in \mathbf{N}$, $CS_A(n)$ is the minimum size (number of gates) required for an n-input, 1-output Boolean (acyclic, combinational) circuit to decide the set $A_{=n}$. (See [Lut92a, BDG88, Weg87] for details of the circuit model, which can be varied in minor ways without affecting this discussion.) Circuit-size complexity has been investigated extensively for over forty years. Shannon[Sha49] proved that *almost every* language $A \subseteq \{0,1\}^*$ has circuit-size complexity

$$CS_A(n) > \frac{2^n}{n} \text{ a.e.} \tag{4.1}$$

That is, if we choose the language $A \subseteq \{0,1\}^*$ probabilistically, according to a random experiment in which an independent toss of a fair coin is used to decide membership of each string $x \in \{0,1\}^*$ in A, then

$$\Pr_A[CS_A(n) > \frac{2^n}{n} \text{ a.e.}] = 1. \tag{4.2}$$

Lupanov[Lup58] proved that *every* language $A \subseteq \{0,1\}^*$ has circuit-size complexity

$$CS_A(n) < \frac{2^n}{n}(1 + O(\frac{1}{\sqrt{n}})). \tag{4.3}$$

Since the lower bound (4.1) and the upper bound (4.3) have asympotic ratio 1, these results say that *almost every* language A has essentially maximum circuit-size complexity almost everywhere. Lupanov named this phenomenon the *Shannon effect.*

Lutz[Lut92a] used Theorem 3 to investigate the Shannon effect in ESPACE. The upper bound (4.3) applies *a fortiori* to languages in ESPACE, but the lower bound (4.2) does not directly say anything about ESPACE because $\Pr_A[A \notin \text{ESPACE}] = 1$ in the same random experiment. However, it is not difficult to see that an upper bound on $CS_A(n)$ *implies* an upper bound on $KS(A_{=n})$. In fact, Lutz[Lut92a] showed that the quantitave details of this relation, combined with Theorem 3(a), imply that, for every real $\alpha < 1$, almost every language $A \in$ ESPACE (and, as a corollary, almost every language $A \subseteq \{0,1\}^*$) has circuit-size complexity

$$CS_A(n) > \frac{2^n}{n}(1 + \frac{\alpha \log n}{n}) \text{ a.e.}$$

Thus the Shannon effect holds with full force in ESPACE.

Example 2. Nisan and Wigderson[NW88] proved that, if E contains a language A that is, in a certain technical sense, "very hard to approximate with circuits," then this language A can be used to construct a pseudorandom generator that is fast enough and secure enough to establish the condition P = BPP. Subsequent to this, Lutz[Lut91] proved that there is a constant $c \in \mathbf{N}$ such that every language A that is *not* "very hard to approximate with circuits" has space-bounded Kolmogorov complexity

$$KS^{2^{cn}}(A_{=n}) < 2^n - 2^{\frac{n}{4}} \text{ i.o.}$$

By Theorem 3(a), this implies that almost every language $A \in$ ESPACE is "very hard to approximate with circuits." This fact, together with the result of Nisan and Wigderson, immediately yields an *upward measure separation* theorem, stating that

$$\mathrm{P} \neq \mathrm{BPP} \Rightarrow \mu(\mathrm{E}|\mathrm{ESPACE}) = 0.$$

(Hartmanis and Yesha[HY84] had previously shown that $\mathrm{P} \neq \mathrm{BPP} \Rightarrow \mathrm{E} \subsetneq$ ESPACE.)

In each of the above examples, space-bounded Kolmogorov complexity is used to prove that some set Z of languages has measure 1 in ESPACE. In each case, the method is simply to prove that every language *not* in Z has *unusually low* space-bounded Kolmogorov complexity for languages in ESPACE. That is, every language not in Z has space-bounded Kolmogorov complexity that infinitely often violates the lower bounds obeyed by almost every element of ESPACE.

In this paper we will use similar arguments to show that almost every language $A \in$ ESPACE fails to be $\leq_m^P$-complete for ESPACE. In fact, we will prove that every language H that is $\leq_m^P$-hard for ESPACE has *unusually low* space-bounded Kolmogorov complexity, by which we mean space-bounded Kolmogorov complexity that violates a lower bound obeyed by almost every language $A \in$ ESPACE (and almost every language $A \subseteq \{0,1\}^*$).

As it turns out, Theorem 3 is not strong enough for this purpose! We will show that every $\leq_m^P$-hard language H for ESPACE has an unusually low upper bound on its space bounded Kolmogorov complexity, but this upper bound will *not* violate the lower bounds of Theorem 3. We are thus led to ask how tight the lower bounds of Theorem 3 are.

We first consider Theorem 3(b). Martin-Löf [Mar71] has shown that, for every real $a > 1$, almost every language $A \subseteq \{0,1\}^*$ has space-bounded Kolmogorov complexity

$$KS^{2^{cn}}(A_{\leq n}) > 2^{n+1} - an \text{ a.e.} \tag{4.4}$$

(In fact, Martin-Löf showed that this holds even in the absence of a space bound.) The following known bounds show that the lower bound (4.4) is tight.

Theorem 4. *There exist constants $c_1, c_2 \in \mathbf{N}$ such that every language A satisfies the following two conditions.*

(i) $KS^{2^n}(A_{\leq n}) < 2^{n+1} + c_1$ *for all n.*
(ii) $KS^{2^{c_2 n}}(A_{\leq n}) < 2^{n+1} - n$ *i.o.*

(Part (i) of Theorem 4 is well known and obvious. Part (ii), proven in [Lut92a], extends a result of Martin-Löf [Mar71].)

Since the bound of Theorem 3(b) is considerably lower than that of (4.4), one might expect to improve Theorem 3(b). However, the following upper bound shows that Theorem 3(b) is also tight. (In comparing Theorems 3(b) and 5 it is critical to note the order in which A and ϵ are quantified.)

Theorem 5. *For every language $A \in$ ESPACE, there exists a real $\epsilon > 0$ such that*

$$KS^{2^{2n}}(A_{\leq n}) < 2^{n+1} - 2^{\epsilon n} \ a.e.$$

Proof. Fix $A \in$ ESPACE and $a \in \mathbf{N}$ such that $A \in$ DSPACE(2^{an}). For each $n \in \mathbf{N}$, let $n' = \lfloor \frac{n}{a+1} \rfloor$ and let y_n be the string of length $2^{n+1} - 2^{n'+1}$ such that $\chi_{A_{\leq n}} = \chi_{A_{\leq n'}} y_n$. Let M be a machine that, on input (y, n), computes $\chi_{A_{\leq n'}}$ using $\leq 2^{an'}$ space and then outputs $\chi_{A_{\leq n'}} y$. Let c be the optimality constant for the machine M (given by the definition of the optimal machine U at the beginning of this section). Then $M(y_n, n)$ outputs $\chi_{A_{\leq n}}$ in $\leq 2^{an'}$ space, so for all sufficiently large n, we have

$$\begin{aligned} KS^{2^{2n}}(A_{\leq n}) &\leq KS_M^{2^{an'}}(A_{\leq n}) + c \\ &\leq |y_n| + c \\ &= 2^{n+1} - 2^{n'+1} + c \\ &< 2^{n+1} - 2^{\epsilon n}, \end{aligned}$$

where $\epsilon = \frac{1}{a+2}$.

Thus we cannot hope to improve Theorem 3(b).

An elementary counting argument shows that, for every $c \in \mathbf{N}$, there *exists* a language $A \in$ ESPACE with $KS^{2^{cn}}(A_{=n}) \geq 2^n$ for all $n \in \mathbf{N}$. This suggests that the prospect for improving Theorem 3(a) may be more hopeful. In fact, we have the following almost-everywhere lower bound result.

Theorem 6. *Let $c \in \mathbf{N}$ and let $f : \mathbf{N} \to \mathbf{N}$ be such that $f \in$ pspace and $\sum_{n=0}^{\infty} 2^{-f(n)}$ is p-convergent. If*

$$X = \{A \subseteq \{0,1\}^* | KS^{2^{cn}}(A_{=n}) > 2^n - f(n) \ a.e.\},$$

then $\mu_{\text{pspace}}(X) = \mu(X|\text{ESPACE}) = 1$.

Proof. Assume the hypothesis. By Lemma 2, it suffices to exhibit a pspace-computable 1-DS d such that

$$\sum_{n=0}^{\infty} d_n(\lambda) \text{ is p-convergent} \tag{4.5}$$

and

$$X^c \subseteq \bigcap_{t=0}^{\infty} \bigcup_{n=t}^{\infty} S[d_n]. \tag{4.6}$$

Some notation will be helpful. For $n \in \mathbf{N}$, let

$$B_n = \{\pi \in \{0,1\}^{\leq 2^n - f(n)} | U(\pi, n) \in \{0,1\}^{2^n} \text{ in } \leq 2^{cn} \text{ space }\}. \tag{4.7}$$

For $n \in \mathbf{N}$ and $\pi \in B_n$, let

$$Z_{n,\pi} = \bigcup_{|z|=2^n-1} C_{zU(\pi,n)}.$$

(Thus $Z_{n,\pi}$ is the set of all languages A such that $U(\pi, n)$ is the 2^n-bit characteristic string of $A_{=n}$.) For $n \in \mathbf{N}$ and $w \in \{0,1\}^*$, let

$$\sigma(n, w) = \sum_{\pi \in B_n} \Pr(Z_{n,\pi} | C_w), \tag{4.8}$$

where the conditional probabilities $\Pr(Z_{n,\pi}|C_w) = \Pr_A[A \in Z_{n,\pi} | A \in C_w]$ are computed according to the random experiment in which a language $A \subseteq \{0,1\}^*$ is chosen probabilistically, using an independent toss of a fair coin to decide membership of each string in A. Finally, define the function $d : \mathbf{N} \times \{0,1\}^* \to [0, \infty)$ as follows. (In all three clauses, $n \in \mathbf{N}$, $w \in \{0,1\}^*$, and $b \in \{0,1\}$.)

(i) If $0 \leq |w| < 2^n - 1$, then $d_n(w) = 2^{1-f(n)}$.
(ii) If $2^n - 1 \leq |w| < 2^{n+1} - 1$, then $d_n(wb) = d_n(w)\frac{\sigma(n,wb)}{\sigma(n,w)}$.
(iii) If $|w| \geq 2^{n+1} - 1$, then $d_n(wb) = d_n(w)$.

(The condition $\sigma(n, w) = 0$ can only occur if $d_n(w) = 0$, in which case we understand clause (ii) to mean that $d_n(wb) = 0$.)

It is clear from (4.8) that

$$\sigma(n, w) = \frac{\sigma(n, w0) + \sigma(n, w1)}{2}$$

for all $n \in \mathbf{N}$ and $w \in \{0,1\}^*$. It follows by a routine induction on the definition of d that d is a 1-DS. It is also routine to check that d is pspace-computable. (The crucial point here is that we are only required to perform computations of the type (4.8) when $|w| \geq 2^n - 1$, so the 2^{cn} space bound of (4.7) is polynomial in $|w|$.) Since $\sum_{n=0}^{\infty} 2^{-f(n)}$ is p-convergent, it is immediate from clause (i) that (4.5) holds. All that remains, then, is to verify (4.6).

For each language $A \subseteq \{0,1\}^*$, let

$$I_A = \{n \in \mathbf{N} \mid KS^{2^{cn}}(A_{=n}) \leq 2^n - f(n)\}.$$

Fix a language A for a moment and let $n \in I_A$. Then there exists $\pi_0 \in B_n$ such that $A \in Z_{n,\pi_0}$. Fix such a program π_0 and let $x, y \in \{0,1\}^*$ be the characterstic strings of $A_{<n}$, $A_{\leq n}$, respectively. (Thus $|x| = 2^n - 1$, $|y| = 2^{n+1} - 1$, and $y = xU(\pi_0, n)$.) The definition of d tells us that $d_n(y)$ is $d_n(x)$ times a telescoping product, i.e.,

$$\begin{aligned} d_n(y) &= d_n(x) \prod_{i=0}^{2^n-1} \tfrac{\sigma(n,y[0..2^n+i])}{\sigma(n,y[0..2^n-1+i])} \\ &= d_n(x)\tfrac{\sigma(n,y)}{\sigma(n,x)} \\ &= 2^{1-f(n)}\tfrac{\sigma(n,y)}{\sigma(n,x)}. \end{aligned} \tag{4.9}$$

Since $C_y \subseteq Z_{n,\pi_0}$, we have

$$\sigma(n,y) = \sum_{\pi\in B_n} \Pr(Z_{n,\pi}|C_y) \geq \Pr(Z_{n,\pi_0}|C_y) = 1. \tag{4.10}$$

For each $\pi \in B_n$, the events C_x and $Z_{n,\pi}$ are independent, so

$$\begin{aligned}\sigma(n,x) &= \sum_{\pi\in B_n} \Pr(Z_{n,\pi}|C_x)\\ &= \sum_{\pi\in B_n} \Pr(Z_{n,\pi})\\ &= |B_n|2^{-2^n}\\ &< 2^{1-f(n)}.\end{aligned} \tag{4.11}$$

By (4.9), (4.10), and (4.11), we have $d_n(y) > 1$. It follows that $A \in C_y \subseteq S[d_n]$. Since $n \in I_A$ is arbitrary here, we have shown that $A \in S[d_n]$ for all $A \subseteq \{0,1\}^*$ and $n \in I_A$. It follows that, for all $A \subseteq \{0,1\}^*$,

$$\begin{aligned}A \in X^c &\Rightarrow |I_A| = \infty\\ &\Rightarrow A \in S[d_n] \text{ i.o.}\\ &\Rightarrow A \in \bigcap_{t=0}^{\infty}\bigcup_{n=t}^{\infty} S[d_n],\end{aligned}$$

i.e., (4.6) holds. This completes the proof.

Corollary 7. *Let* $c \in \mathbf{N}$ *and* $\epsilon > 0$. *If*

$$X = \{A \subseteq \{0,1\}^* | KS^{2^{cn}}(A_{=n}) > 2^n - n^\epsilon \ \textit{a.e.}\},$$

then $\mu_{\mathrm{pspace}}(X) = \mu(X|\mathrm{ESPACE}) = 1$.

Proof. Routine calculus shows that the series $\sum_{n=0}^{\infty} 2^{-n^\epsilon}$ is p-convergent.

Corollary 7 is clearly a substantial improvement of Theorem 3(a). We will exploit this improvement in the following two sections.

5 Complexity Cores

A complexity core for a language A is a fixed set $K \subseteq \{0,1\}^*$ such that every machine consistent with A fails to decide efficiently on almost all inputs from K. In this section we review this notion carefully and prove that upper bounds on the size of complexity cores for a language A imply corresponding upper bounds on the space-bounded Kolmogorov complexity of A.

Given a machine M and an input $x \in \{0,1\}^*$, we write $M(x) = 1$ if M accepts x, $M(x) = 0$ if M rejects x, and $M(x) = \perp$ in any other case (i.e., if M fails to halt or M halts without deciding x). If $M(x) \in \{0,1\}$, we write $space_M(x)$ for the number of tape cells used in the computation of $M(x)$. If $M(x) = \perp$, we define $space_M(x) = \infty$. We partially order the set $\{0,1,\perp\}$ by $\perp < 0$ and $\perp < 1$, with 0 and 1 incomparable. A machine M is *consistent* with a language $A \subseteq \{0,1\}^*$ if $M(x) \leq [\![x \in A]\!]$ for all $x \in \{0,1\}^*$.

Definition 8. Let $s : \mathbf{N} \to \mathbf{N}$ be a space bound and let $A, K \subseteq \{0,1\}^*$. Then K is a DSPACE($s(n)$)-*complexity core* of A if, for every $c \in \mathbf{N}$ and every machine M that is consistent with A, the "fast set"

$$F = \{x \,|\, space_M(x) \le c \cdot s(|x|) + c\}$$

satisfies $|F \cap K| < \infty$. (By our definition of $space_M(x)$, $M(x) \in \{0,1\}$ for all $x \in F$. Thus F is the set of all strings that M "decides efficiently".)

Note that every subset of a DSPACE($s(n)$)-complexity core of A is a DSPACE($s(n)$)-complexity core of A. Note also that, if $t(n) = O(s(n))$, then every DSPACE($s(n)$)-complexity core of A is a DSPACE($t(n)$)-complexity core of A.

Remark. Definition 8 quantifies over all machines consistent with A, while the standard definition of complexity cores (cf. [BDG90]) quantifies only over machines that *decide* A. This difference renders Definition 8 stronger than the standard definition when A is not recursive. For example, consider *tally* languages (i.e., languages $A \subseteq \{0\}^*$). Under Definition 8, every DSPACE(n)-complexity core K of every tally language must satisfy $|K \setminus \{0\}^*| < \infty$. However, under the standard definition, *every* set $K \subseteq \{0,1\}^*$ is *vacuously* a complexity core for *every* nonrecursive language (tally or otherwise). Thus by quantifying over all machines consistent with A, Definition 8 makes the notion of complexity core meaningful for nonrecursive languages A. This enables one to eliminate the extraneous hypothesis that A is recursive from several results. In some cases (e.g., the fact that A is P-bi-immune if and only if $\{0,1\}^*$ is a P-complexity core for A [BS85]), this improvement is of little interest. However in §6 below, we show that *every* $\le_m^P$-hard language H for ESPACE has unusually small complexity cores, hence unusually low space-bounded Kolmogorov complexity. This upper bound holds regardless of whether H is recursive.

It should also be noted that standard existence theorems on complexity cores (e.g., every language $A \notin$ P has an infinite P-complexity core [Lyn75]; every $\le_m^P$-hard language for E has a dense P-complexity core [OS86]) remain true under Definition 8. Thus no harm is done by quantifying over all machines consistent with A.

Intuitively, a language is complex if it has very large complexity cores. The converse implication, that a language is simple if it does not have large complexity cores, is supported by the following technical result.

Theorem 9. *Let $A \subseteq \{0,1\}^*$, $\epsilon > 0$, $b > c > 0$, and $g : \mathbf{N} \to [0, \infty)$. If every DSPACE($2^{cn}$)-complexity core K of A has density $|K_{=n}| \le 2^n - g(n)$ i.o., then $KS^{2^{bn}}(A_{=n}) < 2^n - n^{-\epsilon} g(n) + 3\epsilon \log n$ i.o.*

Proof. Let $A \subseteq \{0,1\}^*$, $\epsilon > 0$, and $b > c > 0$. Let $k = \lceil \frac{1}{\epsilon} \rceil$, fix a, d such that $b > a > d > c$, and let $M_0, M_1, M_2, ...$ be a standard enumeration of the deterministic Turing machines. For each $m \in \mathbf{N}$, define the sets

$$\begin{aligned} F_m &= \{x | space_{M_m}(x) \leq 2^{d|x|}\}, \\ B_m &= F_m \setminus \{0,1\}^{\leq m^k}, \\ B &= \bigcup_{cons(m,A)} B_m, \\ K &= \{0,1\}^* \setminus B, \end{aligned}$$

where the predicate $cons(m, A)$ asserts that M_m is consistent with A. Note that, if M_m is a machine that is consistent with A, then $F_m \cap K = F_m \setminus B \subseteq F_m \setminus B_m \subseteq \{0,1\}^{\leq m^k}$, so $|F_m \cap K| < \infty$. Thus K is a DSPACE(2^{cn})-complexity core for A.

Let

$$S = \{n \Big| |K_{=n}| \leq 2^n - g(n)\} = \{n \Big| |B_{=n}| \geq g(n)\}.$$

Then, for each $n \in S$, we have

$$\begin{aligned} g(n) \leq |B_{=n}| &= |(\bigcup_{cons(m,A)} B_m)_{=n}| \\ &\leq \sum_{cons(m,A)} |(B_m)_{=n}| \\ &= \sum_{(m^k<n)\wedge(cons(m,A))} |(B_m)_{=n}| \\ &\leq \sum_{(0\leq m<n^\epsilon)\wedge(cons(m,A))} |(B_m)_{=n}| \\ &\leq \sum_{(0\leq m<n^\epsilon)\wedge(cons(m,A))} |(F_m)_{=n}| \end{aligned}$$

and there are $\leq n^\epsilon$ terms in this last sum, so there exists $0 \leq m < n^\epsilon$ such that M_m is consistent with A and $|(F_m)_{=n}| \geq n^{-\epsilon} g(n)$.

Now let M be a machine that implements the algorithm of Figure 1 with input $(\langle \beta(m), y\rangle, n)$, where $y \in \{0,1\}^*$ and $\beta(m)$ is the binary representation of a natural number m. (Let $N = 2^n$ and let $w_0, ..., w_{N-1}$ be the lexicographic enumeration of $\{0,1\}^n$. We use the symbol $\perp$ for a bit of z that has not yet been defined. For a string $y \neq \lambda$, $head(y)$ is the first bit of y and $tail(y)$ is the rest of y.) Since $a > d$, it is clear that M can be designed so that $M(\langle\beta(m), y\rangle, n)$ uses $\leq 2^{an}$ workspace. For each $n \in S$, choose $m \in \mathbf{N}$ and $y \in \{0,1\}^*$ such that $0 \leq m < n^\epsilon$, M_m is consistent with A, $|(F_m)_{=n}| \geq n^{-\epsilon} g(n)$, and y consists of the $2^n - |(F_m)_{=n}|$ successive bits $[\![w_i \in A]\!]$ for $w_i \in \{0,1\}^n \setminus F_m$. Then $M(\langle\beta(m), y\rangle, n)$ is the 2^n-bit characteristic string of $A_{=n}$, so

$$\begin{aligned} KS_M^{2^{an}}(A_{=n}) &\leq |\langle\beta(m), y\rangle| \\ &= |y| + 2|\beta(m)| + 2 \\ &\leq 2^n - |(F_m)_{=n}| + 2\log m + 3 \\ &\leq 2^n - n^{-\epsilon} g(n) + 2\epsilon \log n + 3. \end{aligned}$$

```
begin
      z := ⊥^N;
      for i := 0 to N − 1 do
      begin
            Simulate M_m(w_i) as long as this uses ≤ 2^{dn} space.
            if this simulation accepts or rejects
               then set z[i] := 1 or z[i] := 0, respectively
               else (z[i], y) := (head(y), tail(y))
      end;
      output z;
end M.
```

Fig. 1. Algorithm for proof of Theorem 9.

It follows that there is a constant $c_M \in \mathbf{N}$ such that, for all $n \in S$,

$$KS^{2^{bn}}(A_{=n}) \leq 2^n - n^{-\epsilon}g(n) + 2\epsilon \log n + 3 + c_M.$$

Hence,

$$KS^{2^{bn}}(A_{=n}) \leq 2^n - n^{-\epsilon}g(n) + 3\epsilon \log n. \tag{5.1}$$

for all but finitely many $n \in S$.

If the hypothesis of Theorem 9 holds, then S is infinite, so (5.1) holds i.o.

Since almost every language in ESPACE has high space-bounded Kolmogorov complexity almost everywhere, Theorem 9 allows us to conclude that almost every language in ESPACE has very large complexity cores.

Theorem 10. *Fix real constants $c > 0$ and $\epsilon > 0$. Let Y be the set of all languages A such that A has a DSPACE(2^{cn})-complexity core K with $|K_{=n}| > 2^n - n^\epsilon$ a.e. Then $\mu_{\text{pspace}}(Y) = \mu(Y|\text{ESPACE}) = 1$.*

Proof. Let c, ϵ and Y be as given. Assume that $A \notin Y$. Then every DSPACE(2^{cn})-complexity core K of A has $|K_{=n}| \leq 2^n - n^\epsilon$ i.o. Since $\frac{\epsilon}{2} > 0$, it follows by Theorem 9 that

$$KS^{2^{(c+1)n}}(A_{=n}) < 2^n - n^{\frac{\epsilon}{2}} + 2\epsilon \log n \text{ i.o.}$$

Since $n^{\frac{\epsilon}{2}} > n^{\frac{\epsilon}{4}} + 2\epsilon \log n$ a.e., it follows that

$$KS^{2^{(c+1)n}}(A_{=n}) < 2^n - n^{\frac{\epsilon}{4}} \text{ i.o.}$$

Taking the contrapositive, this argument shows that $X \subseteq Y$, where

$$X = \{A \subseteq \{0,1\}^* | KS^{2^{(c+1)n}}(A_{=n}) > 2^n - n^{\frac{\epsilon}{4}} \text{ a.e.}\}.$$

It follows by Corollary 7 that $\mu_{\text{pspace}}(Y) = \mu(Y|\text{ESPACE}) = 1$.

Corollary 11. *For every $c > 0$, almost every language in ESPACE has a co-sparse DSPACE(2^{cn})-complexity core.*

6 The Distribution of Hardness

In this section we use the results of §§4-5 to investigate the complexity and distribution of the $\leq_m^P$-hard languages for ESPACE. From a technical standpoint, the main result of this section is Theorem 12, which says that every $\leq_m^P$-hard language for ESPACE is DSPACE(2^n)-decidable on a dense, DSPACE(2^n)-decidable set of inputs.

Two simple notations will be useful in the proof of Theorem 12. First, the *nonreduced image* of a language $S \subseteq \{0,1\}^*$ under a function $f : \{0,1\}^* \to \{0,1\}^*$ is

$$f^{\geq}(S) = \{f(x) \big| x \in S \text{ and } |f(x)| \geq |x|\}.$$

Note that

$$f^{\geq}(f^{-1}(S)) = S \cap f^{\geq}(\{0,1\}^*)$$

for all f and S.

The *collision set* of a function $f : \{0,1\}^* \to \{0,1\}^*$ is

$$C_f = \{x \mid (\exists y < x) f(x) = f(y)\}.$$

(Here, we are using the standard ordering $s_0 < s_1 < s_2 < \ldots$ of $\{0,1\}^*$.) Note that f is one-to-one if and only if $C_f = \emptyset$. Also,

$$|S| \leq |f(S)| + |C_f|$$

holds for every set $S \subseteq \{0,1\}^*$.

A language $A \subseteq \{0,1\}^*$ is *incompressible* by $\leq_m^P$-reductions if $|C_f| < \infty$ for every $\leq_m^P$-reduction f of A.

Theorem 12. *For every $\leq_m^P$-hard language H for ESPACE, there exist $B, D \in$ DSPACE(2^n) such that D is dense and $B = H \cap D$.*

Proof. By a construction of Meyer[Mey77], there is a language $A \in$ DSPACE(2^n) that is incompressible by $\leq_m^P$-reductions. For the sake of completeness, we review the construction of A at the end of this proof. First, however, we use A to prove Theorem 12.

Let H be $\leq_m^P$-hard for ESPACE. Then there is a $\leq_m^P$-reduction f of A to H. Let $B = f^{\geq}(A)$, $D = f^{\geq}(\{0,1\}^*)$. Since $A \in$ DSPACE(2^n) and $f \in PF$, it is clear that $B, D \in$ DSPACE(2^n).

Fix a polynomial q and a real number $\epsilon > 0$ such that $|f(x)| \leq q(|x|)$ for all $x \in \{0,1\}^*$ and $q(n^{2\epsilon}) < n$ a.e. Let $W = \{x \Big| |f(x)| < |x|\}$. Then, for all sufficiently large $n \in \mathbf{N}$, writing $m = \lfloor n^{2\epsilon} \rfloor$, we have

$$\begin{aligned} f(\{0,1\}^{\leq m}) \setminus \{0,1\}^{<m} &\subseteq f(\{0,1\}^{\leq m}) \setminus f(W_{\leq m}) \\ &\subseteq f^{\geq}(\{0,1\}^{\leq m}) \\ &\subseteq D_{\leq q(m)} \\ &\subseteq D_{\leq n}, \end{aligned}$$

whence

$$\begin{aligned}|D_{\leq n}| &\geq |f(\{0,1\}^{\leq m})| - |\{0,1\}^{<m}| \\ &\geq |\{0,1\}^{\leq m}| - |C_f| - |\{0,1\}^{<m}| \\ &= 2^m - |C_f|.\end{aligned}$$

Since $|C_f| < \infty$, it follows that $|D_{\leq n}| > 2^{n^\epsilon}$ for all sufficiently large n. Thus D is dense.

Finally, note that $B = f^{\geq}(A) = f^{\geq}(f^{-1}(H)) = H \cap f^{\geq}(\{0,1\}^*) = H \cap D$. This completes the proof of Theorem 12.

We now describe Meyer's construction of the language A. It is well-known that there is a function $g \in \mathrm{DTIMEF}(n^{\log n})$ that is universal for PF in the sense that

$$\mathrm{PF} = \{g_k | k \in \mathbf{N}\}.$$

(Recall that g_k is defined by $g_k(x) = g(\langle 0^k, x\rangle)$ for all $x \in \{0,1\}^*$.) Fix such a function g. Let $A = L(M)$, where M is a machine that implements the algorithm

```
begin
      input x ;
      R := ∅; S := ∅;
      for n := 0 to |x| do
      begin
            R := R ∪ {n};
            if there exists (k, y, z) ∈ R × {0,1}^n × {0,1}^≤n
               such that z < y and g_k(y) = g_k(z) then
            begin
                  find the lexicographically first such (k, y, z);
                  if z ∉ S then S := S ∪ {y};
                  R := R \ {k}
            end
      end;
      if x ∈ S then accept else reject
end M.
```

Fig. 2. Meyer's construction (for proof of Theorem 12).

in Figure 2. It is clear by inspection that $A \in \mathrm{DSPACE}(2^n)$. To see that A is incompressible by $\leq_m^P$-reductions, suppose that $f \in \mathrm{PF}$ and $|C_f| = \infty$. It suffices to show that f is not a $\leq_m^P$-reduction of A. Fix $k \in \mathbf{N}$ such that $f = g_k$. Then there is some $n \in \mathbf{N}$ such that, on input $x = 0^n$, M finds a triple (k, y, z) on cycle n of the for-loop. We then have $f(y) = g_k(y) = g_k(z) = f(z)$ and $y \in A \iff z \notin A$, so $f^{-1}(f(A)) \neq A$, so f is not a $\leq_m^P$-reduction of A.

We now use Theorem 12 to prove our upper bound on the size of complexity cores for hard languages.

Theorem 13. *Every DSPACE(2^n)-complexity core of every $\leq_m^P$-hard language for ESPACE has a dense complement.*

Proof. Let H be $\leq_m^P$-hard for ESPACE, and let K be a DSPACE(2^n)- complexity core of H. Choose B, D for H as in Theorem 12. Fix machines M_B, and M_D that decide B and D respectively, with $space_{M_B}(x) = O(2^{|x|})$ and $space_{M_D}(x) = O(2^{|x|})$. Let M be a machine that implements the following algorithm.

```
begin
      input x;
      if M_D(x) accepts
         then simulate M_B(x)
         else run forever
end M.
```

Then $x \in D \Rightarrow M(x) = [\![x \in B]\!] = [\![x \in H \cap D]\!] = [\![x \in H]\!]$ and $x \notin D \Rightarrow M(x) = \perp \leq [\![x \in H]\!]$, so M is consistent with H. Also, there is a constant $c \in \mathbf{N}$ such that for all $x \in D$,

$$space_M(x) \leq c2^n + c.$$

Since K is a DSPACE(2^n)-complexity core of H, it follows that $K \cap D$ is finite. But D is dense, so this implies that $D \setminus K$ is dense, whence K^c is dense.

Our upper bound on the size of complexity cores now yields an upper bound on the space-bounded Kolmogorov complexity of hard languages.

Theorem 14. *For every $\leq_m^P$-hard language H for ESPACE, there exists $\epsilon > 0$ such that*

$$KS^{2^{2n}}(H_{=n}) < 2^n - 2^{n^\epsilon} \ \ i.o.$$

Proof. Let H be $\leq_m^P$-hard for ESPACE. By Theorem 13, there exists $\epsilon > 0$ such that every DSPACE(2^n)-complexity core K of H has density $|K_{=n}| \leq 2^n - 2^{n^{2\epsilon}}$ i.o. It follows by Theorem 9 that $KS^{2^{2n}}(H_{=n}) < 2^n - n^{-1}2^{n^{2\epsilon}} + 3\log n$ i.o. Since $n^{-1}2^{n^{2\epsilon}} > 2^{n^\epsilon} + 3\log n$ a.e., this implies that $KS^{2^{2n}}(H_{=n}) < 2^n - 2^{n^\epsilon}$ i.o.

Theorems 13 and 14 give upper bounds on the complexity of hard languages. All that remains is to observe that it is unusual for languages in ESPACE to satisfy these bounds:

Theorem 15. *Let $\mathcal{H}$, $\mathcal{C}$ be the sets of languages that are $\leq_m^P$-hard, $\leq_m^P$-complete for ESPACE, respectively. (Thus, $\mathcal{C} = \mathcal{H} \cap$ ESPACE.) Then $\mathcal{H}$ has pspace-measure 0, so $\mathcal{C}$ is a measure 0 subset of ESPACE.*

Proof. By Theorem 14, $\mathcal{H} \cap \{A \subseteq \{0,1\}^* | KS^{2^{2n}}(A_{=n}) > 2^n - \sqrt{n} \text{ a.e.}\} = \emptyset$, so this follows from Corollary 7.

7 Conclusion

Very roughly speaking, our results (together with earlier work of [OS86, Huy86]) admit the following simple summary. We use $KS(A_{=n})$ and $|K_{=n}|$ as measures of the complexity of a language A, where K is a "largest" complexity core for A. These measures *roughly* satisfy the condition $0 \leq KS(A_{=n}) \leq |K_{=n}| \leq 2^n$. In both measures, *almost every* language in ESPACE has complexity $\approx 2^n$ for almost every n. In both measures, *every hard* language for ESPACE has complexity between 2^{n^ϵ} and $2^n - 2^{n^\epsilon}$ for infinitely many n. In fact [JL92], these bounds are tight.

Acknowledgment

We thank Osamu Watanabe and two anonymous reviewers for suggestions that have improved the exposition of this paper.

References

[All89] E. W. Allender. Some consequences of the existence of pseudorandom generators. *Journal of Computer and System Sciences* 39:101–124, 1989.

[AR88] E. W. Allender and R. Rubinstein. P-printable sets. *SIAM Journal on Computing* 17:1193–1202 ,1988.

[AW90] E. W. Allender and O. Watanabe. Kolmogorov complexity and degrees of tally sets. *Information and Computation* 86:160–178, 1990.

[Amb86] K. Ambos-Spies. Randomness, relativizations, and polynomial reducibilities. In *Proceedings of the First Annual Structure in Complexity Theory Conference*, pages 23–34, 1986.

[BB86] J. L. Balcázar and R. Book. Sets with small generalized Kolmogorov complexity. *Acta Informatica* 23:679–688, 1986.

[BDG88] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, Springer-Verlag, 1988.

[BDG90] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*, Springer-Verlag, 1990.

[BS85] J. L. Balcázar and U. Schöning. Bi-immune sets for complexity classes. *Mathematical Systems Theory* 18:1–10, 1985.

[Ber76] L. Berman. On the structure of complete sets: almost everywhere complexity and infinitely often speed-up. In *Proceedings of the 17th. IEEE Symp. of the Foundations of Computer Science*, pages 76–80, 1976.

[BH77] L. Berman and J. Hartmanis. On isomorphism and density of NP and other complete sets. *SIAM Journal on Computing* 6:305–322, 1977.

[BD87] R. Book and D.-Z. Du. The existence and density of generalized complexity cores. *Journal of the Association for Computing Machinery* 34:718–730, 1987.

[BDR88] R. Book, D.-Z Du, and D. Russo. On polynomial and generalized complexity cores. In *Proceedings of the Third Structure in Complexity Theory Conference*, pages 236–250, 1988.

[Cha66] G. J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the Association for Computing Machinery* 13:547–569, 1966.

[Du85] D.-Z. Du. Generalized complexity cores and levelability of intractable sets. Ph.D. dissertation, University of California, Santa Barbara, CA. 1985.

[DB89] D.-Z. Du and R. Book. On inefficient special cases of *NP*-complete problems. *Theoretical Computer Science* 63:239–252, 1989.

[ESY85] S. Even, A. Selman, and Y. Yacobi. Hard core theorems for complexity classes. *Journal of the Association for Computing Machinery* 35:205–217, 1985.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman and Company, 1979.

[Har83] J. Hartmanis. Generalized Kolmogorov complexity and the structure of feasible computations. In *Proceedings of the 24th IEEE Symposium on the Foundations of Computer Science*, pages 439–445, 1983.

[HY84] J. Hartmanis and Y. Yesha. Computation times of NP sets of different densities. *Theoretical Computer Science* 34:17–32, 1984.

[Huy86] D. T. Huynh. Resource-bounded Kolmogorov complexity of hard languages. In *Proceedings of the First Annual Structure in Complexity Theory Conference*, pages 184–195, 1986.

[Huy87] D. T. Huynh. On solving hard problems by polynomial-size circuits. *Information Processing Letters* 24:171–176, 1987.

[JL92] D. W. Juedes and J. H. Lutz. The complexity and distribution of hard problems. in preparation.

[Kan82] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control* 55:40–56, 1982.

[Kar72] R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations.* Ed. R. E. Miller, and J. W. Thatcher, 85–104. New York: Plenum Press, 1972.

[KL80] R. M. Karp and R. J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309, 1980. Also published as Turing machines that take advice. *L'Enseignement Mathematique* 28:191–209, 1982.

[Ko86] K. I. Ko. On the notion of infinite pseudorandom sequences. *Theoretical Computer Science* 48:9–33, 1986.

[KM81] K. I. Ko, and D. Moore. Completeness, approximation, and density. *SIAM Journal on Computing* 10:787–796, 1981.

[Kol65] A. N. Kolmogorov. Three approaches to the quantitative definition of 'information'. *Problems of Information Transmission* 1:1–7, 1965.

[Lev84] L. A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control* 61:15–37, 1984.

[Lon86] L. Longpré. Resource bounded Kolmogorov complexity, a link between computational complexity and information theory. Ph.D. thesis, Cornell University, 1986. Technical Report TR-86-776.

[Lup58] O. B. Lupanov. On the synthesis of contact networks. *Dokl. Akad. Nauk SSSR* 19:23–26, 1958.

[Lut90] J. H. Lutz. Category and measure in complexity classes. *SIAM Journal on Computing* 19:1100–1131, 1990.

[Lut91] J. H. Lutz. An upward measure separation theorem. *Theoretical Computer Science* 81:127–135, 1991.

[Lut92a] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences* 44, 1992, to appear.

[Lut92b] J. H. Lutz. Resource-bounded measure. in preparation.

[Lyn75] N. Lynch. On reducibility to complex or sparse sets. *Journal of the Association for Computing Machinery* 22:341–345, 1975.

[Mar71] P. Martin-Löf. Complexity oscillations in infinite binary sequences. *Zeitschrift für Wahrscheinlichkeitstheory und Verwandte Gebiete* 19:225-230, 1971.

[May91] E. Mayordomo. Almost every set in exponential time is P-bi-immune. In *Proceedings of the Seventeenth International Symposium on Mathematical Foundations of Computer Science*, Springer–Verlag, 1992, to appear.

[Mey77] A. R. Meyer. reported in [BH77].

[MS72] A. R. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the* 13^{th} *IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.

[NW88] N. Nisan and A. Wigderson. Hardness vs. randomness. In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science*, pages 2–11, 1988.

[Orp86] P. Orponen. A classification of complexity core lattices. *Theoretical Computer Science* 70:121–130, 1986.

[OS86] P. Orponen and U. Schöning. The density and complexity of polynomial cores for intractable sets. *Information and Control* 70:54–68, 1986.

[RO87] D. A. Russo and P. Orponen On P-subset structures. *Mathematical Systems Theory* 20:129–136, 1987.

[Sha49] C. E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Technical Journal* 28:59–98, 1949.

[Sip83] M. Sipser. A complexity-theoretic approach to randomness. In *Proceedings of the 15th ACM Symposium of the Theory of Computing*, pages 330–335, 1983.

[Sol64] R. J. Solomonoff. A formal theory of inductive inference. *Information and Control* 7:1–22,224–254, 1964.

[SC89] L. Stockmeyer and A. K. Chandra. Provably difficult combinatorial games. *SIAM Journal on Computing* 8:151–174, 1979.

[Weg87] I. Wegener. *The Complexity of Boolean Functions*. (Wiley–Teubner series in computer science), Stuttgart: Wiley–Teubner, 1987.

[Wil85] C. B. Wilson. Relativized circuit complexity. *Journal of Computer and System Sciences* 31:169–181, 1985.

[Ye90] H. Ye. Complexity cores for P/poly. submitted.

This article was processed using the LaTeX macro package with LMAMULT style