New Bounds for the Language Compression Problem

H. Buhrman* S. Laplante† P. B. Miltersen‡

Abstract

The **CD** *complexity of a string x is the length of the shortest polynomial time program which accepts only the string x. The language compression problem consists of giving an upper bound on the* $\mathbf{CD}^{A^{\leq n}}$ *complexity of all strings x in some set A. The best known upper bound for this problem is $2\log(\|A^{\leq n}\|) + O(\log(n))$, due to Buhrman and Fortnow. We show that the constant factor 2 in this bound is optimal. We also give new bounds for a certain kind of random sets $R \subseteq \{0,1\}^n$, for which we show an upper bound of $\log(\|R^{\leq n}\|) + O(\log(n))$.*

1 Introduction

Kolmogorov complexity is a notion that measures the amount of regularity in a finite string. It has turned out to be a very useful tool in theoretical computer science. A simple counting argument showing that for each length there exist random strings, i.e. strings with no regularity, has had many applications (see [LV97]).

Early in the history of computational complexity resource bounded notions of Kolmogorov complexity were studied [Har83, Lon90, Lon86]. In particular Sipser [Sip83] introduced a new version of resource bounded Kolmogorov complexity, **CD** complexity, where one considers the size of the smallest program that accepts the given string and no others.

Sipser showed that one can approximate the size of sets using **CD** complexity with random advice and then used this to show that **BPP** $\subseteq$ **PH**.

In particular he shows the following theorem:

Theorem 1 *[Sip83] For any $A^{\leq n}$ there exists a string r, $|r| \leq p(n)$ for p some polynomial, such that for all $x \in A^{\leq n}$: $\mathbf{CD}^{p,A^{\leq n}}(x \mid r) \leq \log(\|A^{\leq n}\|) + O(\log(n))$.*

Note that this is almost tight since by simple counting there has to be a string $x \in A^{\leq n}$ such that its time unbounded Kolmogorov complexity, $\mathbf{C}(x) \geq \log(\|A^{\leq n}\|)$. Theorem 1 has one drawback and that is the requirement of the polynomial size random advice string r. Is it possible to eliminate this advice string r?

Buhrman and Fortnow [BF97] prove that this is possible at the cost of a factor of 2:

Theorem 2 *[BF97] For any $A^{\leq n}$ and for all $x \in A^{\leq n}$: $\mathbf{CD}^{p,A^{\leq n}}(x) \leq 2\log(\|A^{\leq n}\|) + O(\log(n))$ for some polynomial p.*

In many applications of resource bounded Kolmogorov complexity it is desirable to have Theorem 2 without the factor of 2. See for example [BF97] and [BT98] for applications of Theorem 2. In both papers Theorem 2 is used to estimate the size of sets in **P** and a lot of additional work is needed to deal with the factor of 2. Therefore Fortnow and Laplante attempt to remove it. They almost succeed in doing this and show that the factor of 2 can be removed for all but a small faction of the strings in $A^{\leq n}$:

Theorem 3 *[FL98] For all but an ϵ fraction of the x in $A^{\leq n}$, $\mathbf{CD}^{p,A^{\leq n}}(x) \leq \log(\|A^{\leq n}\|) + (\log(\frac{n}{\epsilon}))^{O(1)}$ for some polynomial p.*

In this paper we show that in general the factor of 2 can *not* be avoided and that Theorem 2 is optimal. We show that for any n there is a set $B \subseteq \{0,1\}^n$, $\|B\| > 2^{\Omega(n)}$, and a string $x_0 \in B$ such that $\mathbf{CD}^{poly,B}(x_0) \geq 2\log(\|B\|)$. To this end we employ a combinatorial lemma that gives a bound on the size of k-cover free families [DR82].

In contrast to this we also show that for "random" sets $R \subseteq \{0,1\}^n$ the factor of 2 is not necessary and for all $x \in R$: $\mathbf{CD}^{p,R} \leq \log(\|R\|) + O(\log(n))$. By "random" we mean the following. We take a string y, of length dn with high Kolomogorov complexity ($\mathbf{C}(y) \geq |y|$) and chop it up into d strings of length n. These strings will then form R, which will have cardinality d. This leads to a somewhat strange situation since the set B, used to show that the factor

*Supported in part by the EU fifth framework program project QAIP IST-1999-11234. CWI, P.O. Box 94709, Amsterdam, The Netherlands. E-mail: buhrman@cwi.nl.

†Research supported in part by an NSERC postdoctoral fellowship and the EU fifth framework program project QAIP IST-1999-11234. LRI, Université Paris-Sud, Bâtiment 490, 91405 Orsay, France, E-mail: laplante@lri.fr.

‡BRICS and Department of Computer Science, University of Aarhus, 8000 Aarhus C, Denmark, E-mail:bromille@brics.dk.

of 2 is necessary, turns out to be a subset of such a "random" set R.

We then proceed by asking for which versions of **CD** complexity the factor of 2 is necessary. For **NP** $\cap$ **coNP**-complexity we show that the factor of 2 in general is still necessary but for Σ_2^p-complexity it can be removed.

2 Preliminaries

We use basic concepts and notation from computational complexity theory texts like Balcázar, Díaz, and Gabarró [BDG88] and Kolmogorov complexity from the excellent book by Li and Vitányi [LV93]. We use $|x|$ to represent the length of a string x and $\|A\|$ to represent the number of elements in the set A. All of the logarithms are base 2.

Formally, we define the Kolmogorov complexity function $\mathbf{C}(x|y)$ by $\mathbf{C}(x|y) = \min_p\{|p| : U(p,y) = x\}$ where U is some fixed universal deterministic Turing machine. We define unconditional Kolmogorov complexity by $\mathbf{C}(x) = \mathbf{C}(x|\epsilon)$.

A few basic facts about Kolmogorov complexity:

- The choice of U affects the Kolmogorov complexity by at most an additive constant.
- For some constant c, $\mathbf{C}(x) \leq |x| + c$ for every x.
- For every n and every y, there is an x such that $|x| = n$ and $\mathbf{C}(x|y) \geq n$.

We will also use time-bounded Kolmogorov complexity. Fix a fully time-computable function $t(n) \geq n$. We define the $\mathbf{C}^t(x|y)$ complexity function as

$$\mathbf{C}^t(x|y) = \min_p\{|p| : U(p,y) = x \text{ and } U(p) \text{ runs in at most } t(|x|+|y|) \text{ steps}\}.$$

As before we let $\mathbf{C}^t(x) = \mathbf{C}^t(x|\epsilon)$. A different universal U may affect the complexity by at most a additive constant and the time by a $\log t$ factor.

While the usual Kolmogorov complexity asks about the smallest program to *produce* a given string, we may also want to know about the smallest program to *distinguish* a string. While this difference affects the unbounded Kolmogorov complexity by only a constant it can make a difference for the time-bounded case. Sipser [Sip83] defined the distinguishing complexity $\mathbf{CD}^t$ by

$$\mathbf{CD}^t(x|y) = \min_p$$

$$|p| \quad : \quad \begin{array}{ll} (1)\ U(p,x,y) & \text{accepts.} \\ (2)\ U(p,z,y) & \text{rejects for all } z \neq x. \\ (3)\ U(p,z,y) & \text{runs in at most } t(|z|+|y|) \\ & \text{ steps for all } z \in \Sigma^*. \end{array}$$

Fix a universal nondeterministic Turing machine U'. We define the nondeterministic distinguishing complexity $\mathbf{CND}^t$ by

$$\mathbf{CND}^t(x|y) = \min_p$$

$$|p| \quad : \quad \begin{array}{ll} (1)\ U'(p,x,y) & \text{accepts.} \\ (2)\ U'(p,z,y) & \text{rejects for all } z \neq x. \\ (3)\ U'(p,z,y) & \text{runs in at most } t(|z|+|y|) \\ & \text{ steps for all } z \in \Sigma^*. \end{array}$$

Once again we let $\mathbf{CND}^t(x) = \mathbf{CND}^t(x|\epsilon)$.

We can also allow for relativized Kolmogorov complexity. For example for some set A, $\mathbf{CD}^{t,A}(x|y)$ is defined as above except that the universal machine U has access to A as an oracle. Similarly we define **NP** $\cap$ **coNP**-complexity as $\mathbf{CD}^{t,\mathbf{NP}\cap\mathbf{coNP}}(x \mid y)$ and Σ_2^p-complexity as $\mathbf{CD}^{t,\Sigma_2^p}(x \mid y)$, where the universal machine has access to some set in **NP** $\cap$ **coNP**, or Σ_2^P respectively.

3 Random Sets

In Section 4 we will prove that Theorem 2 is optimal and that the factor of 2 can not be removed. In contrast to this we will show in this section that for a certain kind of "random" set the factor of 2 from Theorem 2 *can* be removed. First we will define what we mean by a "random" set.

Definition 1 *Fix n. For any $d \leq 2^{n/5}$, let y be a string of length dn such that $\mathbf{C}(y) \geq dn$. Chop y into d parts of length n, and set $R = \{y_1 \ldots y_n\} \cup \ldots \cup \{y_{(d-1)n+1} \ldots y_{dn}\}$. We will call such a set R random.*

Theorem 4 *For any n let R be a random set with d elements of length n such that $\log(\log(d)) > 3$. For all $x \in R$: $\mathbf{CD}^{p,R}(x) \leq \log(d) + O(\log(n))$, for p some polynomial.*

Proof: Let $x \in R$. The idea is to use the first $\log(d)$ bits $a_x = x_1 \ldots x_{\log(d)}$ of x as a description for x. The first $\log(d)$ bits do not have to describe x uniquely since there may be other strings in R that start with a_x. We will show (see Claim 5) however that there are at most $\log(d) + O(1)$ strings in R that start with a_x. These strings, including x, thus form a set $S_{a_x} \subset R$ with $\|S_{a_x}\| \leq \log(d) + O(1)$. Note that given a_x, we can check in polynomial time whether $z \in S_{a_x}$ since $z \in S_{a_x}$ iff $[z_1 \ldots z_{\log(d)} = a_x$ and $z \in R]$. We can thus use Theorem 2 to describe $x \in S_{a_x}$ using at most $2\log(\log(d)) + O(\log(n)) = O(\log(n))$ additional bits. This means a total description of $\log(d) + O(\log(n))$ bits.

It remains to be shown that there are only few strings in R that start with a_x:

Claim 5 $\|S_{a_x}\| \leq \log(d) + O(1)$

We use the fact that R was constructed from an incompressible string y of length dn. Suppose that there are $k > \log(d)$ strings in R that start with a_x. We will show how to describe y with fewer than dn bits contradicting the fact that $\mathbf{C}(y) \geq dn$. We describe y as follows:

1. a_x with $\log(d)$ bits.
2. $x_1, \ldots, x_k \in S_{a_x}$, using $kn - k\log(d)$ bits.
3. k pointers in y to where $x_1, \ldots, x_k$ are.
4. The rest of y in $(d-k)n$ bits.

Let's estimate the number of bits needed for point 3 in this description. We need to give an index in the ensemble of $\binom{d}{k}$ possible positions. That costs $\log(\binom{d}{k})$ many bits. Since $\binom{d}{k} \leq (\frac{ed}{k})^k$, this costs less than $k\log(d) - k\log(k) + 2k$ bits. Hence the total amount of bits in this description (point 1 + 2 + 3 + 4) is: $dn + \log(d) + 2k - k\log(k)$ which is less than dn if $k > \log(d)$ and $\log(\log(d)) > 3$. $\square$

4 The Lower Bound

In this section we will prove that Theorem 2 is optimal and that the factor of 2 can not be removed. We will use a combinatorial lemma on the size of families of sets that are called k-cover free.

Definition 2 *A family of sets $\mathcal{F}$ is k-cover-free if for any sets $F_0, ..., F_k \in \mathcal{F}$, $F_0 \not\subseteq \bigcup_{i=1}^{k} F_i$.*

Let $m(\mathcal{F})$ denote the size of the universe from which the elements are taken, that is, $m(\mathcal{F}) = \|\bigcup_{F\in\mathcal{F}} F\|$. We will need the following bound on k-cover-free families.

Lemma 1 *[DR82] If $\mathcal{F}$ is a family containing N sets, k-cover-free, and $N > k^3$, then $m(\mathcal{F}) \geq \frac{k^2 \log N}{2\log k + c}$, for some constant c.*

Similar bounds can be found in the literature [Für96, RC96, Rus94]. The paper by Füredi [Für96] has the most accessible proof.

We are now ready to prove our lower bound on the language compression problem.

Theorem 6 *There exist constants c_1 and c_2 such that for any polynomial time bound $t(n)$ and for all sufficiently large input length n and all r the following holds. Let $B \subseteq \{0,1\}^n$ be a set with r elements such that for all $x, y \in B$ it holds that $\mathbf{C}^t(x \mid y) > 2\log(r) + \log(t(n)) + c_1$. Then there exists $A \subseteq B$ of size $\lceil r^{1/3} \rceil$ for which at least some $x \in A$ has $\mathbf{CD}^{p,A}(x) > 2\log(\|A\|) - c_2$.*

Proof: Let n be a large enough input size and let $k = \lceil r^{1/3} \rceil - 1$ determine a set size. Let $P = \{p_1, p_2, ..., p_m\}$ be the set that contains all the polynomial-time programs that run in time t such that $|p_i| \leq 2\log(k+1) - c_2$. The constant c_2 will be determined later.

Assume for a contradiction that for any set $A \subset B$, $\|A\| = k+1$ it is the case that $\mathbf{CD}^{t,A}(x) \leq 2\log(k+1) - c_2$ via a program in P. We want to first make the claim that on input $x \in A$ any $\mathbf{CD}^A$ program from P only queries x and no other string from A. We will show that this is true in the next two lemmata.

Lemma 2 *For any $x \in A$, and any program $p \in P$, if $p^{\{x\}}(x)$ accepts then $p^{\{x\}\cup A}(x) = p^A(x)$ accepts.*

Proof: Assume $p^{\{x\}}(x)$ accepts, but $p^{\{x\}\cup A}(x)$ rejects. Then p must query some string y in A. Therefore $\mathbf{C}^t(y|x) \leq \log(t(n)) + 2\log(k+1) + O(1) \leq 2\log(r) + \log(t(n)) + c_1$ where $t(n)$ bounds the number of queries made by p and c_1 is chosen appropriately. This contradicts the assumption that $\mathbf{C}^t(y \mid x) > 2\log(r) + \log(t(n)) + c_1$. $\square$

Lemma 3 *For any $x \in A \subseteq B$, and any program $p \in P$, if $p^{\{x\}\cup A}(x)$ accepts then $p^{\{x\}}(x)$ accepts.*

Proof: Assume $p^{\{x\}\cup A}(x)$ accepts, but $p^{\{x\}}(x)$ rejects. Then as above, there is a y in A such that $\mathbf{C}^t(y|x) \leq \log(t(n)) + 2\log(k+1) + c_1$, a contradiction. $\square$

To continue the proof of the theorem, consider the following set family. For every x in B,

$$F_x = \{i : p_i^{\{x\}}(x) \text{ accepts}\}.$$

Set $\mathcal{F} = \{F_x \mid x \in B\}$. The family $\mathcal{F}$ contains $\|B\| = r$ sets. We claim that it is k-cover-free. Consider any sets $F_{x_0}, ..., F_{x_k}$ in the family, and let $A = x_0, ..., x_k$. Let $p_i \in P$ be the $\mathbf{CD}^{t,A}$ program for x_0. By Lemma 3, $i \in F_{x_0}$. On the other hand, i cannot be in $\bigcup_{i=1}^{k} F_i$, because by Lemma 2, p_i^A accepts only x_0.

Therefore $m(\mathcal{F}) \geq \frac{k^2 \log(\|B\|)}{2\log(k)+c}$, that is, $\log(m(\mathcal{F})) \geq 2\log(\|A\|) + \log(\log(\|B\|)) - \log(\log(\|A\|)) - O(1)$. For suitable choice of c_2 this contradicts the fact that $\log(m) \leq 2\log(\|A\|) - c_2$. $\square$

Corollary 1 *For every polynomial time bound p and any sufficiently large n there exist sets $A \subseteq B \subseteq \{0,1\}^n$*

1. *$\|B\| < 2^{n/10}$ and $\|A\| > 2^{n/50}$.*
2. *$\exists x \in A^{=n} : \mathbf{CD}^{p,A}(x) \geq 2\log(\|A\|) - O(1)$.*
3. *$\forall x \in B^{=n} : \mathbf{CD}^{p,B}(x) \leq \log(\|B\|) + O(\log(n))$.*

Proof: Take B a random set (see definition 1) with the right values of n and $r = \|B\|$ such that plugging it into Theorem 6 with the correct cardinality of A yields item 2. Item 3 then follows from Theorem 4 and the fact that B is a random set. □

It turns out that the proof of Theorem 6 also works for **NP** $\cap$ **coNP**-complexity.

Corollary 2 *For every polynomial time bound p and any sufficiently large n there exists $A \subseteq \{0,1\}^n$ with $\|A\| > 2^{n/50}$ such that $\exists x \in A$: $\mathbf{CD}^{p,(\mathbf{NP}\cap\mathbf{coNP})^A}(x) \geq 2\log(\|A\|) - O(1)$.*

Proof: An oracle in $(\mathbf{NP} \cap \mathbf{coNP})^A$ is modeled by two linear time **NP**-predicates M_1^A and M_2^A such that for all x exactly one of the two predicates accepts and the other rejects: [$M_1^A(x)$ accepts and $M_2^A(x)$ rejects] or [$M_1^A(x)$ rejects and $M_2^A(x)$ accepts].

In order to let Theorem 6 go through for **NP** $\cap$ **coNP**-complexity we only have to make sure that Lemma 2 and 3 work when the programs $p \in P$ have access to an $(\mathbf{NP} \cap \mathbf{coNP})^A$ oracle. First we slightly change the definition of B and require that for all $x, y \in B, C(x \mid y) \geq 2\log(r) + O(\log(t(n))) + O(1)$.

The statement of Lemma 2 becomes: For any $x \in A \subseteq B$, and any program $p \in P$, if $p^{(M_1,M_2)^{\{x\}}}(x)$ accepts then $p^{(M_1,M_2)^{\{x\}\cup A}}(x)$ accepts. Suppose this is not true, then there has to be a query that was answered differently by $(M_1, M_2)^{\{x\}}$ than by $(M_1, M_2)^A$. Let q be the first such query. Note that q can be described with $\log(t(n))$ bits from x and a description of p. Suppose that $M_1^{\{x\}}(q)$ accepts and $M_1^A(q)$ rejects; the other case is similar but then M_2 plays the role of M_1. Since M_1 changes from accept to reject this has to be because of a string in A. Moreover it has to be a string y in A queried on the leftmost accepting path of $M_1^{\{x\}}(x)$. Hence we can describe y with an additional $\log(t(n))$ bits contradicting the fact that $C(y \mid x) \geq 2\log(r) + O(\log(t(n))) + O(1)$.

Lemma 3 is proven in a similar way. □

The point is that although non-deterministic machines can query every string in oracle A of length n they can not change from an accepting state to a rejecting state when the string we add or remove from A is random. Note that a non-deterministic machine can change from a rejecting state to an accepting state if we add a random string to the oracle. This is the reason why we only are able to prove a lower bound for **NP** $\cap$ **coNP**-complexity and not for **CND**-complexity.

5 Upper bounds with oracles

Our goal is to close the gap between the lower and upper bounds for the language compression problem. In particular, we would like to find the weakest possible oracle with respect to which the $\log(\|A\|)$ bound can be obtained. We obtain that $\log(\|A\|)\log^2(\|A\|n^2)$ queries to a $\Sigma_2^{p,A}$ oracle suffice.

Theorem 7 *There is a polynomial time bound t such that for any set $A \subseteq \{0,1\}^n$, and any $x \in A$, $\mathbf{CD}^{t,\Sigma_2^{p,A}}(x) \leq \log(\|A\|) + O(\log(n))$. Furthermore the **CD** program makes $\log(\|A\|)\log^2(\|A\|n^2)$ queries to the oracle.*

Proof: The proof uses a construction from [BF97].

This construction hinges on the following fact:

Proposition 1 *[BF97] For any set A of size k of strings of length n, there is a set of primes $P = \{p_1, ..., p_{\log(k)}\}$, with $p_i \leq kn^2$ such that for any $x \in A$, there exists a prime $p_i \in P$ such that for any other $y \in A$, $x \not\equiv y \pmod{p_i}$.*

Buhrman and Fortnow use this fact to show that for any $x \in A$, $\mathbf{CD}^{poly,A}(x) \leq 2\log(\|A\|) + O(\log(n))$, by encoding the prime and the modulus that corresponds to x.

Using a $\Sigma_2^{p,A}$-oracle, we can compute the appropriate prime and omit its encoding, and encode only the modulus. This gives us that $\mathbf{CD}^{poly,\Sigma_2^{p,A}}(x) \leq \log(\|A\|) + O(\log(n))$. We now give the details.

For any set $A \subseteq \Sigma^n$, define the language $\mathbf{Good}^A$ to be the set of strings P encoding primes $p_1, ...p_m$, where $m = \log(\|A\|)$ as follows:

$$\mathbf{Good}^A = \{P \mid \forall x, y \in A\ \exists p_i \in P : x \not\equiv y \pmod{p_i}\}$$

Observe that $\mathbf{Good}^A \in \Pi_1^{p,A}$ since the existential quantifier ranges over at most n indices in P.

We define the auxiliary language $\mathbf{GoodPrefix}^A$ to be

$$\mathbf{GoodPrefix}^A = \{x \mid \exists y : xy \in \mathrm{Good}^A\}.$$

It is easy to see that $\mathbf{GoodPrefix}^A \in \Sigma_2^{p,A}$.

To compute the lexicographically least $P = p_1, ...p_m \in \mathbf{Good}^A$, it is enough to make $\log k \log kn^2$ queries to $\mathbf{Good}^A$: the ith query gives us the ith bit of the lexicographically smallest $P = p_1, ...p_m$, in the usual way.

The description for an $x \in A$ then is an index i in the lexicographically smallest P such that for all other $y \in A : x \not\equiv y \pmod{p_i}$ and the value $x \bmod p_i$. The **CD** program for $x \in A$ is as follows: On input z, first, compute the lexicographically least $P = p_1, ...p_m \in \mathbf{Good}^A$. The index i of a prime "good" for x is given, and the modulus $x \bmod p_i$ is also given. Compute $z \bmod p_i$ and compare with $x \bmod p_i$. Accept if and only of they are identical. The length of the program is bounded by $\log(\log(k)) + \log(kn^2) + O(1)$. □

6 Open Problems

We have shown that the factor of 2 in the language compression problem for $\mathbf{CD}^{poly,\mathbf{NP}\cap\mathbf{coNP}}$ is necessary. Furthermore we showed that with the help of a Σ_2^p oracle we can avoid the factor of 2. For which versions of **CD** complexity can we avoid the factor of 2? In particular is this possible for $\mathbf{CND}^{poly}$?

Our lower bound also shows that the theorem of Laplante and Fortnow, achieving optimal compression for all but an ϵ fraction of the strings, is the best we can in general hope for. What is the optimal tradeoff between ϵ and the bound for the language compression problem?

References

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

[BF97] H. Buhrman and L. Fortnow. Resource bounded kolmogorov complexity revisited. In Reischuk and Morvan, editors, *14th Annual Symposium on Theoretcal Computer Science*, volume 1200 of *Lecture Notes in Computer Scienc*, pages 105–116. Springer, 1997.

[BT98] H. Buhrman and L. Torenvliet. Randomness is hard. In *Proceedings of 13th Annual IEEE Conference on Computational Complexity*, 1998.

[DR82] A.G. Dyachkov and V.V. Rykov. Bounds on the length of disjunctive codes. *Problemy Peredachi Informatsii*, 18(3):7–13, 1982. In Russian.

[FL98] L. Fortnow and S. Laplante. Nearly optimal language compression using extractors. In *15th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1373 of *Lecture Notes in Computer Science*, pages 84–93, Paris France, 25–27 February 1998. Springer.

[Für96] Z. Füredi. On r-cover-free families. *Journal of Combinatorial Theory, Series A*, 73:172–173, 1996.

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

[Lon86] L. Longpré. *Resource bounded Kolmogorov complexity, a link between computational complexity and information theory*. PhD thesis, Cornell University, 1986. Technical Report TR86-776.

[Lon90] L. Longpré. Resource bounded Kolmogorov complexity and statistical tests. In *Working Notes, AAAI Spring Symposium Series*, pages 6–10, 1990.

[LV93] M. Li and P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, 1993.

[LV97] M. Li and P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Graduate Texts in Computer Science. Springer-Verlag, second edition, 1997.

[RC96] J. Radhakrishnan and S. Chaudhuri. Deterministic restrictions in circuit complexity. In *Proceedings of the 28th STOC*, 1996.

[Rus94] M. Ruszinkó. On the upper bound of the size of r-cover-free families. *Journal of Combinatorial Theory, Series A*, 66:302–310, 1994.

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