RESOURCE-BOUNDED CATEGORY AND MEASURE IN EXPONENTIAL COMPLEXITY CLASSES

Thesis by

Jack H. Lutz

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology
Pasadena, California

1987

(Submitted May 18, 1987)

Acknowledgment

I am very much indebted to Alexander Kechris and Yaser Abu-Mostafa, my major and minor advisors, respectively. I thank them very gratefully for many helpful discussions and for the constant support, encouragement, and guidance I have received from them.

Many other professors have been very helpful to me at various stages, both at Caltech and at Kansas. I particularly wish to thank Tom Apostol, Alain Martin, Randy Bryant, Steve Muchnick, Fred Van Vleck, Judy Roitman, Jack Porter and Roger Ware. I also thank Leonid Levin, W.A.J. Luxemburg, and Brock Fuller for serving on my committee.

I have greatly enjoyed and benefited from my stay here at Caltech. The mathematics department has been an especially stimulating and congenial environment in which to work. I hope that the many friends I have made here understand how much they have meant to me.

I also want to thank my friends in Yaser's complexity and information theory group for useful discussions, pleasant Monday evenings, and most of all for patiently enduring early versions of this work.

I thank Lillian Chappelle for her skillful and timely typing of this thesis and two earlier papers.

The financial support which Caltech has provided me through various assistantships and fellowships has been quite generous. For this I am grateful to the departments of mathematics, computer science, and electrical engineering. I am particularly grateful for the research support I have received from Caltech's Program in Advanced Technologies, sponsored by

Aerojet General, General Motors, GTE, and TRW, while working on this thesis.

I hope that my older, non-Caltech friends understand how important their continued friendship has been to me during this undertaking.

My debts to family dwarf all others. This happy circumstance applies to my extended family, including my wife's family. I thank everyone.

My mother and my late father deserve special mention. Their love and support have been constant and indispensable. Their active intellects gave me an early start in many things and, later, the great gift of knowing that education could never carry me away from them. I am ever thankful for such fortune.

My daughter Laura and my son Neil have added incalculable joy and love to my last years of school. I record my thanks here for their future reading.

Above all, I thank my wife Robyn for more things than I will ever articulate. With love, this thesis is dedicated to her.

Abstract

This thesis presents *resource-bounded category* and *resource-bounded measure* - two new tools for computational complexity theory - and some applications of these tools to the structure theory of exponential complexity classes.

Resource-bounded category, a complexity-theoretic version of the classical Baire category method, identifies certain subsets of PSPACE, E, ESPACE, and other complexity classes as *meager*. These meager sets are shown to form a nontrivial ideal of "small" subsets of the complexity class. The meager sets are also (almost) characterized in terms of certain two-person infinite games called *resource-bounded Banach-Mazur games*.

Similarly, resource-bounded measure, a complexity-theoretic version of Lebesgue measure theory, identifies the *measure* 0 subsets of E, ESPACE, and other complexity classes, and these too are shown to form nontrivial ideals of "small" subsets. A resource-bounded extension of the classical Kolmogorov zero-one law is also proven. This shows that measurable sets of complexity-theoretic interest either have measure 0 or are the complements of sets of measure 0.

Resource-bounded category and measure are then applied to the investigation of uniform versus nonuniform complexity. In particular, Kannan's theorem that ESPACE $\not\subseteq$ P/Poly is extended by showing that P/Poly $\cap$ ESPACE is only a meager, measure 0 subset of ESPACE. A theorem of Huynh is extended similarly by showing that all but a meager, measure 0 subset of the languages in ESPACE have high space-bounded Kolmogorov complexity.

These tools are also combined with a new hierarchy of exponential time complexity classes to refine known relationships between nonuniform complexity and time complexity.

In the last part of the thesis, known properties of hard languages are extended. In particular, recent results of Schöning and Huynh state that any language L which is $\leq_m^P$-hard for E or $\leq_T^P$-hard for ESPACE cannot be feasibly approximated (i.e., its symmetric difference with any feasible language has exponential density). It is proven here that this conclusion in fact holds unless only a meager subset of E is $\leq_m^P$-reducible to L and only a meager, measure 0 subset of ESPACE is $\leq_m^{PSPACE}$-reducible to L. (It is conjectured, but not proven, that this result is actually stronger than those of Schöning and Huynh.) This suggests a new lower bound method which may be useful in interesting cases.

Contents

1. Introduction

This thesis presents *resource-bounded category* and *resource-bounded measure* - two new tools for computational complexity theory - and some applications of these tools to the structure theory of exponential complexity classes.

Like the reducibilities $\leq_T^P$ and $\leq_m^P$ introduced by Cook [1971], Karp [1972], and Levin [1973], and like the generalized Kolmogorov complexities investigated by Hartmanis [1983], Sipser [1983], and others, these tools are complexity-theoretic generalizations of well-developed mathematical methods. Specifically, resource-bounded category generalizes the classical Baire category method and resource-bounded measure generalizes Lebesgue measure theory.

This thesis falls naturally into two main parts. In sections 3 through 5 we introduce resource-bounded category and measure and their basic properties. In sections 6 through 10 we apply these tools to the structural investigation of exponential complexity classes.

Resource-bounded category and measure impose new structure on (i.e., reveal new structure in) certain complexity classes by identifying certain subsets of these classes as "small."

Sets which are small in the sense of category are called *meager*. The *classical* Baire category method (in Oxtoby [1971], for example) says what it means for a subset of a complete metric space to be meager. A computable or *effective*, version of Baire category was introduced by Mehlhorn [1973] and has also been investigated by Lisagor [1979]. This effective version says what it means for a subset of the set of recursive functions to be meager. The *resource-bounded* version of Baire category developed in section 3 is a

natural extension of these ideas which enables us to discuss meager subsets of complexity classes like PSPACE, E, ESPACE, etc. (See section 2 for notations and terminology used in this introduction.) As it turns out, our formulation is general enough to include classical and effective versions as special cases, so the treatment here is self-contained.

In classical Baire category, meager sets admit a characterization (described in Oxtoby [1971]) in terms of certain two-person infinite games of perfect information, called *Banach-Mazur games*. *Computable* Banach-Mazur games were introduced in Lisagor [1979] and shown to give an analogous characterization in the effective setting. *Resource-bounded* Banach-Mazur games are introduced in section 4 and shown to (almost) characterize sets which are meager in the corresponding sense.

Suppose a language L is chosen probabilistically by using an independent toss of a fair coin to decide whether each string is in L. Then *classical* Lebesgue measure theory (described in Halmos [1950] and Oxtoby [1971], for example) identifies certain *measurable sets* of languages (also called *events*) and assigns to each measurable set X a *measure* $\mu(X)$, which is the probability that the language so chosen will be an element of X. A set X of languages is then small in the sense of measure if it has measure 0. *Effective* versions of measure theory, which say what it means for a set of computable languages to have measure 0 as a subset of the set of all such languages, have been investigated by Friedzon [1972], Mehlhorn [1974], and many others. The *resource-bounded* measure theory introduced in section 5 has the classical and effective theories as special cases, but also defines measurability and measure for subsets of many complexity classes. The small subsets of the complexity class are then the measure 0 sets.

It is tempting and thought-provoking to regard the measure of a subset X of a complexity class $\mathbf{C}$ as the "conditional probability" that $L \in X$, given that $L \in \mathbf{C}$, when L is chosen by the above-mentioned experiment. However, this interpretation should not be taken seriously because $\mathbf{C}$ is itself a countable, hence measure 0, subset of the set of all languages. (See the remarks on the Borel paradox in Kolmogorov [1933], for example.)

The main results of sections 3 through 5 are the definitions of the resource-bounded meager and measure 0 sets, the justification for calling these sets small (especially Theorems 3.12 and 5.9), the game characterization of meager sets (Theorems 4.3 and 4.4), and a resource-bounded generalization of the classical Kolmogorov zero-one law (Theorem 5.15) which indicates that measurable sets of interest in complexity theory have measure 0 or 1. Many other results can be proven but are not included here because they are not needed for the applications in the ensuing sections.

The applications in sections 6 through 10 all concern the structure of exponential time and space complexity classes. Despite the fact that such classes are far beyond the realm of feasible computation, there are three good reasons for studying their structure.

The first reason is the unfortunate circumstance that many known methods do not work below this level. One of the main areas of complexity theory, the effort to clarify relationships between uniform and nonuniform complexity measures, is currently in this predicament.

A central part of the study of uniform versus nonuniform complexity is the ongoing investigation of (nonuniform, Boolean) circuit-size versus (uniform, algorithmic) time and space. In particular, if P/Poly is the set of languages which have polynomial-size circuits, then it is clear that $P \subseteq P/Poly$ and that

P/Poly $\not\subseteq$ REC. The following is also known.

Theorem 1.1 (Kannan [1982]). ESPACE $\not\subseteq$ P/Poly. □

It is generally believed that NP $\not\subseteq$ P/Poly and in fact Karp and Lipton [1980] have shown that NP $\subseteq$ P/Poly has the unlikely consequence of collapsing the polynomial-time hierarchy to the second level. Nevertheless, the weaker conjectures NP $\not\subseteq$ SIZE(n) and E $\not\subseteq$ P/Poly have yet to be proven, and the results of Wilson [1985] show that even these will require nonrelativizable proof techniques.

In section 7 we extend Theorem 1.1 by "widening the separation" between ESPACE and P/Poly $\cap$ ESPACE. As a matter of fact, our development of resource-bounded category and measure began with the following question. Among languages in ESPACE, is the phenomenon of not having small (e.g., polynomial-size) circuits rare or is it in some sense typical? In section 7 we show that the phenomenon is very typical in the senses of category and measure. For example, as a subset of ESPACE, P/Poly $\cap$ ESPACE is meager and has measure 0.

Although the results in section 7 were originally proven directly, they are proven here as easy consequences of the results in section 6, where we investigate the relationships between (nonuniform) resource-bounded Kolmogorov complexity and uniform time and space complexity. Our starting point for this is the following known fact, which is roughly equivalent to Theorem 1.1.

Theorem 1.2 (Huynh [1986b]). There is a language $L \in$ ESPACE such that $KS[2^n](L_{\leq n}) > 2^{n-1}$ a.e. □

In section 6 we extend this existence theorem by proving an "abundance

theorem" which says that a comeager, measure 1 set of the languages L in ESPACE have $KS[2^n](L_{\leq n}) > 2^n$ i.o. It is interesting to note that the category portion of this result is proven by formulating the "voting argument," by which Theorems 1.1 and 1.2 were originally proven, as a winning strategy for a resource-bounded Banach-Mazur game. Moreover, playing this strategy against itself immediately gives Huynh's proof of Theorem 1.2.

In sections 6 and 7 we also investigate nonuniform complexity in exponential time classes, but the results here are less satisfying. As mentioned earlier, an analogue of Theorem 1.1 for E is conjectured but will probably be very hard to prove. The same holds for Theorem 1.2. Nevertheless, we can generalize the notion of exponential time to more accurately pinpoint the limits of relativizable methods, and then prove category and measure results right up to these limits.

To this end, we introduce the G-hierarchy G_0 , G_1 , ... in section 2. Each G_i is a class of functions from $\mathbb{N}$ to $\mathbb{N}$, these functions being regarded as growth rates. The class G_0 contains all linearly bounded growth rates and the class G_1 contains all polynomially bounded growth rates. Each class G_i is closed under composition and each class G_{i+1} contains growth rates which asymptotically dominate all growth rates in G_i . Thus, for $i > 1$, G_i contains superpolynomial growth rates. Nevertheless, every element of $\bigcup_{i=0}^{\infty} G_i$ is $o(2^n)$, i.e., subexponential.

We then define a hierarchy E_1 , E_2 , ... via $E_i = DTIME(2^{G_{i-1}})$. The first two levels of this hierarchy are the widely studied exponential time complexity classes $E_1 = DTIME(2^{linear}) = E$ and $E_2 = DTIME(2^{polynomial}) = EXP$. Here we use the expression "exponential time complexity class" to refer to any of the classes E_i . In sections 6 and 7 we investigate nonuniform

complexity in these classes. Among other things, we show that P/Poly is meager and has measure 0 in E_3 and that SIZE(n^k) is meager and has measure 0 in E_2 . Since Wilson [1985] has exhibited oracles relative to which $E_2 \subseteq$ P/Poly and E $\subseteq$ SIZE(n), these results are essentially the strongest that can be proven by relativizable means.

The second reason for studying the structure of exponential complexity classes is the recent emergence of results which relate questions about these classes to open questions about classes at lower levels. As just one example, we note that Hartmanis and Yesha [1984] have shown that P/Poly $\cap$ PSPACE $\neq$ P if and only if E $\neq$ ESPACE.

The third reason for studying the structure of exponential complexity classes is derivative from the first, and is the motivation for sections 8 through 10 of this thesis. This is the fact that, unlike lower-level classes, the exponential classes have been proven to contain intractable problems. For the purpose of proving intractability of specific problems - arguably the most important objective of complexity theory - this existence of intractability is a valuable resource.

In practice, proofs that specific languages L are intractable have taken the following three-part form.

(i) A complexity class C is shown by diagonalization to contain an intractable language. (The language so constructed does not correspond to any natural problem.)

(ii) The specific language L is then shown to be polynomial-time *hard* for C, i.e., it is shown that every language in C is polynomial-time reducible to L.

(iii) It is inferred from (i) and (ii) that some intractable problem is

reducible to L, whence L itself must be intractable.

Thus the structure of the class $\mathcal{C}$ under polynomial-time reducibility allows us to infer the intractability of a specific problem from the existence of intractability in $\mathcal{C}$.

The advantage of this method is that part (ii), a "positive" assertion about what *can* be efficiently computed, is easier to establish by known methods than a direct proof of the "negative" assertion that L *cannot* be efficiently computed.

For example, a number of problems are now known to be intractable because they are polynomial-time hard for E or ESPACE. (For examples, see Meyer and Stockmeyer [1972] and Stockmeyer and Chandra [1979].) Similarly, the real significance of the P versus NP question is the fact that many extremely important computational problems are known to be hard for NP, so a proof by any means that $P \neq NP$ will imply immediately that these problems are not in P.

The properties of languages which are hard for various complexity classes have been investigated extensively. Recently it has been shown that the intractability of hard languages for E and ESPACE also includes lower bounds for "approximate recognition." In particular, the following two facts are known.

Theorem 1.3 (Schöning [1986], Huynh [1986a]). If L is $\leq_m^P$-hard for E, then L is 2^{n^c} far from P for some $c > 0$. □

Theorem 1.4 (Huynh [1986b]). If L is $\leq_T^P$-hard for ESPACE, then L is 2^{n^c} far from P for some $c > 0$. □

Unfortunately, most problems which we would like to prove intractable

are probably not hard for such large classes as E or ESPACE. Efforts to prove the intractability of these problems have thus focused on carrying out part (i) of the above method for smaller classes $\mathcal{C}$.

Here we propose a different remedy. Let $\mathcal{R}(L)$ be the set of languages which are polynomial-time reducible to L. Part (ii) of the above method requires us to show that $\mathcal{C} \subseteq \mathcal{R}(L)$, i.e., that L contains *all* information about $\mathcal{C}$ in $\mathcal{R}$-accessible form. In sections 9 and 10 we prove that, for $\mathcal{C} = E$ and $\mathcal{C} = ESPACE$, it suffices just to prove that $\mathcal{R}(L) \cap \mathcal{C}$ is a non-small subset of $\mathcal{C}$, i.e., that L contains a "substantial amount" of information about $\mathcal{C}$. Specifically, we prove that the conclusion of Theorem 1.3 holds if a nonmeager subset of the languages in E are $\leq_m^p$-reducible to L. Similarly, the conclusion of Theorem 1.4 holds if either a nonmeager or a non-measure 0 subset of the languages in ESPACE are $\leq_T^{PSPACE}$-reducible to L. Stated in the contrapositive, these results say that any language which is feasibly approximable contains very little accessible information about the class $\mathcal{C}$. In the course of proving these results we also prove that "most" languages in E and ESPACE are intractable, even to approximation.

Although it appears that we have greatly weakened the hypotheses of Theorems 1.3 and 1.4, this has not been proven. It is conceivable that every language which contains nonmeager or non-measure 0 accessible information about one of these classes is actually hard for that class. In section 8, after introducing some notation, we formulate some *partial information hypotheses*. These are conjectures which assert the existence of languages $L \in \mathcal{C}$ such that $\mathcal{R}(L) \cap \mathcal{C}$ is not small in $\mathcal{C}$ and also does not equal $\mathcal{C}$, i.e., L contains "accessible partial information" about $\mathcal{C}$. If these conjectures hold, and can be proven, then the methods provided by sections 9 and 10 may lead

to interesting intractability proofs.

The main results of sections 6 through 10 are the extensions of Theorems 1.1, 1.2, 1.3, and 1.4 (Theorems 7.2, 6.1, 10.6, and 9.3, respectively), the analysis of nonuniform complexity versus exponential time (Theorems 6.6 and 7.6), and the fact that "most" languages in E and ESPACE are hard to approximate (Theorems 9.1 and 10.5).

It should be noted here that the very interesting "highness" and "lowness" properties investigated by Schöning [1983], Balcázar, Book, and Schöning [1986], and others are somewhat analogous to the notions of accessible information content introduced in section 8.

2. Preliminaries

If X and Y are sets, then $X \backslash Y = \{x \in X \mid x \notin y\}$, $X \Delta Y = (X \backslash Y) \cup (Y \backslash X)$ is the symmetric difference of X and Y, and $|X|$ is the cardinality of X.

We write $\{0,1\}^*$ for the set of all finite binary strings, $|x|$ for the length of a string x, λ for the empty string, $\{0,1\}^+$ for $\{0,1\}^* \backslash (\lambda)$, $\{0,1\}^{\leq n}$ for $\{x \in \{0,1\}^* \mid |x| \leq n\}$, and $\{0,1\}^n$ for $\{x \in \{0,1\}^* \mid |x| = n\}$. We fix the lexicographic enumeration $s_0 = \lambda$, $s_1 = 0$, $s_2 = 1$, $s_3 = 00$, $\cdots$ of $\{0,1\}^*$ and let next be the successor function for this enumeration, i.e., $\text{next}(s_k) = s_{k+1}$. We write $x \sqsubseteq y$ to indicate that x is an initial substring of y, i.e., $y = xz$ for some z.

All functions, unless otherwise stated, are from $\{0,1\}^*$ into $\{0,1\}^*$. Such functions are also called *transductions*. We write f^n for the n-fold composition of f with itself.

We say a condition $\Theta(n)$ holds *almost everywhere* (*a.e.*) if it holds for all but finitely many $n \in \mathbb{N}$. We say $\Theta(n)$ holds *infinitely often* (*i.o.*) if it holds for infinitely many $n \in \mathbb{N}$.

All *languages* here are sets $L \subseteq \{0,1\}^*$; we write $\mathcal{P}(\{0,1\}^*)$ for the set of all languages. We identify a language L with its *characteristic bitstring* $b_0 b_1 b_2 \ldots$, where b_k is 1 if $s_k \in L$ and 0 otherwise. A string x is an *initial bitmap* of a language L, and we write $x \sqsubseteq L$, if x is an initial substring of the characteristic bitstring of L. We write $L_{\leq n}$ for $L \cap \{0,1\}^{\leq n}$ and $L_{=n}$ for $L \cap \{0,1\}^n$.

Our model of algorithmic computation is the multitape Turing machine (TM). We write REC for the set of languages L which can be recognized by a deterministic TM. For a resource bound $t: \mathbb{N} \to \mathbb{N}$ we write DTIME(t)

(respectively, DSPACE(t)) for the set of languages L which can be decided by a deterministic TM which halts in O(t(n)) steps (respectively, after using O(t(n)) workspace) on inputs of length n. Similarly, NTIME(t) is the set of languages L which can be accepted by a nondeterministic TM which halts in O(t(n)) steps on all accepting computations. We mention the following well-known uniform complexity classes.

$$P = \cup\{DTIME(n^k) \mid k \in \mathbb{N}\}$$

$$NP = \cup\{NTIME(n^k) \mid k \in \mathbb{N}\}$$

$$PSPACE = \cup\{DSPACE(n^k) \mid k \in \mathbb{N}\}$$

$$E = \cup\{DTIME(2^{kn}) \mid k \in \mathbb{N}\}$$

$$ESPACE = \cup\{DSPACE(2^{kn}) \mid k \in \mathbb{N}\}$$

$$EXP = \cup\{DTIME(2^{n^k}) \mid k \in \mathbb{N}\}$$

$$EXPSPACE = \cup\{DSPACE(2^{n^k}) \mid k \in \mathbb{N}\}$$

For each $i \in \mathbb{N}$ we define a class G_i of functions from $\mathbb{N}$ into $\mathbb{N}$ as follows.

$$G_0 = \{f \mid (\exists k) f(n) \leq kn \text{ a.e.}\}$$

$$G_{i+1} = 2^{G_i(\log n)} = \{f \mid (\exists g \in G_i) f(n) \leq 2^{g(\log n)} \text{ a.e.}\}$$

For $i \geq 1$, we define the function $\hat{g}_i \in G_i$ by $\hat{g}_1(n) = n^2$, $\hat{g}_{i+1}(n) = 2^{\hat{g}_i(\log n)}$. We will think of the functions in these classes as growth rates. In particular, G_0 contains the linearly bounded growth rates and G_1 contains the polynomially bounded growth rates. It is easy to prove by induction that for each $i \in \mathbb{N}$, the following hold.

(i) G_i is closed under composition.

(ii) For each $f \in G_i$, $f(n) = o(\hat{g}_{i+1}(n))$.

(iii) $\hat{g}_{i+1}(n) = o(2^n)$.

Thus, for each $i > 1$, G_i contains superpolynomial growth rates, but all growth rates in the G-hierarchy are subexponential.

Using the G-hierarchy, we define, for $i \geq 1$, the following uniform complexity classes.

$$E_i = \text{DTIME}(2^{G_{i-1}}) = \cup\{\text{DTIME}(2^g) \mid g \in G_{i-1}\}$$

$$E_i\text{SPACE} = \text{DSPACE}(2^{G_{i-1}})$$

Using the standard hierarchy theorems for deterministic time and space, it is clear that $E_i \subsetneq E_{i+1}$ and $E_i\text{SPACE} \subsetneq E_{i+1}\text{SPACE}$ for each $i \geq 1$. It is also clear that $E_1 = E$, $E_2 = EXP$, $E_1\text{SPACE} = ESPACE$, and $E_2\text{SPACE} = EXPSPACE$, i.e., the first two levels of these hierarchies are the well-known exponential complexity classes. We generalize this terminology by calling each E_i an exponential time complexity class and each $E_i\text{SPACE}$ an exponential space complexity class. (Note that all these are well below doubly exponential levels.) The class $E_3 = \text{DTIME}\left(2^{n^{(\log n)^{O(1)}}}\right)$ will be of particular interest.

We also define some classes of transductions, i.e., of functions which transform strings. The computational model we use for this is the TM transducer, which is a TM augmented with a write-only output tape, the contents of which are not counted as workspace. To avoid confusing transduction classes with complexity classes of languages, we will write transduction classes using lower-case letters. The following classes are used.

$$\text{all} = \{f \mid f : \{0,1\}^* \to \{0,1\}^*\}$$

$$\text{rec} = \{f \in \text{all} \mid f \text{ is recursive}\}$$

$$p_i = \{f \in \text{all} \mid f \text{ is computable in } G_i \text{ time}\}$$

$$p_i\text{space} = \{f \in \text{all} \mid f \text{ is computable in } G_i \text{ space}\}$$

$$\log^i\text{space} = \{f \in \text{all} \mid f \text{ is computable in } O((\log n)^i) \text{ space}\}$$

$$\text{polylogspace} = \bigcup_{i \geq 1} \log^i\text{space}$$

We write p, pspace, and logspace for p_1, $p_1$space, and $\log^1$space, respectively.

If L_1 and L_2 are languages, then a *many-one reduction* of L_1 to L_2 is a transduction g such that for all $x \in \{0,1\}^*$, $x \in L_1$ iff $g(x) \in L_2$. As usual, then, L_1 is *polynomial-time many-one reducible* to L_2, and we write $L_1 \leq_m^P L_2$, if some $g \in p$ is a many-one reduction of L_1 to L_2.

For Turing reducibilities we use oracle machines. An oracle machine is a TM M augmented with a write-only oracle tape. An arbitrary language $A \subseteq \{0,1\}^*$ may be used as the oracle. During any cycle of execution, the machine is allowed to "query the oracle," i.e., to base its next action on whether or not the string currently written on the oracle tape is an element of A. We write $L(M^A)$ to denote the language decided by M when equipped with the oracle A. We then say L_1 is *polynomial-time Turing reducible* to L_2, and write $L_1 \leq_T^P L_2$, if there is a polynomial-time-bounded oracle machine M such that $L_1 = L(M^{L_2})$. The *polynomial-space Turing reducibility* $\leq_T^{PSPACE}$ is defined analogously, with the convention that the oracle tape is counted as workspace. For $i \geq 1$, we also consider the Turing reducibility $\leq_T^{P_i}$, which is like $\leq_T^P$, except that the oracle machine is G_i-time-bounded.

For a fixed TM M, a resource bound t, a language L, and $n \in \mathbb{N}$, the *t-time-bounded Kolmogorov complexity* of $L_{\leq n}$ relative to M is

$$KT_M[t](L_{\leq n}) = |\pi| ,$$

where π is the shortest "program," i.e., binary string, such that for each $x \in \{0,1\}^{\leq n}$, the machine M on input $\langle \pi, x \rangle$ correctly decides in $\leq t$ steps whether $x \in L$. Since $\{0,1\}^{\leq n} = \{s_0, \ldots, s_{2^n-2}\}$, we abbreviate this condition by saying that $M\left(\langle \pi, s_i \rangle\right)_{i=0}^{2^n-2} = L_{\leq n}$ in $\leq t$ time. If there is no such program, then $KT_M[t](L_{\leq n}) = \infty$. The *space-bounded Kolmogorov complexity* $KS_M[t](L_{\leq n})$ is defined similarly, except that M decides membership in L using

$\leq$ t cells of worktape.

It is well-known (see Huynh [1986b], for example) that there exist a universal TM U and a polynomial p such that for each TM M there is a constant c such that the following hold for all t, L, and n.

(i) $KT_U[t](L_{\leq n}) \leq KT_M[p(ct + c)](L_{\leq n}) + c.$

(ii) $KS_U[t](L_{\leq n}) \leq KS_M[ct + c](L_{\leq n}) + c.$

As usual, we fix such a universal machine U and omit it from the notation. The time and space-bounded Kolmogorov complexities $KT[t](L_{=n})$ and $KS[t](L_{=n})$ are defined analogously.

All *circuits* here are Boolean (combinational, acyclic) circuits over the basis {and, or, not, 0, 1}. A circuit has some number n of inputs and a distinguished output gate at which it computes a set $S \subseteq \{0,1\}^n$ in the usual way. The *size* of a circuit c is the number size(c) of gates in c. (Inputs are not gates.) The *circuit-size compexity* of a language L is the function $CS_L : \mathbb{N} \to \mathbb{N}$ defined by

$$CS_L(n) = \min\{\text{size}(c) \mid c \text{ computes } L_{=n}\}.$$

We will insist that a *circuit-size bound* be a function $f:\mathbb{N} \to \mathbb{N}$ which is nonzero, computable in space polynomial in n, and such that $\lim_{n\to\infty} n2^{-n}f(n)$ exists (or is infinite). For each such f we define the nonuniform complexity class

$$SIZE(f) = \{L \subseteq \{0,1\}^* \mid CS_L = O(f)\}.$$

We call f *trivial* if $SIZE(f) = \mathcal{P}(\{0,1\}^*)$, otherwise *nontrivial*. Well-known theorems of Lupanov and Shannon establish that a circuit-size bound f is nontrivial if and only if $\lim_{n\to\infty} n2^{-n} f(n) = 0$. For any circuit-size bound f, we note that SIZE(f) has the cardinality of the continuum and hence contains many nonrecursive languages. Finally, we define the set $P/Poly = \cup\{SIZE(n^k) \mid k \in \mathbb{N}\}$

of all languages which have polynomial-size circuits.

Following Yesha [1983], Schöning [1986], and others, a language L is f(n) *close* to a complexity class $\mathcal{C}$ if there is a language $L' \in \mathcal{C}$ such that $|(L \triangle L')_{\leq n}| < f(n)$ a.e. Otherwise, L is f(n) *far* from $\mathcal{C}$.

3. Resource-Bounded Baire Category

In this section we introduce and develop a general formulation of notions of Baire category on $\mathcal{P}(\{0,1\}^*)$. The formulation is general enough to admit the classical and effective notions as special cases, but its real significance is that it admits *resource-bounded* versions of Baire category which will be of use in computational complexity theory.

We will define a notion of category to be a class of functions from $\{0,1\}^*$ to $\{0,1\}^*$ which contains certain initial functions and is closed under certain operations. Our first task is thus to specify these initial functions and operations.

We fix once and for all a one-to-one *pairing function* $\langle\ ,\ \rangle$ from $\{0,1\}^*$ onto $\{0,1\}^*$ such that the pairing function and its associated *projections* $\pi_1(\langle x,y\rangle) = x$, $\pi_2(\langle x,y\rangle) = y$ are computable in logarithmic space. We insist further that $\langle x,y\rangle \in \{0\}^*$ iff $x, y \in \{0\}^*$. This latter condition canonically induces a pairing function $\langle\ ,\ \rangle$ from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$. We write $\langle x,y,z\rangle$ for $\langle x,\langle y,z\rangle\rangle$, etc., so that tuples of any fixed length are coded by the pairing function.

By the *conditional function* we mean the function $\mathrm{cond}(\langle x,y,z,w\rangle)$ whose value is z if $x \sqsubseteq y$ and w otherwise.

The *composition* $f \circ g$, *concatenation* fg, and *pairing* $\langle f,g\rangle$ of two functions f and g are defined by $(f \circ g)(x) = f(g(x))$, $(fg)(x) = f(x)g(x)$, and $\langle f,g\rangle(x) = \langle f(x),g(x)\rangle$, respectively.

The functions and operations defined thus far are natural and somewhat standard in the theory of subrecursive function classes. Two more operations which will be needed here are specified in the following definition. These

operations are a little more awkward to state than the preceding ones but are natural in the context of resource-bounded computation. Both are called "inversion" operations because they involve reconstructing ways in which a string could have been built up by recursion.

For a function f and $k \in \mathbb{N}$ it will be convenient to define the function $f_k(x) = f(<0^k,x>)$. The function f can thus be considered a "uniform enumeration" of the functions $f_0, f_1, \cdots$.

Definition 3.1. Let f be a function.

1) The *type I inversion* of f is $f^I(x) = <0^k,x_k>$, where k is maximum such that the sequence $x_0 = \lambda$, $x_{n+1} = f_n(x_n)$ satisfies

$$x_0 \subsetneq x_1 \subsetneq \cdots \subsetneq x_k \sqsubseteq x .$$

2) The *type II inversion* of f is $f^{II}(x) = 0^m$, where $m = 0$ if $x = \lambda$ and otherwise m is maximum such that there exists a sequence $x_0, \ldots, x_{m-1}$ satisfying

$$\lambda \subsetneq x_0 \subsetneq f_0(x_0) \subsetneq \cdots \subsetneq x_{m-1} \subsetneq f_{m-1}(x_{m-1}) \subsetneq x .$$

Since $f^I(x)$ and $f^{II}(x)$ specify "internal" properties of the string x with respect to f, it is also natural to think of these inversion operations as "internal primitive recursion" and "internal search recursion," respectively.

The general formulation of notions of Baire category can now be stated and developed.

Definition 3.2. A *notion of category* is a class Γ of functions from $\{0,1\}^*$ into $(0,1)^*$ which contains the projections, all constant functions, and the conditional function, and which is closed under composition, concatenation, pairing, and type I inversion. A notion of category Γ has the *Mazur property* if it is also closed under type II inversion.

From now on, Γ, Γ', etc. will denote notions of category.

The essential link between Definition 3.2 and the development of a Baire category theory for $\mathcal{P}(\{0,1\}^*)$ is the simple observation that binary languages can be constructed by using functions from $\{0,1\}^*$ into $\{0,1\}^*$.

A *constructor* is a function γ which satisfies $x \sqsubsetneq \gamma(x)$ for all $x \in \{0,1\}^*$. The *result* of a constructor γ (i.e., the language *constructed* by γ) is the unique language $R(\gamma)$ such that $\gamma^k(\lambda) \sqsubseteq R(\gamma)$ for every $k \in \mathbb{N}$. A *clocked constructor* is a function γ which satisfies $x \sqsubsetneq \gamma(\langle w,x\rangle)$ for all $w \in \{0\}^*$ and $x \in \{0,1\}^*$. The *result* of a clocked constructor γ is the unique language $R(\gamma)$ such that $x_k \sqsubseteq R(\gamma)$ for every $k \in \mathbb{N}$, where $x_0 = \lambda$ and $x_{k+1} = \gamma_k(x_k)$. (No confusion will result from the obvious ambiguity here.)

Intuitively, a constructor or clocked constructor γ builds a language $R(\gamma)$ by starting with λ and then iteratively generating successively longer initial bitmaps of $R(\gamma)$. A clocked constructor is permitted to have an "agenda that varies over time," i.e., to extend the initial bitmap in a way which depends upon the stage k of the construction.

For each Γ, Γ_c denotes the set of all constructors in Γ and Γ_{cc} denotes the set of all clocked constructors in Γ. It is then natural to define the sets $R(\Gamma_c)$, $R(\Gamma_{cc})$ of all $R(\gamma)$ such that $\gamma \in \Gamma_c$, $\gamma \in \Gamma_{cc}$, respectively.

Lemma 3.3. If Γ is a notion of category, then $R(\Gamma_c) = R(\Gamma_{cc})$.

Proof. If $\gamma \in \Gamma_c$, then $\gamma \circ \pi_2 \in \Gamma_{cc}$ and $R(\gamma \circ \pi_2) = R(\gamma)$, so $R(\gamma) \in R(\Gamma_{cc})$. Conversely, if $\gamma \in \Gamma_{cc}$, then $\gamma \circ \gamma^I \in \Gamma_c$ and $R(\gamma \circ \gamma^I) = R(\gamma)$, so $R(\gamma) \in R(\Gamma_c)$. □

Lemma 3.3 says that it makes no difference whether constructors or clocked constructors are used in a notion of category. This justifies the

following definition.

Definition 3.4. The *result class* of Γ is $R(\Gamma) = R(\Gamma_c) = R(\Gamma_{cc})$.

The following routine lemma is the reason for our interest in the transduction classes defined in section 2.

Lemma 3.5. The classes all, rec, $p_i (i \geq 1)$, $p_i$space$(i \geq 1)$, $\log^i$space$(i \geq 1)$, and polylogspace are notions of category with result classes as follows.

(i) $R(\text{all}) = \mathcal{P}(\{0,1\}^*)$.

(ii) $R(\text{rec}) = \text{REC}$.

(iii) $R(p_i) = E_i$.

(iv) $R(p_i\text{space}) = E_i\text{SPACE}$.

(v) $R(\log^i\text{space}) = \text{DSPACE}(n^i)$.

(vi) $R(\text{polylogspace}) = \text{PSPACE}$.

The classes all, rec, and $p_i$space $(i \geq 1)$ have the Mazur property. □

The classes all and rec are, respectively, the *classical* and *effective* notions of category on $\mathcal{P}(\{0,1\}^*)$. The classes p_i , $p_i$space, $\log^i$space, and polylogspace are *resource-bounded* notions of category. Of course many other such notions can be defined, e.g., by relativization, variation of the resources or bounds, etc.

We conjecture that the time-bounded classes p_i do not have the Mazur property, but this is probably hard to prove, since it implies, for example, that p does not contain every function which is computable on-line in polynomial space.

The significance of a notion of category Γ lies in the structure it imposes on $\mathcal{P}(\{0,1\}^*)$ and on the result class $R(\Gamma)$. In particular, these structures yield natural notions of "smallness" for subsets of $\mathcal{P}(\{0,1\}^*)$ and

$R(\Gamma)$. The structures on $\mathcal{P}(\{0,1\}^*)$ and on $R(\Gamma)$ are analogous and will be developed in parallel.

Definition 3.6. For each $x \in \{0,1\}^*$, the *basic set* about x in $\mathcal{P}(\{0,1\}^*)$ is $B_x = \{L \subseteq \{0,1\}^* \mid x \sqsubseteq L\}$. The corresponding *basic set* about x in $R(\Gamma)$ is $B_x \cap R(\Gamma)$.

Definition 3.7. A set $Z \subseteq \mathcal{P}(\{0,1\}^*)$ is Γ-*nowhere dense* (respectively, *nowhere dense in* $R(\Gamma)$) if there exists $h \in \Gamma_c$ such that $B_{h(x)} \cap Z = \varnothing$ (respectively, $B_{h(x)} \cap R(\Gamma) \cap Z = \varnothing$) holds for every $x \in \{0,1\}^*$.

Intuitively, a set Z is Γ-nowhere dense if Γ provides sufficient resources to compute from any basic set B_x a basic set $B_y \subsetneq B_x$ which completely avoids Z.

The following lemma summarizes some easy properties of nowhere dense sets.

Lemma 3.8.

1) If Z is Γ-nowhere dense, then Z is nowhere dense in $R(\Gamma)$.

2) If Z is Γ-nowhere dense and $\Gamma \subseteq \Gamma'$, then Z is Γ'-nowhere dense.

3a) Subsets of Γ-nowhere dense sets are Γ-nowhere dense.

3b) Subsets of sets nowhere dense in $R(\Gamma)$ are nowhere dense in $R(\Gamma)$.

4a) Finite unions of Γ-nowhere dense sets are Γ-nowhere dense.

4b) Finite unions of sets nowhere dense in $R(\Gamma)$ are nowhere dense in $R(\Gamma)$.

5) Finite subsets of $R(\Gamma)$ are Γ-nowhere dense.

Proof. Assertions 1, 2, 3a, and 3b are obvious.

To prove 4a, let Z, Z′ be Γ-nowhere dense sets, with $h, h' \in \Gamma_{cc}$ as witnesses. Then $h \circ h' \in \Gamma_c$ testifies that $Z \cup Z'$ is Γ-nowhere dense. The

result for arbitrary finite unions follows inductively. The proof of 4b is identical.

To prove 5, fix $\gamma \in \Gamma_{cc}$. By 4a it suffices to prove that the singleton set $\{R(\gamma)\}$ is Γ-nowhere dense.

Define

$$h(x) = \begin{cases} x0 & \text{if } x1 \sqsubseteq \gamma(\gamma^I(x)) \\ x1 & \text{otherwise .} \end{cases}$$

Then $h \in \Gamma_c$, so it suffices to show that $B_{h(x)} \cap \{R(\gamma)\} = \varnothing$ holds for all x. If $x \not\sqsubseteq R(\gamma)$ this is trivial, so fix $x \sqsubseteq R(\gamma)$. Let $z = \gamma(\gamma^I(x))$. Then $z \sqsubseteq R(\gamma)$ and $z \not\sqsubseteq x \sqsubseteq R(\gamma)$, so $x \sqsubsetneq z$, so either $x0 \sqsubseteq z \sqsubseteq R(\gamma)$ or $x1 \sqsubseteq z \sqsubseteq R(\gamma)$. In the first case $h(x) = x1 \not\sqsubseteq R(\gamma)$; in the second case $h(x) = x0 \not\sqsubseteq R(\gamma)$. Either way, $B_{h(x)} \cap \{R(\gamma)\} = \varnothing$. □

We are finally ready to define what it means for a set to be "small" with respect to a notion of category Γ.

Definition 3.9. A set $X \subseteq \mathcal{P}(\{0,1\}^*)$ is Γ-meager (respectively, *meager in* $R(\Gamma)$) if there exist a function $h \in \Gamma$ and a family $\{Z_k \mid k \in \mathbb{N}\}$ of sets such that

(i) $X \subseteq \cup\{Z_k \mid k \in \mathbb{N}\}$,

(ii) for each $k \in \mathbb{N}$, the function h_k testifies that Z_k is Γ-nowhere dense (respectively, nowhere dense in $R(\Gamma)$).

(Note that this forces $h \in \Gamma_{cc}$.) A set $X \subseteq \mathcal{P}(\{0,1\}^*)$ is Γ-*comeager* (respectively, *comeager in* $R(\Gamma)$) if its complement $\mathcal{P}(\{0,1\}^*)\backslash X$ is Γ-meager (respectively, meager in $R(\Gamma)$).

Thus a set X is Γ-meager if Γ provides sufficient resources to "uniformly enumerate" a family of Γ-nowhere dense sets which covers X. More concisely, X is Γ-meager if it is contained in a "Γ-union of Γ-nowhere

dense sets."

If Γ = all, then "Γ-meager" and "meager in $R(\Gamma)$" are both equivalent to the classical notion of "meager set", i.e., "set of first category". If Γ = rec, then "meager in $R(\Gamma)$" is equivalent to the effective notion of meagerness in Mehlhorn [1973] and Lisagor [1979].

The rest of this section will develop the interpretation of meager sets as "small" subsets of $R(\Gamma)$. For this, we need the following definition.

Definition 3.10. A Γ-*union* of Γ-meager sets (respectively, of sets meager in $R(\Gamma)$) is a set X such that there exist a function $g \in \Gamma$ and a family $\{X_k \mid k \in \mathbb{N}\}$ of sets such that

(i) $X = \cup\{X_k \mid k \in \mathbb{N}\}$,

(ii) for each $k \in \mathbb{N}$, the function g_k testifies that X_k is Γ-meager (respectively, meager in $R(\Gamma)$).

Lemma 3.11.

0) Γ-nowhere dense sets are Γ-meager and sets nowhere dense in $R(\Gamma)$ are meager in $R(\Gamma)$.

5) Lemma 3.3 holds with "nowhere dense" replaced by "meager" throughout.

6a) Γ-unions of Γ-meager sets are Γ-meager.

6b) Γ-unions of sets meager in $R(\Gamma)$ are meager in $R(\Gamma)$.

Proof. Assertions 0, 1, 2, 3, and 5 are obvious and assertion 4 follows trivially from assertion 6.

To prove 6a, assume X is a Γ-union of Γ-meager sets with $g \in \Gamma$ and $\{X_k \mid k \in \mathbb{N}\}$ as witnesses. Then for each k, g_k testifies that X_k is Γ-meager, i.e., there is a family $\{Z_{kj} \mid j \in \mathbb{N}\}$ of sets such that $X_k \subseteq \cup\{Z_{kj} \mid j \in \mathbb{N}\}$

and each $g_{kj} = (g_k)_j$ testifies that Z_{kj} is Γ-nowhere dense. Define $h(\langle\langle y,z\rangle,x\rangle) = g(\langle y,\langle z,x\rangle\rangle)$ and $Z'_n = Z_{\pi_1(n)\pi_2(n)}$. Then $h \in \Gamma$, $X = \cup\{X_k \mid k \in \mathbb{N}\} \subseteq \cup\{Z_{kj} \mid j, k \in \mathbb{N}\} = \cup\{Z'_n \mid n \in \mathbb{N}\}$, and each $h_n = g_{\pi_1(n)\pi_2(n)}$ testifies that Z'_n is Γ-nowhere dense, so X is Γ-meager.

The proof of 6b is similar. □

Assertions 1 and 2 of Lemma 3.11, though obvious, are important because they unify results for various Γ. For example, if we prove that a set X is p-meager, then we know that it is meager in E, but we also know that it is pspace-meager, rec-meager, and all-meager, whence it is immediately meager in ESPACE, REC, and $\mathcal{P}(\{0,1\}^*)$ as well. Thus a proof that X is p-meager yields considerably more information than a proof that it is meager in E. This is why results are usually stated in terms of Γ-meagerness, even when the matter of primary interest is meagerness in $R(\Gamma)$.

In the classical case, i.e., when Γ=all, a Γ-union is simply a countable union, so assertions 3 and 6 of Lemma 3.11 say that the meager sets form a σ-*ideal* of subsets of $\mathcal{P}(\{0,1\}^*)$. Accordingly, in the general case, we interpret 3 and 6 as saying that the Γ-meager sets form a "Γ-ideal in $\mathcal{P}(\{0,1\}^*)$" and that the sets meager in $R(\Gamma)$ form a "Γ-ideal in $R(\Gamma)$". Assertion 5 then tells us that these Γ-ideals contain many sets. (Note, however, that a singleton set $\{L\}$ need not be Γ-meager if $L \notin R(\Gamma)$.)

It is natural to think of the sets in a Γ-ideal as small, provided that the Γ-ideal is *proper*, i.e., does not contain every set. The following generalization of the classical Baire category theorem establishes this for the meager sets and thereby completes our argument that meager sets can be thought of as small sets. The simple diagonalization proof is a natural extension of the classical one.

Theorem 3.12.

1) A Γ-meager set contains no basic set.

2) A set meager in $R(\Gamma)$ contains no basic set in $R(\Gamma)$.

In particular, $\mathcal{P}(\{0,1\}^*)$ is not Γ-meager and $R(\Gamma)$ is not meager in $R(\Gamma)$.

Proof. Assertion 1 follows easily from assertion 2 via part 1 of Lemma 3.11. (Alternatively, assertion 1 follows from the classical Baire category theorem via part 2 of Lemma 3.11.)

To prove 2, let $X \subseteq \mathcal{P}(\{0,1\}^*)$ be meager in $R(\Gamma)$ with $h \in \Gamma_{cc}$ and $\{Z_k \mid k \in \mathbb{N}\}$ as witnesses and let $B'_z = B_z \cap R(\Gamma)$ be a basic set in $R(\Gamma)$. Define

$$\gamma(\langle w,x\rangle) = \begin{cases} h(\langle \lambda,z\rangle) & \text{if } x = \lambda \\ h(\langle w,x\rangle) & \text{if } x \neq \lambda , \end{cases}$$

and define the sequence $x_0 = \lambda$, $x_{k+1} = \gamma_k(x_k)$.

Note five things:

(i) $\gamma \in \Gamma_{cc}$, so $R(\gamma) \in R(\Gamma)$.

(ii) $R(\gamma) \in B'_{x_k}$ for each $k \in \mathbb{N}$.

(iii) $R(\gamma) \in B'_{x_1} = B'_{h(\lambda,z)} \subseteq B'_z$.

(iv) $B'_{x_1} \cap Z_0 = B'_{h_0(z)} \cap Z_0 = \varnothing$, so $R(\gamma) \notin Z_0$.

(v) For $k \geq 1$, $x_k \neq \lambda$, so $B'_{x_{k+1}} \cap Z_k = B'_{h_k(x_k)} \cap Z_k = \varnothing$, so $R(\gamma) \notin Z_k$.

These things together imply that

$$R(\gamma) \in B'_z \setminus \cup\{Z_k \mid k \in \mathbb{N}\} \subseteq B'_z \setminus X ,$$

whence X does not contain B'_z . □

Thus the sets which are meager in $R(\Gamma)$ form a proper Γ-ideal.

4. Resource-Bounded Banach-Mazur Games

It is usually awkward to explicitly exhibit a Γ-meager set as a Γ-union of Γ-nowhere dense sets. In this section we give an alternate characterization of Γ-meager sets (and sets meager in $R(\Gamma)$) which is often easier to use when proving that a set is Γ-meager. The characterization is in terms of certain two-person infinite games of perfect information, called *Banach-Mazur games*. In the present setting, the games will be *resource-bounded* in the sense that a player may be required to play according to a strategy which can be computed within the resources provided by Γ. Thus the "perfect information" may not always be available in a usable form.

Informally, a Banach-Mazur game is an infinite game in which two players construct a language L by taking turns extending an initial bitmap of L. There is a distinguished set X of languages such that player I wins a play of the game if $L \in X$ and player II wins otherwise.

More formally, a *strategy* for a Banach-Mazur game is a constructor σ. A *play* of a Banach-Mazur game is an ordered pair (α,β) of strategies. The *result* of the play (α,β) is the language $R(\alpha,\beta) = R(\beta \circ \alpha)$. This result is the language constructed when player I uses strategy α and player II uses strategy β. If $X \subseteq \mathcal{P}(\{0,1\}^*)$ and $\sum_I$, $\sum_{II}$ are classes of functions, then $G[X;\sum_I,\sum_{II}]$ is the Banach-Mazur game in which X is the distinguished set, player I is required to use a strategy $\alpha \in \sum_I$, and player II is required to use a strategy $\beta \in \sum_{II}$. A *winning strategy* for player I in $G[X;\sum_I,\sum_{II}]$ is a strategy $\alpha \in \sum_I$ such that $R(\alpha,\beta) \in X$ holds for every $\beta \in \sum_{II}$. A *winning strategy* for player II in $G[X;\sum_I,\sum_{II}]$ is a strategy $\beta \in \sum_{II}$ such that $R(\alpha,\beta) \notin X$ holds for every $\alpha \in \sum_I$. A player *wins* $G[X;\sum_I,\sum_{II}]$ if he has a winning strategy in $G[X;\sum_I,\sum_{II}]$.

For any strategy β, we define the associated function $\beta[\ ]$ from $\{0,1\}^{+*}$ into $\{0,1\}^*$ by the recursion $\beta[\lambda] = \lambda$, $\beta[x_1,\ldots,x_{n+1}] = \beta(\beta[x_1,\ldots,x_n]x_{n+1})$. We then define the sets

$$Y_k[\beta] = \cup\{B_{\beta[u]} \mid u \in \{0,1\}^{+*}, ||u|| \geq k\},$$

$$Y[\beta] = \cap\{Y_k[\beta] \mid k \in \mathbb{N}\},$$

$$Z_k[\beta] = \mathcal{P}(\{0,1\}^*)\backslash Y_k[\beta],$$

$$Z[\beta] = \mathcal{P}(\{0,1\}^*)\backslash Y[\beta] = \cup\{Z_k[\beta] \mid k \in \mathbb{N}\}.$$

(Here $\{0,1\}^{+*}$ is the set of all sequences of nonempty binary strings and, for $u = (x_1,\ldots,x_n) \in \{0,1\}^{+*}$, $||u|| = |x_1|+\cdots+|x_n|$.) Intuitively, $\beta[x_1,\ldots,x_n]$ is the status of a Banach-Mazur game immediately after player II's n^{th} move if player I appends x_i to the bitmap in his i^{th} move and player II uses strategy β. Thus $B_{\beta[x_1,\ldots,x_n]}$ is the set of all languages which could "conceivably" result from this play of the game, no matter what strategy either player uses after player II's n^{th} move. In this same sense, $Y_k[\beta]$ is the set of all languages which could conceivably result from any play of the game in which player II uses strategy β in all moves up to and including his response to the move by which player I's total contribution to the bitmap reaches or exceeds k bits. Hence $Y[\beta]$ is the set of all languages which could result from any play of the game in which player II uses strategy β.

Note that $\beta[\ ]$ is recursive in β, but that $\beta[\ ]$ need not be in Γ, even when $\beta \in \Gamma$. Also note that $|\beta[u]| \geq ||u||$ and $\beta[u] \sqsubseteq \beta[v]$ hold for all $u, v \in \{0,1\}^{+*}$ with u an initial subsequence of v.

Lemma 4.1. For any strategy $\beta \in \Gamma$, $Z[\beta]$ is Γ-meager.

Proof. Let $\beta \in \Gamma_c$ and define

$$h(<x,y>) = \begin{cases} \beta(x) & \text{if } y = \lambda \\ \beta(\beta(y)x) & \text{if } y \neq \lambda \ . \end{cases}$$

Then $h \in \Gamma$ and for each $k \in \mathbb{N}$ we have

$$\lambda \sqsubseteq 0^k \sqsubsetneq \beta(0^k) = h(<0^k,\lambda>) = h_k(\lambda)$$

and

$$x \neq \lambda \Rightarrow x \sqsubsetneq \beta(x) \sqsubseteq \beta(x)0^k \sqsubsetneq \beta(\beta(x)0^k)$$
$$= h(<0^k,x>) = h_k(x),$$

so $h \in \Gamma_{cc}$. Thus it suffices to show that each h_k testifies that $Z_k[\beta]$ is Γ-nowhere dense, i.e., that $B_{h_k(x)} \subseteq Y_k[\beta]$ holds for each k and x. This is trivial for $k = 0$ because $Y_0[\beta] = \mathcal{P}(\{0,1\}^*)$, so assume $k > 0$. Then $B_{h_k(\lambda)} = B_{\beta(0^k)} = B_{\beta[0^k]} \subseteq Y_k[\beta]$ and for $x \neq \lambda$, $B_{h_k(x)} = B_{\beta(\beta(x)0^k)} = B_{\beta[x,0^k]} \subseteq Y_k[\beta]$, so we are finished in any case. □

If Γ is a notion of category, then it is easy to check that the class $rec(\Gamma)$ of all functions f such that f is recursive in some $g \in \Gamma$ is also a notion of category.

Lemma 4.2.

1) If β is a strategy and $x \sqsubseteq L \in Y[\beta]$, then there exist $u \in \{0,1\}^{+*}$ and $y \in \{0,1\}^+$ such that $x \sqsubsetneq \beta[u]y \sqsubsetneq \beta(\beta[u]y) \sqsubseteq L$.
2) If β is a strategy and $L \in Y[\beta]$, then there is a strategy α such that $R(\alpha,\beta) = L$.
3) If $\beta \in \Gamma$ is a strategy and $L \in Y[\beta] \cap R(\Gamma)$, then there is a strategy $\alpha \in rec(\Gamma)$ such that $R(\alpha,\beta) = L$.

Proof. Assume the hypothesis of assertion 1. Then $L \in Y_{|x|+1}[\beta]$, so there exists $u' = (x_1,\ldots,x_n) \in \{0,1\}^{+*}$ with $||u'|| > |x|$ and $L \in B_{\beta[u']}$, i.e., $\beta[u'] \sqsubseteq L$. Note that $n \geq 1$, let $u = (x_1,\ldots,x_{n-1})$, and let $y = x_n$. Then $|\beta[u]y| \geq ||u'|| > |x|$ and $\beta[u]y \sqsubsetneq \beta(\beta[u]y) = \beta[u'] \sqsubseteq L$. Since $x \sqsubseteq L$, it follows that $x \sqsubsetneq \beta[u]y$, whence 1 holds.

Assertion 2 follows trivially from assertion 3 by taking Γ = all.

To prove 3, assume the hypothesis. Then by 1 and the fact that rec(Γ) is closed under simple searching, there is a strategy $\alpha \in \mathrm{rec}(\Gamma)$ such that

(i) if $x \not\sqsubseteq L$, then $\alpha(x) = x0$;

(ii) if $x \sqsubseteq L$, then $\alpha(x) = \beta[u]y$, where u and y are such that

$$x \underset{\neq}{\sqsubseteq} \alpha(x) \underset{\neq}{\sqsubseteq} \beta(\alpha(x)) \sqsubseteq L.$$

It follows immediately that $R(\alpha,\beta) = L$. □

We now characterize the Γ-meager sets in terms of Banach-Mazur games.

Theorem 4.3. For a set $X \subseteq \mathcal{P}(\{0,1\}^*)$, consider the following conditions.

(a) Player II wins G[X ; all, Γ].

(b) X is Γ-meager.

In any case, (a) implies (b). If Γ has the Mazur property, then (b) implies (a).

Proof. Assume (a) holds with the winning strategy $\beta \in \Gamma$ as witness. Assume $L \in Y[\beta]$. By part 2 of Lemma 4.2, there is a strategy α such that $R(\alpha,\beta) = L$. Since β is a winning strategy for player II, it follows that $L \notin X$. Taking the contrapositive, this argument shows that $X \subseteq Z[\beta]$. Since $Z[\beta]$ is Γ-meager by Lemma 4.1, it follows that (b) holds.

Conversely, assume that Γ has the Mazur property and that (b) holds, i.e., that X is Γ-meager with $h \in \Gamma_{cc}$ and $\{Z_k \mid k \in \mathbb{N}\}$ as witnesses. Note that if $|h^{II}(x)| > k$, then there is a sequence $x_0, \ldots, x_k$ with

$$\lambda \underset{\neq}{\sqsubseteq} x_0 \underset{\neq}{\sqsubseteq} h_0(x_0) \underset{\neq}{\sqsubseteq} \cdots \underset{\neq}{\sqsubseteq} x_k \underset{\neq}{\sqsubseteq} h_k(x_k) \underset{\neq}{\sqsubseteq} x,$$

so $B_x \cap Z_k \subseteq B_{h_k(x_k)} \cap Z_k = \varnothing$, the last equality holding because h_k testifies that Z_k is Γ-nowhere dense. This shows that h^{II} has the property that

$$|h^{II}(x)| > k \Rightarrow B_x \cap Z_k = \varnothing \qquad (*)$$

holds for all x and k. Define $\beta(x) = h(\langle h^{II}(x),x\rangle)$. Since $h \in \Gamma_{cc}$ and Γ has the Mazur property, β is a strategy in Γ. To see that β wins $G[X ; all, \Gamma]$ for player II, let α be an arbitrary strategy for player I. It is immediate from the definitions of h^{II} and β that $|h^{II}((\beta \circ \alpha)^n(\lambda))|$ is strictly increasing in n. It follows by (*) that for each k there exists n such that $B_{(\beta\circ\alpha)^n(\lambda)} \cap Z_k = \varnothing$. Since $(\beta \circ \alpha)^n(\lambda) \sqsubseteq R(\alpha,\beta)$ for each n, it follows that $R(\alpha,\beta) \notin Z_k$ for each k. But then $R(\alpha,\beta) \notin X$, i.e., player II wins, so (a) holds. □

Analogously, we characterize the sets which are meager in $R(\Gamma)$.

Theorem 4.4. For a set $X \subseteq \mathcal{P}(\{0,1\}^*)$, consider the following conditions.

(a) Player II wins $G[X \cap R(\Gamma); all, \Gamma]$.

(b) Player II wins $G[X \cap R(\Gamma); rec(\Gamma), \Gamma]$.

(c) X is meager in $R(\Gamma)$.

In any case, (a) implies (b) and (b) implies (c). If Γ has the Mazur property, then (c) implies (a).

Proof. It is trivial that (a) implies (b).

The proof that (b) implies (c) is the same as the proof that (a) implies (b) in Theorem 4.3, except that $X \cap R(\Gamma)$ is used in place of X and part 3 of Lemma 4.2 is used in place of part 2.

If Γ has the Mazur property, then the proof that (c) implies (a) is the same as the proof that (b) implies (a) in Theorem 4.3, except that $X \cap R(\Gamma)$ is used in place of X and basic sets $B_z \cap R(\Gamma)$ in $R(\Gamma)$ are used in place of basic sets B_z. □

In the case where Γ has the Mazur property, the equivalence of (a) and (b) in Theorem 4.4 is somewhat remarkable. For example, if Γ = pspace and

$X \subseteq$ ESPACE, this says that player II wins G[X; all, pspace] if he wins G[X; rec, pspace]. That is, if player II can beat every recursive strategy, he can beat every strategy whatsoever. Intuitively, this says that, in the game G[X; all, pspace], most of player I's available resources are no help to him.

The role of the Mazur property in Theorems 4.3 and 4.4 illustrates an interesting aspect of the resource-bounded setting. In the classical and effective settings (where the Mazur property holds trivially), the fact that player II can win whenever the designated set is meager is the "easy direction" of the characterization. (This direction was noted by Mazur when he invented the classical game; it was Banach who subsequently proved the converse.) In the resource-bounded setting, this direction seems to require an additional property, which we have called the Mazur property. For example, it is not clear that player II wins G[X; rec, p], or even G[X;p,p], whenever X is a meager subset of E.

We conclude this section with an easy application. A language L is *sparse* if there is a polynomial q such that $|L_{=n}| \leq q(n)$ for all n. The sparse languages are a central concern of current research in computational complexity theory. It is easy to see that the set SPARSE of all sparse languages has the cardinality of the continuum, i.e., of $\mathcal{P}(\{0,1\}^*)$. The following theorem shows that SPARSE is nevertheless small in the logarithmic space-bounded sense of category.

Theorem 4.5. SPARSE is logspace-meager, hence meager in DSPACE(n).

Proof. Consider the strategy β that extends x by appending $4|x| + 1$ 1's to it. It is clear that β is a constructor and $\beta \in$ logspace. It is also easy to check that, for any strategy α, there are infinitely many n such that $|R(\alpha,\beta)_{=n}| = 2^n$, whence $R(\alpha,\beta) \notin$ SPARSE. Thus β wins G[SPARSE; all,

logspace] for player II, so the present theorem follows from Theorem 4.3. □

5. Resource-Bounded Measure

The sense in which meager sets are small is not always the most intuitive one. For example, consider the set X of all languages L such that $xx \sqsubseteq L$ only holds for finitely many strings x. The strategy $\beta(x) = xx$ testifies readily that X is p-meager. However, if L is chosen probabilistically by using an independent toss of a fair coin to decide whether each $s_i \in L$, then it is easy to see that L will be in X with probability 1, i.e., almost certainly. Thus the Baire category notion of smallness disagrees sharply with this very intuitive probability measure on $\mathcal{P}(\{0,1\}^*)$.

This independent coin-toss measure is precisely the classical Lebesgue measure on $\mathcal{P}(\{0,1\}^*)$. That is, if we identify a language L with the real number $x \in [0,1]$ whose binary expansion is the characteristic bitstring of L, then the measure of a set of languages is (when defined) precisely the usual Lebesgue measure of the corresponding subset of the unit interval. Equivalently, this measure is the product probability measure on $\prod_{x \in \{0,1\}^*} \{0,1\}$, where $\{0,1\}$ has the uniform distribution.

In this section we introduce and develop *resource-bounded (Lebesgue) measure*, i.e., *resource-bounded probability*, for complexity classes of languages. This will provide a notion of smallness for subsets of these classes which corresponds nicely with the classical Lebesgue measure on $\mathcal{P}(\{0,1\}^*)$.

The subject of this thesis is resource-bounded category and measure in exponential complexity classes. It will thus suffice here to say that a *notion of measure* is a class Δ of functions from $\{0,1\}^*$ into $\{0,1\}^*$and that the classes all, rec, p_i $(i \geq 1)$, and $p_i$space $(i \geq 1)$ are notions of measure. Some results will be stated in terms of arbitrary notions of measure Δ, but we will only require their proofs to be valid for these examples. This approach is

less general than that of section 3 but is still general enough to encompass classical, effective, and resource-bounded notions.

From now on, Δ, Δ', etc., will denote notions of measure in the above sense. The result classes $R(\Delta)$ are defined exactly as in section 3, so the language classes treated here are $\mathcal{P}(\{0,1\}^*)$, REC, E_i ($i \geq 1$), and E_iSPACE ($i \geq 1$).

Definition 5.1. The *measure* of a basic set B_x is $\mu(B_x) = 2^{-|x|}$. The *measure* of the empty set is $\mu(\phi) = 0$.

Definition 5.2. A Δ-*cover* is a pair $(h,m) \in \Delta^2$ such that $m(\{0\}^*) \subseteq \{0\}^*$ and

$$\sum_{k=|m(0^\ell)|}^{\infty} \mu\left[B_{h(0^k)}\right] \leq 2^{-\ell}$$

holds for each $\ell \in \mathbb{N}$. If (h,m) is a Δ-cover, then h is called the *enumerator*, m is called the *modulus*, and the real number

$$\mu^*(h) = \sum_{k=0}^{\infty} \mu\left[B_{h(0^k)}\right]$$

exists and is called the *total measure* of (h,m).

Intuitively, a Δ-cover is a family of basic sets $B_{h(\lambda)}$, $B_{h(0)}$, $B_{h(00)}$, ... such that Δ provides sufficient resources to enumerate the family and to compute approximations of the finite total measure of the family.

Definition 5.3. A Δ-*cover of* a set $X \subseteq \mathcal{P}(\{0,1\}^*)$ is a Δ-cover (h,m) such that $X \subseteq \cup\{B_{h(0^k)} | k \in \mathbb{N}\}$.

Definition 5.4. A Δ-*null cover* of a set $X \subseteq \mathcal{P}(\{0,1\}^*)$ is a pair $(h,m) \in \Delta^2$ such that the following two conditions hold for each $k \in \mathbb{N}$.

(i) (h_k, m_k) is a Δ-cover of X.

(ii) $\mu^*(h_k) \leq 2^{-k}$.

Definition 5.5. Let $X \subseteq \mathcal{P}(\{0,1\}^*)$ and let $X^c = \mathcal{P}(\{0,1\}^*)\backslash X$ be the complement of X.

1) X has Δ-*measure* O, and we write $\mu_\Delta(X) = 0$, if there exists a Δ-null cover of X.

2) X has *measure* O *in* $R(\Delta)$, and we write $\mu(X|R(\Delta)) = 0$, if $\mu_\Delta(X \cap R(\Delta)) = 0$.

3) X has Δ-*measure* 1, and we write $\mu_\Delta(X) = 1$, if $\mu_\Delta(X^c) = 0$.

4) X has *measure* 1 *in* $R(\Delta)$, and we write $\mu(X|R(\Delta)) = 1$, if $\mu(X^c|R(\Delta)) = 0$.

Thus a set X of languages has Δ-measure 0 if Δ contains sufficient resources to uniformly enumerate Δ-covers of X with rapidly vanishing total measure.

Note that $\mu(X|R(\Delta))$ depends only on the set $X \cap R(\Delta)$. In particular, the conditions $\mu(X|R(\Delta)) = 1$ and $\mu_\Delta(X \cap R(\Delta)) = 1$ are *not* equivalent.

It is amusing to think of $\mu(X|R(\Delta))$ as the "conditional probability" that $L \in X$, given that $L \in R(\Delta)$, when L is chosen by independent tosses of a fair coin. It should be emphasized, however, that this interpretation is not meaningful (and is probably misleading) because, in cases of interest, $R(\Delta)$ will be a countable, hence measure 0, subset of $\mathcal{P}(\{0,1\}^*)$.

The next definition and lemma are analogous to 3.10 and 3.11.

Definition 5.6. A Δ-*union* of Δ-measure 0 sets (respectively, of sets of measure 0 in $R(\Delta)$) is a set X such that there exist a pair $(h,m) \in \Delta^2$ and a family $\{X_k | k \in \mathbb{N}\}$ of sets such that

(i) $X = \cup\{X_k | k \in \mathbb{N}\}$,

(ii) for each $k \in \mathbb{N}$, the pair $(h_k , m_k) \in \Delta^2$ testifies that X_k has

Δ-measure 0 (respectively, has measure 0 in $R(\Delta)$).

Of course any finite union is a Δ-union here.

Lemma 5.7.

1) If X has Δ-measure 0, then X has measure 0 in $R(\Delta)$.

2) If X has Δ-measure 0 and $\Delta \subseteq \Delta'$, then X has Δ'-measure 0.

3a) Subsets of Δ-measure 0 sets have Δ-measure 0.

3b) Subsets of sets with measure 0 in $R(\Delta)$ have measure 0 in $R(\Delta)$.

4a) Δ-unions of Δ-measure 0 sets have Δ-measure 0.

4b) Δ-unions of sets with measure 0 in $R(\Delta)$ have measure 0 in $R(\Delta)$.

5) Finite subsets of $R(\Delta)$ have Δ-measure 0.

Proof. Assertions 1, 2, and 3 are clear and 4b follows immediately from 4a, so it suffices to prove 4a and 5.

To prove 4a, let $(h,m) \in \Delta^2$ and the family $\{X_k | k \in \mathbb{N}\}$ testify that X is a Δ-union of Δ-measure 0 sets. Define

$$h'(\langle u,\langle v,w\rangle\rangle) = h(\langle v,\langle uv0,w\rangle\rangle).$$

Then $h' \in \Delta$ and for each k, i, $j \in \mathbb{N}$, $h'_k(\langle 0^i , 0^j\rangle) = h_{i,k+i+1}(0^j)$. That is, h'_k "weaves together" the enumerators $h_{i,k+i+1}$ for $i \in \mathbb{N}$. Note that each $(h_{i,k+i+1} , m_{i,k+i+1})$ is a Δ-cover of X_i with total measure $\mu^*(h_{i,k+i+1}) \leq 2^{-(k+i+1)}$. In Δ we can compute a function $m':\{0,1\}^* \to \{0,1\}^*$ such that

(i) $m'(\{0\}^*) \subseteq \{0\}^*$;

(ii) for each i, j, k, $\ell \in \mathbb{N}$, the condition $\langle i,j\rangle \geq |m'_k(0^\ell)|$ implies that $i \geq \ell + 1$ or $j \geq |m_{i,k+i+1}(0^{2\ell+1})|$.

Then $(h',m') \in \Delta^2$. We will show that (h',m') is in fact a Δ-null cover of X. For this, it suffices to prove that the following three things hold for each k, $\ell \in \mathbb{N}$.

(a) $X \subseteq \cup\{B_{h'_k(0^n)} | n \in \mathbb{N}\}$.

(b) $\sum_{n=|m'_k(0^\ell)|}^{\infty} \mu\left(B_{h'_k(0^n)}\right) \leq 2^{-\ell}$.

(c) $\mu^*(h'_k) \leq 2^{-k}$.

So fix $k, \ell \in \mathbb{N}$. To prove (a) just note that each $X_i \subseteq \cup\{B_{h_{i,k+i+1}(0^j)} | j \in \mathbb{N}\}$ $= \cup\{B_{h'_k(<0^i,0^j>)} | j \in \mathbb{N}\}$, whence $X = \cup\{X_i | i \in \mathbb{N}\} \subseteq \cup\{B_{h'_k(<0^i,0^j>)} | i, j \in \mathbb{N}\}$ $= \cup\{B_{h'_k(0^n)} | n \in \mathbb{N}\}$, i.e., (a) holds. For convenience, write $\hat{m} = |m_{i,k+i+1}(0^{2\ell+1})|$.

Before proving (b) and (c), note that the following two things hold for each $i \in \mathbb{N}$.

(iii) $\sum_{j=0}^{\infty} \mu\left(B_{h'_k(<0^i,0^j>)}\right) = \mu^*(h_{i,k+i+1})$.

(iv) $\sum_{j=\hat{m}}^{\infty} \mu\left(B_{h'_k(<0^i,0^j>)}\right) \leq 2^{-(2\ell+1)}$.

Now to prove (b), note that (ii) tells us that

$$\sum_{n=|m'_k(0^\ell)|}^{\infty} \mu\left(B_{h'_k(0^n)}\right)$$

$$\leq \sum_{i=\ell+1}^{\infty} \sum_{j=0}^{\infty} \mu\left(B_{h'_k(<0^i,0^j>)}\right)$$

$$+ \sum_{i=0}^{\ell} \sum_{j=\hat{m}}^{\infty} \mu\left(B_{h'_k(<0^i,0^j>)}\right),$$

whence by (iii) and (iv) we have

$$\sum_{n=|m'_k(0^\ell)|}^{\infty} \mu\left(B_{h'_k(0^n)}\right)$$

$$\leq \sum_{i=\ell+1}^{\infty} \mu^*(h_{i,k+i+1}) + \sum_{i=0}^{\ell} 2^{-(2\ell+1)}$$

$$\leq \sum_{i=\ell+1}^{\infty} 2^{-(k+i+1)} + (\ell+1)2^{-(2\ell+1)}$$
$$\leq 2^{-\ell},$$

i.e., (b) holds. Finally, (iii) tells us that

$$\mu^*(h'_k) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \mu\left(B_{h'_k(<0^i,0^j>)}\right)$$
$$= \sum_{i=0}^{\infty} \mu^*(h_{i,k+i+1})$$
$$\leq \sum_{i=0}^{\infty} 2^{-(k+i+1)}$$
$$= 2^{-k},$$

i.e., (c) holds. Thus (h′ , m′) is a Δ-null cover of X, so 4a is affirmed.

To prove 5, it is sufficient by 4a to prove that singleton subsets of R(Δ) have Δ-measure 0. So let L = R(δ), where $\delta \in \Delta$. Define $h(<u,v>) = \delta^{n}(\lambda)$, where n is least such that $|\delta^{n}(\lambda)| > |uv|$, and $m(<u,v>) = v$. Then it is easy to check that (h,m) is a Δ-null cover of {L}. □

Assertions 1 and 2 of Lemma 5.7 have the same unifying effect as the corresponding assertions of Lemma 3.11. Thus, for example, if we prove that a set $X \subseteq \mathcal{P}(\{0,1\}^*)$ has p-measure 0, then we can conclude immediately that $0 = \mu(X|E) = \mu(X|ESPACE) = \mu(X|REC) = \mu(X)$.

Assertions 3 and 4 of Lemma 5.7 say that the Δ-measure 0 sets form a Δ-ideal in $\mathcal{P}(\{0,1\}^*)$ and that the sets which have measure 0 in R(Δ) form a Δ-ideal in R(Δ). Assertion 5 then says that these Δ-ideals contain many sets.

The following theorem is analogous to Theorem 3.12 in that it shows that these Δ-ideals are proper. The main idea of the proof is a diagonalization, which we first isolate in a lemma. (This lemma is a resource-bounded version of a classical theorem of Borel.)

Lemma 5.8. If $X \subseteq R(\Delta)$ has a Δ-cover with total measure less than $2^{-|z|}$, then X does not contain $B_z \cap R(\Delta)$.

Proof. Let (h,m) be a Δ-cover of $X \subseteq R(\Delta)$ with $\mu^*(h) < 2^{-|z|}$ and let $X' = \cup\{B_{h(0^k)} | k \in \mathbb{N}\}$. Note that $X \subseteq X'$. Fix a binary rational q and a positive integer ℓ such that $2^{|z|}\,\mu^*(h) \leq q < q + 2^{-\ell} < 1$. Define from h a real-valued "density function"

$$d(x) = 2^{|x|} \sum_{k=0}^{\infty} \mu\left[B_x \cap B_{h(0^k)}\right]$$

and an "approximate density function"

$$e(x,n) = 2^{|x|} \sum_{k=0}^{n-1} \mu\left[B_x \cap B_{h(0^k)}\right].$$

Note that each $e(x, n) \leq d(x)$ and $\lim_{n\to\infty} e(x, n) = d(x)$. Note also that for each $x \in \{0,1\}^*$,

(i) $B_x \subseteq X'$ implies $d(x) \geq 1$, and

(ii) $d(x) = \frac{1}{2}[d(x0) + d(x1)]$.

Define a clocked constructor δ by

$$\delta_k(x) = \begin{cases} x0 & \text{if } e(x0,\hat{m}) \leq q + 2^{-\ell}(1-2^{-k}) \\ x1 & \text{otherwise,} \end{cases}$$

where $\hat{m} = |m(0^{|x|+k+\ell+2})|$. Since h, $m \in \Delta$, it is easy to check that $\delta \in \Delta$.

Assume for a moment that $d(x) \leq q + 2^{-\ell}(1 - 2^{-k})$. Then (ii) tells us that $\min\{d(x0),d(x1)\} \leq q + 2^{-\ell}(1 - 2^{-k})$, whence $e(\delta_k(x), \hat{m}) \leq q + 2^{-\ell}(1 - 2^{-k})$. It follows that

$$d(\delta_k(x)) \leq e(\delta_k(x),\hat{m}) + 2^{|x|+1} \sum_{j=\hat{m}}^{\infty} 2^{-|h(0^j)|}$$

$$\leq q + 2^{-\ell}(1 - 2^{-k}) + 2^{|x|+1}\, 2^{-|x|-k-\ell-2}$$

$$= q + 2^{-\ell}(1 - 2^{-k-1}) .$$

Now define the clocked constructor $\hat{\delta}$ by $\hat{\delta}_k(x) = \delta_k(\hat{x})$, where $\hat{x} = z$ if $x \sqsubseteq z$ and $\hat{x} = x$ otherwise, and define the sequence $x_0 = \lambda$, $x_{k+1} = \hat{\delta}_k(x_k)$. It is clear that $\hat{\delta} \in \Delta$ and that $z \sqsubseteq x_1$, whence $R(\hat{\delta}) \in B_z$. It is also clear that $d(z) \leq 2^{|z|}\, \mu^*(h) \leq q$, whence the preceding paragraph provides an inductive proof that

(iii) $\quad d(x_k) \leq q + 2^{-\ell}(1 - 2^{-k})$

holds for each $k \geq 1$.

Now let $k \in \mathbb{N}$ be arbitrary and let $\hat{k} = |h(0^k)| + 1$. Then (iii) tells us that $d(x_{\hat{k}}) \leq q + 2^{-\ell} < 1$, so $B_{x_{\hat{k}}} \not\sqsubseteq X'$ by (i). In particular, then, $B_{x_{\hat{k}}} \not\sqsubseteq B_{h(0^k)}$. Since $|x_{\hat{k}}| \geq \hat{k} > |h(0^k)|$ and $R(\hat{\delta}) \in B_{x_{\hat{k}}}$, it follows that $R(\hat{\delta}) \notin B_{h(0^k)}$. Since k is arbitrary here, it follows that $R(\hat{\delta}) \notin X'$, whence $R(\hat{\delta}) \notin X$.

We now have $\hat{\delta} \in \Delta$ such that $R(\hat{\delta}) \in B_z \backslash X$. It follows that X does not contain $B_z \cap R(\Delta)$. □

Theorem 5.9.

1) A Δ-measure 0 set contains no basic set.

2) A set of measure 0 in $R(\Delta)$ contains no basic set in $R(\Delta)$.

Proof. Assertion 1 is immediate from assertion 2 via part 1 of Lemma 5.7.

To prove 2, let the null cover (h,m) testify that $\mu(X|R(\Delta)) = 0$ and fix a basic set $B_z \cap R(\Delta)$ in $R(\Delta)$. Then the pair $(h_{|z|+1}\, ,\, m_{|z|+1})$ is a Δ-cover of $X \cap R(\Delta)$ with total measure $\mu^*(h_{|z|+1}) \leq 2^{-|z|-1} < 2^{-|z|}$. It follows by

Lemma 5.8 that X does not contain $B_Z \cap R(\Delta)$. □

By Lemma 5.7 and Theorem 5.9, the measure 0 subsets of $R(\Delta)$ are "small" subsets of $R(\Delta)$.

Now once again let X be the set considered in the first paragraph of this section. For each k, $\ell \in \mathbb{N}$, let $h_k(0^\ell) = xx$, where x is the ℓ^{th} binary string of length $>k$, and let $m_k(0^\ell) = 0^{2^{\ell+1}}$. It is easy to check that (h,m) is a pspace-null cover of the complement of X, whence X has pspace-measure 1. On the other hand, we saw that X is p-meager, hence certainly pspace-meager. It follows that X is meager and has measure 1 in ESPACE. Thus, just as in the classical case, resource-bounded notions of category and measure do not always agree as to which sets are small.

It is conspicuous that the resource-bounded measure theory developed so far only assigns measures of 0 or 1 to sets of languages. We now present the basic ideas of a more comprehensive resource-bounded measure theory and explain why sets of intermediate measures are of very little interest in complexity theory.

The following definition is based on Lebesgue's original formulation of measurability via inner and outer measure.

Definition 5.10. Let $X \subseteq \mathcal{P}(\{0,1\}^*)$ and let $X^c = \mathcal{P}(\{0,1\}^*)\backslash X$ be the complement of X.

1) X is Δ-*measurable* if there is a triple $(g,h,m) \in \Delta^3$ such that the conditions

   (i) (g_k , m_k) is a Δ-cover of X,

   (ii) (h_k , m_k) is a Δ-cover of X^c,

   (iii) $\mu^*(g_k) + \mu^*(h_k) \leq 1 + 2^{-k}$

all hold for each $k \in \mathbb{N}$. In this case, the real number

$$\mu_\Delta(X) = \lim_{k\to\infty} \mu^*(g_k) = 1 - \lim_{k\to\infty} \mu^*(h_k)$$

exists and is called the Δ-*measure* of X.

2) X is *measurable in* $R(\Delta)$, i.e., is an *event in* $R(\Delta)$, if there is a triple $(g,h,m) \in \Delta^3$ such that conditions (i'), (ii'), and (iii) hold for each $k \in \mathbb{N}$, where (i') and (ii') are conditions (i) and (ii) above with X and X^c replaced by $X \cap R(\Delta)$ and $R(\Delta)\backslash X$, respectively. In this case, the real number

$$\mu(x|R(\Delta)) = \lim_{k\to\infty} \mu^*(g_k) = 1 - \lim_{k\to\infty} \mu^*(h_k)$$

exists and is called the *measure* of X *in* $R(\Delta)$.

It is easy to see that if X is Δ-measurable (respectively, measurable in $R(\Delta)$), then $\mu_\Delta(X)$ (respectively, $\mu(X|R(\Delta))$) is well-defined, i.e., does not depend on the witness (g,h,m). Also, each of the conditions $\mu_\Delta(X) = 0$, $\mu_\Delta(X) = 1$, $\mu(X|R(\Delta)) = 0$, $\mu(X|R(\Delta)) = 1$ holds under Definition 5.10 if and only if it holds under Definition 5.5.

If Δ = all, then Δ-measurability and measurability in $R(\Delta)$ are equivalent to each other and to classical Lebesgue measurability in $\mathcal{P}(\{0,1\}^*)$. Similarly, a set is measurable in R(rec) = REC if and only if it is effectively measurable in the sense of Friedzon [1972].

For an easy resource-bounded example, one can check that each B_x is p-measurable with $\mu_p(B_x) = \mu(B_x)$. It follows easily that each B_x is measurable in E with $\mu(B_x|E) = \mu(B_x)$. As a cautionary example, however, note that $B_x \cap E$ is not p-measurable for any x.

We now give two useful lemmas. The first is immediate from Definition 5.10.

Lemma 5.11.

1) If X is Δ-measurable, then X is measurable in $R(\Delta)$ and $\mu(X|R(\Delta)) = \mu_\Delta(X)$.

2) If X is Δ-measurable and $\Delta \subseteq \Delta'$, then X is Δ'-measurable and $\mu_{\Delta'}(X) = \mu_\Delta(X)$. □

Lemma 5.12.

1) If X is Δ-measurable and (h,m) is a Δ-cover of X, then $\mu^*(h) \geq \mu_\Delta(X)$.

2) If X is measurable in $R(\Delta)$ and (h,m) is a Δ-cover of $X \cap R(\Delta)$, then $\mu^*(h) \geq \mu(X|R(\Delta))$.

Proof. Assertion 1 follows immediately from assertion 2 by part 1 of Lemma 5.11.

To prove 2, let (g', h', m') testify that X is measurable in $R(\Delta)$ and let $k \in \mathbb{N}$ be arbitrary. Then (h'_k, m'_k) is a Δ-cover of $R(\Delta)\backslash X$ with $\mu^*(h'_k) \leq 2^{-k} + \lim_{n\to\infty} \mu^*(h'_n)$. Now (h,m) and (h'_k, m'_k) can be "woven together" to give a Δ-cover (h'', m'') of $R(\Delta)$ with $\mu^*(h'') = \mu^*(h) + \mu^*(h'_k)$. By Lemma 5.8, then, $1 \leq \mu^*(h'') = \mu^*(h) + \mu^*(h'_k)$, so $\mu^*(h) \geq 1 - \mu^*(h'_k) \geq 1 - \lim_{n\to\infty} \mu^*(h'_n) - 2^{-k} = \mu(X|R(\Delta)) - 2^{-k}$. Since k is arbitrary here, it follows that $\mu^*(h) \geq \mu(X|R(\Delta))$. □

The following definition formalizes what it means for a class of languages to be "insensitive to finite alterations."

Definition 5.13.

1) Two languages $L_1, L_2 \subseteq \{0,1\}^*$ are *equivalent almost everywhere*, and we write $L_1 \equiv L_2$ a.e., if their symmetric difference $L_1 \Delta L_2$ is finite.

2) A set $X \subseteq \mathcal{P}(\{0,1\}^*)$ is a *tail set* if for all L_1 , $L_2 \subseteq \{0,1\}^*$ such that $L_1 \equiv L_2$ a.e., $L_1 \in X$ if and only if $L_2 \in X$.

In complexity theory, virtually all language classes of interest are tail sets. In classical measure theory, the Kolmogorov [1933] zero-one law states that every measurable tail set has measure 0 or 1. We now prove a resource-bounded generalization of this law. We first need a lemma.

Lemma 5.14. If X is a tail set which has a Δ-cover of total measure < 1, then X has Δ-covers of arbitrarily small total measure.

Proof. Let (h,m) be a Δ-cover of the tail set X with $\mu^*(h) = r < 1$. Fix k such that $r^2 + 2^{-k} < r^{3/2}$ and let n be the maximum of all $|h(0^i)|$ for $0 \leq i < \hat{m}$, where $\hat{m} = |m(0^k)|$. Let w_0 , w_1 , ..., w_{s-1} be a list of distinct strings of length n such that $\cup\{B_{w_i} | 0 \leq i < s\} = \cup\{B_{h(0^i)} | 0 \leq i < \hat{m}\}$. Define from h a real-valued density function d exactly as in the proof of Lemma 5.8. Note that $d(w_i) \geq 1$ for $0 \leq i < s$. Fix a string u of length n such that $d(u) \leq r$. (This u exists because $d(\lambda) = r$ and each $d(x) = \frac{1}{2}[d(x0) + d(x1)]$.)

Now modify the list $h(\lambda)$, $h(0)$, $h(0^2)$, ... as follows.

(i) Delete the first $\hat{m}$ entries.

(ii) In each place where there is an entry of the form uv, insert the entries $w_0 v$, ..., $w_{s-1} v$ immediately after.

Since k, $\hat{m}$, n, s, the list w_0, ..., w_{s-1} , and u are all constants, there is an enumerator $h' \in \Delta$ such that the resulting list is exactly $h'(\lambda)$, $h'(0)$, $h'(0^2)$, ... Also, the function $m' \in \Delta$ defined by $m'(x) = m(x0^{s+1})^{s+1}$ satisfies $|m'(0^i)| \leq (s + 1)|m(0^{i+s+1})|$, hence is clearly a modulus for h', i.e., (h' , m') is a Δ-cover. In fact, since X is a tail set, (h,m) is a Δ-cover of X, and h' replicates on each $X \cap B_{h(0^i)}$,$0 \leq i < \hat{m}$, the cover of $X \cap B_u$ by h,

(h', m') is a Δ-cover of X. This new Δ-cover of X has total measure

$$u^*(h') = 2^{-n}\, sd(u) + \sum_{i=\hat{m}}^{\infty} \mu\left(B_{h(0^i)}\right)$$

$$\leq \mu^*(h)d(u) + 2^{-k}$$

$$\leq r^2 + 2^{-k} < r^{3/2}.$$

We have now shown that if $r < 1$ and X has a Δ-cover of total measure r, then it has a Δ-cover of total measure $< r^{3/2}$. Since the sequence $r_0 = r$, $r_{n+1} = r_n^{3/2}$ converges to 0, this proves the lemma. □

We now prove the zero-one law for resource-bounded measure.

Theorem 5.15.

1) If X is a Δ-measurable tail set, then $\mu_\Delta(X) = 0$ or $\mu_\Delta(X) = 1$.

2) If X is a tail set which is measurable in $R(\Delta)$, then $\mu(X|R(\Delta)) = 0$ or $\mu(X|R(\Delta)) = 1$.

Proof. Assertion 1 follows immediately from assertion 2 by part 1 of Lemma 5.11.

To prove 2, let X be a tail set which is measurable in $R(\Delta)$ and assume that $\mu(X|R(\Delta)) < 1$. Let $\epsilon > 0$ be arbitrary. By Definition 5.10, $X \cap R(\Delta)$ has a Δ-cover of total measure < 1, whence by Lemma 5.14, it has a Δ-cover (h,m) with $\mu^*(h) < \epsilon$. By part 2 of Lemma 5.12, then, $\mu(X|R(\Delta)) < \epsilon$. Thus $\mu(X|R(\Delta)) = 0$. □

As we have noted, most sets of interest in complexity theory are tail sets. By Theorem 5.15, every measurable tail set in $R(\Delta)$ has measure 0 or 1 in $R(\Delta)$, so sets of intermediate measure are of very little complexity-theoretic interest. We should emphasize, however, that for resource-bounded notions Δ, measurability in $R(\Delta)$ is a very strong hypothesis. Thus we do not

interpret Theorem 5.15 to mean that sets of interest necessarily have measure 0 or 1. The third possibility, non-measurability in $R(\Delta)$, must be considered.

In this section we have presented a mere beginning of resource-bounded measure theory. We have not selected an axiomatization, we have restricted attention to measures induced by the usual Lebesgue measure on $\mathcal{P}(\{0,1\}^*)$, and we have omitted even the most basic properties of measurable sets (e.g., they form a "Δ-algebra," the measure is monotone and "Δ-additive," etc.). The only theorems we have proven are 5.9, the nontriviality of the measure, and 5.15, the resource-bounded zero-one law. Nevertheless, we have enough to begin applying the theory to the structure of exponential complexity classes.

6. Resource-Bounded Kolmogorov Complexity

Theorem 1.2 says that some languages in ESPACE have high space-bounded Kolmogorov complexity. In this section we prove that, with respect to both category and measure, nearly all languages in ESPACE have this property. We then prove an analogous but weaker result for exponential time complexity classes.

Theorem 6.1. For any $c > 0$ and $b < 1$, the set of all languages L such that $KS[2^{cn}](L_{\leq n}) < b \cdot 2^{n+1}$ a.e. is pspace-meager and has pspace-measure 0.

Proof. Let X be the set of all such L, where we assume without loss of generality that c is a positive integer and b is a binary rational between 0 and 1.

To see that X is pspace-meager, it suffices by Theorem 4.3 to exhibit a winning strategy β for player II in the game G[X;all, pspace]. Let β be defined as follows.

```
begin
    input x;
    n := the least integer such that (1 − b)2^{n+1} ≥ |x| + 1;
    while |x| < 2^{n+1} − 1 do
    begin // decide s_{|x|} //
        VOTE;
        if yes < (1/2)(total)
            then x := x1
            else x := x0
    end while;
    output x
end β.
```

The macro VOTE operates as follows.

```
begin
    yes, total, π := 0, 0, λ;
    while |π| < b·2^(n+1) do
    begin
        if OK (π,n,x) then
        begin //π gets to vote //
            total := total + 1;
            if U(<π,s_|x|>) outputs 1 in ≤ 2^cn space
                then yes := yes + 1
        end if;
        π := next (π)
    end while
end VOTE.
```

The predicate OK(π,n,x) here asserts that $|x| < 2^{n+1} - 1$, $U(<\pi,s_i>)$ halts in $\leq 2^{cn}$ space for each $0 \leq i \leq |x|$, and $U(<\pi,s_i>)$ outputs the i^{th} bit of x for each $0 \leq i < |x|$. It is clear that this condition can be tested in space polynomial in $2^n + |x| + |\pi|$, whence it follows easily that $\beta \in$ pspace.

Now fix one of the values of n computed by player II during a play (α,β) of the game, where α is an arbitrary strategy for player I. For each $|x| \leq j \leq 2^{n+1} - 1$, let total(j) be the final value of total computed by VOTE during the while-loop cycle in which β decides s_j . (Here $|x|$ denotes the length of the original input to β and we insist that total $(2^{n+1} - 1)$ be defined even though the corresponding cycle of the while-loop is not actually executed.) Then β ensures that $0 \leq$ total $(|x|) < 2^{b\cdot 2^{n+1}}$ and $0 \leq$

total $(j+1) \leq \frac{1}{2}$ total (j) for $|x| \leq j < 2^{n+1} - 1$. It follows from these and the choice of n that $0 \leq$ total $(2^{n+1} - 1) < 1$, whence total $(2^{n+1} - 1) = 0$. But this implies that $KS[2^{cn}](R(\alpha,\beta)_{\leq n}) \geq b \cdot 2^{n+1}$. Since β establishes this condition for a different value of n during each of player II's turns, it follows that $R(\alpha,\beta) \notin X$. Thus β wins G[X;all, pspace] for player II, so X is pspace-meager.

We now turn to the proof that X has pspace-measure 0. For each $j \in \mathbb{N}$, let X_j be the set of all languages L such that $KS[2^{cn}](L_{\leq n}) < b \cdot 2^{n+1}$ holds for all $n \geq \log j$. Clearly, $X = \cup\{X_j | j \in \mathbb{N}\}$. By Lemma 5.7, it suffices to show that this union is in fact a pspace-union of pspace-measure 0 sets.

Fix $j, k \in \mathbb{N}$ for a moment and choose the least $n \in \mathbb{N}$ such that $(1 - b)2^{n+1} \geq j + k + 2$. Let $\pi^{(0)}, \ldots, \pi^{(N-1)}$ be the lexicographic enumeration of all programs π of length $< b \cdot 2^{n+1}$ such that $U(<\pi,x>)$ halts in $\leq 2^{cn}$ space for all $|x| \leq n$. Then it is easily checked that there is an enumerator $h_{jk} \in$ pspace such that

$$h_{jk}(0^{\ell}) = \begin{cases} U(<\pi^{(\ell)},s_i>)_{i=0}^{2^{n+1}-2} & \text{if } \ell < N \\ 0^{k+\ell+2} & \text{if } \ell \geq N \end{cases}$$

holds for each $\ell \in \mathbb{N}$. In fact, since 2^n is linear in $j + k$, there is a function $h \in$ pspace such that this holds for all $j, k, \ell \in \mathbb{N}$. Similarly, there is a function $m \in$ pspace such that $m_{jk}(0^{\ell}) = 0^{N+\ell}$ for all $j, k, \ell \in \mathbb{N}$. It is now routine to check that each (h_{jk}, m_{jk}) is a pspace-cover of X_j with total measure $\mu^*(h_{jk}) \leq 2^{-k}$. It follows that each (h_j, m_j) is a pspace-null cover of X_j, whence (h,m) testifies that X is a pspace-union of the pspace-measure 0 sets X_j. □

Corollary 6.2. For any $c > 0$ and $b < 1$, the set of languages L with

$KS[2^{cn}](L_{\leq n}) \geq b \cdot 2^{n+1}$ i.o. is comeager and has measure 1 in ESPACE. □

If the game strategy β used in the first part of the proof of Theorem 6.1 is played against itself, then the result $R(\beta,\beta)$ is essentially the language constructed by Huynh [1986b] in his proof of Theorem 1.2. In this sense, Theorem 1.2 is an immediate corollary of Theorem 6.1.

A simple modification of Theorem 6.1 and its proof gives the following result, which will be useful in section 7.

Corollary 6.3. For any $c > 0$ and $b < 1$, the set of all languages L such that $KS[2^{cn}](L_{=n}) < b \cdot 2^n$ a.e. is pspace-meager and has pspace-measure 0. □

The situation in exponential time complexity classes is not as well understood as the situation in ESPACE. It is reasonable to conjecture that E contains languages whose KT-complexities are superpolynomial, but this implies $E \not\subseteq P/Poly$, which is a major open problem of complexity theory and cannot be proven by relativizable methods.

Here we prove a weaker analogue of Theorem 6.1. In order to formulate this result, we use the G-hierarchy of section 2 to define the following hierarchy of time-bounded Kolmogorov complexity classes.

Definition 6.4. For each $i \geq 1$,

$$KE_i = \{L \mid KT[2^{G_{i-1}}](L_{\leq n}) \in G_{i-1}\}.$$

Each KE_i is an uncountable nonuniform complexity class. Nevertheless, these classes have the following useful properties.

Lemma 6.5. For each $i \geq 1$, $E_i \subseteq KE_i \subseteq KE_{i+1}$ and KE_{i+1} is closed under $\leq_T^{P_i}$-reductions and G_i closeness.

Proof. It is obvious that $KE_i \subseteq KE_{i+1}$. If $L \in E_i$, then $KT[2^{G_{i-1}}]$ is

bounded, so it is also clear that $E_i \subseteq KE_i$.

Assume $L' \leq_T^{P_i} L \in KE_{i+1}$. Then there exist a G_i-time-bounded oracle machine M and a sequence $\pi^{(0)}$, $\pi^{(1)}$,... of programs such that $L' = L(M^L)$, $|\pi^{(n)}| \in G_i$, and $U(<\pi^{(n)}, x>)$ decides whether $x \in L$ in $G_i(n)$ time for all $x \in \{0,1\}^{\leq n}$ and $n \in \mathbb{N}$. For each n, then, consider a program $\hat{\pi}^{(n)}$ which simulates M, using $\pi^{(t(n))}$ to answer oracle queries, where $t \in G_i$ is a bound on the running time of M. It is easily checked that the programs $\hat{\pi}^{(n)}$ testify that $L' \in KE_{i+1}$. This proves that KE_{i+1} is closed under $\leq_T^{P_i}$-reductions.

Finally, assume L' is G_i close to KE_{i+1} , i.e., that $|(L' \Delta L)_{\leq n}| \in G_i$ for some $L \in KE_{i+1}$. Then, since $i \geq 1$, each $(L' \Delta L)_{\leq n}$ has a listing whose length is G_i as a function n. These listings can then be combined with programs testifying that $L \in KE_{i+1}$ to get programs testifying that $L' \in KE_{i+1}$. Thus KE_{i+1} is also closed under G_i closeness. □

We now prove a time analogue of Theorem 6.1. This says that most languages in E_{i+1} have high KT-complexity in the sense that they are not in KE_i .

Theorem 6.6. For $i \geq 1$, KE_i is p_{i+1}-meager and has p_{i+1}-measure 0.

Proof. Let X be the set of all languages L such that $KT[2^{\hat{g}_i(n)}](L_{\leq n}) < \hat{g}_i(n)$ a.e. Then $KE_i \subseteq X$, so it suffices to show that X is p_{i+1}-meager and has p_{i+1}-measure 0.

To see that X is p_{i+1}-meager, modify the strategy β used in the proof of Theorem 6.1 in the following ways.

(i) In the assignment to n, replace $(1 - b)2^{n+1}$ with 2^n .

(ii) In VOTE, replace $b \cdot 2^{n+1}$ with $\hat{g}_i(n)$ and replace 2^{cn} space with $2^{\hat{g}_i(n)}$ time.

Since $2^{\hat{g}_i(n)} = \hat{g}_{i+1}(2^n)$ and 2^n is linear in $|x|$, it is easy to check that the modified strategy β runs in G_{i+1} time, i.e., that $\beta \in p_{i+1}$. Also, in a play (α,β) of the game, player II establishes the condition $0 \leq \text{total}\,(2^{n+1}-1) < 2^{\hat{g}_i(n)}-2^n$ for a different value of n during each of his turns. Since $\hat{g}_i(n) = o(2^n)$, it follows that $\text{total}\,(2^{n+1}-1) = 0$ for all sufficiently large such n, whence $R(\alpha,\beta) \notin X$. That is, β wins $G[X;\text{all},p_{i+1}]$ for player II, so X is p_{i+1}-meager by Theorem 4.3.

Now for each $j \in \mathbb{N}$, let X_j be the set of languages L such that $KT[2^{\hat{g}_i(n)}](L_{\leq n}) < \hat{g}_i(n)$ for all $n \geq \log j$. We will show that X is a p_{i+1}-union of the p_{i+1}-measure 0 sets X_j.

Since $2^{\hat{g}_i(\log k)} \geq \hat{g}_i(\hat{g}_i(\log k)) + k + 2$ holds for all but finitely many k, there is a finite modification g of $\hat{g}_i$ such that $g(k) \geq k$ and $2^n \geq \hat{g}_i(n) + k + 2$ hold for all k and all $n \geq g(\log k)$.

Now, given j and k, we let $n = g(\log(j+k))$. This is easily computed and 2^n is G_{i+1} as a function of $j + k$.

There are fewer than $2^{\hat{g}_i(n)} = \hat{g}_{i+1}(2^n)$ programs π of length $< \hat{g}_i(n)$. The total time to run $U(\langle\pi,x\rangle)$ for $2^{\hat{g}_i(n)}$ steps for each such π and each $x \in \{0,1\}^{\leq n}$ is thus G_{i+1} as a function of 2^n, hence as a function of $j + k$. Using this simulation, we can now imitate the second half of the proof of Theorem 6.1 to get $(h,m) \in p^2_{i+1}$ such that each (h_{jk}, m_{jk}) is a p_{i+1}-cover of X_j with total measure $\mu^*(h_{jk}) \leq 2^{-k}$. This shows that X is a p_{i+1}-union of the p_{i+1}-measure 0 sets X_j, whence X has p_{i+1}-measure 0 by Lemma 5.7. □

Corollary 6.7. For $i \geq 1$, KE_i is meager and has measure 0 in E_{i+1}. □

The case $i = 2$ here says that most languages in E_3 have superpolynomial time-bounded Kolmogorov complexity.

7. Small Circuits in Exponential Classes

Theorem 1.1 separates ESPACE from P/Poly $\cap$ ESPACE. In this section we widen this separation by proving that P/Poly is meager and has measure 0 in ESPACE. We then examine circuit-size complexity in exponential time complexity classes and prove, among other things, that P/Poly is meager and has measure 0 in E_3.

This investigation of small circuits in exponential complexity classes was our original motivation for the development of resource-bounded category and measure. Consequently, the main results of this section were originally proven directly. Here, however, we use Lemmas 7.1 and 7.5, which express a well-understood relationship between Kolmogorov complexity and circuit-size complexity, to easily derive the present results from those of section 6. For both these lemmas, we fix a one-to-one *coding scheme*

$$\#:\{\text{circuits}\} \to \{0,1\}^*$$

and a constant $k_\# \in \mathbb{N}$ such that

(i) given $w,y \in \{0,1\}^*$, a deterministic TM can compute in polynomial time whether y is the code of a circuit c with $|w|$ inputs and, if so, the output of c on input w;

(ii) $|\#(c)| < k_\#$ size (c) log (n + size (c)).

Lemma 7.1. If f is a nontrivial circuit-size bound, then every $L \in SIZE(f)$ has $KS[2^n](L_{=n}) < 2^{n-1}$ a.e.

Proof. Let f be such a bound and let

$$h(n) = k_\# g(n) \log (n + g(n)),$$

where g is a nontrivial circuit-size bound chosen so that $f = o(g)$. Note that $h(n) = o(2^n)$.

Now assume $L \in SIZE(f)$ and for each $n \in \mathbb{N}$, let c_n be a minimum-size circuit computing $L_{=n}$. Then for each n, we can combine a circuit-simulating machine M with the circuit code $\#(c_n)$ to get a program $\pi^{(n)}$ such that the following conditions hold for almost every n.

(i) $|\pi^{(n)}| < h(n) < 2^{n-1}$.

(ii) For each $x \in \{0,1\}^n$, $U(<\pi^{(n)}, x>)$ correctly decides whether $x \in L$ in $\leq 2^n$ space.

That is, the programs $\pi^{(n)}$ testify that $KS[2^n](L_{=n}) < 2^{n-1}$ a.e. □

From Lemma 7.1 and Corollary 6.3, we immediately get the following.

Theorem 7.2. If f is any nontrivial circuit-size bound, then SIZE(f) is pspace-meager and has pspace-measure 0. □

(The category portion of this was proven directly by a voting argument in Lutz [1987].)

Corollary 7.3. If f is any nontrivial circuit-size bound, then SIZE(f) is meager and has measure 0 in ESPACE. □

Corollary 7.4. P/Poly is meager and has measure 0 in ESPACE. □

We now turn to the matter of small circuits in exponential time complexity classes. Here it is convenient to use the KE-hierarchy introduced in section 6.

Lemma 7.5. If $i \geq 1$, then $SIZE(G_i) \subseteq KE_{i+1}$.

Proof. Let f be an arbitrary circuit-size bound in G_i. Since $i \geq 1$, we can choose $g \in G_i$ such that $f = o(g)$. If we then define h from g as in the proof of Lemma 7.1, we will have $h \in G_i$ also.

Now assume $L \in SIZE(f)$. Then an easy modification of the argument

used in the proof of Lemma 7.1 shows that $KT[G_i](L_{\leq n}) \in G_i$, whence $L \in KE_{i+1}$. Thus $SIZE(f) \subseteq KE_{i+1}$. □

From Lemma 7.5 and Theorem 6.6, the following theorem is immediate.

Theorem 7.6. If $i \geq 1$, then $SIZE(G_i)$ is p_{i+2}-meager and has p_{i+2}-measure 0. □

Corollary 7.7. P/Poly is meager and has measure 0 in E_3. □

If we fix a particular circuit-size bound $f \in G_i$ ($i \geq 1$) and a language $L \in SIZE(f)$, then the proof of Lemma 7.5 gives us a function $g \in G_i$ such that $KT[g(n)](L_{\leq n}) < g(n)$ a.e. It follows by the proof of Theorem 6.6 (with $\hat{g}_i$ replaced by g) that SIZE(f) is p_{i+1}-meager. This argument gives us the following.

Corollary 7.8. If $i \geq 1$ and $f \in G_i$, then SIZE(f) is p_{i+1}-meager and has p_{i+1}-measure 0. □

Corollary 7.9. For each $k \in \mathbb{N}$, $SIZE(n^k)$ is meager and has measure 0 in E_2. □

Since Wilson [1985] exhibits oracles under which $E_2 \subseteq P/Poly$ and $E \subseteq SIZE(n)$, Corollaries 7.7 and 7.9 take us about as far as we can go with relativizable techniques.

8. Information Accessible by Reducibilities

As mentioned in section 1, most intractability proofs for specific problems have taken the same form. Here we describe the more general *reducibility method*, which includes this form as a special case but may also lead to new lower bound arguments.

The reducibility method can be stated simply and informally as follows. Given a language L, let $\mathcal{R}(L)$ be the set of all languages which are efficiently reducible to L. Then the size of $\mathcal{R}(L)$, which is a measure of the amount of $\mathcal{R}$-accessible information in L, provides a lower bound for the complexity of computing L.

In applications of this method which are taken here as paradigmatic, the size of $\mathcal{R}(L)$ is simply the matter of whether or not $\mathcal{R}(L)$ contains a particular complexity class $\mathcal{C}$, i.e., whether or not L is $\mathcal{R}$-hard for $\mathcal{C}$. To date, most uses of the reducibility method have followed this paradigm.

The recently proven Theorems 1.3 and 1.4 show that the paradigmatic reducibility method also gives lower bounds for "approximate recognition" of languages.

The primary weakness of the paradigm is its extremely primitive interpretation of the "size" of $\mathcal{R}(L)$. Unless L is $\mathcal{R}$-hard for $\mathcal{C}$, i.e., contains all information about $\mathcal{C}$ in $\mathcal{R}$-accessible form, the paradigm deems $\mathcal{R}(L)$ to be small and offers no nontrivial lower bound. Since most interesting intractable problems are probably not hard for classes now known to contain intractability, this limitation is a severe one. It means, for example, that Theorems 1.3 and 1.4 are not likely to be applicable to interesting problems.

The remedy we propose is to use resource-bounded category and

measure to refine this primitive notion of size. If we do this, then the reducibility method, as stated above, gives a quantitative relationship between the $\mathcal{R}$-accessible information content of L and the computational complexity of L.

The specific reducibilities of interest here are $\leq_m^P$, $\leq_T^P$, $\leq_T^{PSPACE}$, and $\leq_T^{P_i}$ $(i \geq 1)$. It is thus convenient to define the set $P_m(L) = \{L' \mid L' \leq_m^P L\}$, and to define the sets $P_T(L)$, $PSPACE_T(L)$, and $P_{iT}(L)$ similarly from the other reducibilities.

Under the hypothesis that L is 2^{n^c} close to P for every $c > 0$, Theorems 1.3 and 1.4 tell us that $E \not\subseteq P_m(L)$ and $ESPACE \not\subseteq P_T(L)$, respectively. In sections 9 and 10 we will show that this hypothesis in fact implies that $P_m(L)$ is meager in E and that $PSPACE_T(L)$ is meager and has measure 0 in ESPACE. That is, we replace conclusions of the form $\mathcal{C} \not\subseteq \mathcal{R}(L)$ with conclusions asserting that $\mathcal{R}(L) \cap \mathcal{C}$ is a very small subset of $\mathcal{C}$. Put differently, we replace conclusions stating that L does not contain all information about $\mathcal{C}$ in $\mathcal{R}$-accessible form with conclusions stating that L contains very little $\mathcal{R}$-accessible information about $\mathcal{C}$. Although these new conclusions appear to be considerably stronger, this has not been proven. We thus formulate the following hypotheses.

Definition 8.1. If $\mathcal{C}$ is a complexity class and $\mathcal{R}$ is a reducibility, then the *category partial information hypothesis* for $\mathcal{C}$ and $\mathcal{R}$ is the assertion $PIH_{category}(\mathcal{C},\mathcal{R})$ which says that there is a language $L \in \mathcal{C}$ such that $\mathcal{R}(L)$ does not contain $\mathcal{C}$ and $\mathcal{R}(L)$ is not meager in $\mathcal{C}$. The *measure partial information hypothesis* $PIH_{measure}(\mathcal{C},\mathcal{R})$ is defined similarly, except that "meager" is replaced by "measure 0."

Thus partial information hypotheses assert the existence of languages

containing "substantial but incomplete" information about $\mathcal{C}$ in $\mathcal{R}$-accessible form.

We make the following three conjectures.

Conjecture 8.2. $\mathrm{PIH}_{\mathrm{category}}(E, \leq_m^P)$ holds.

Conjecture 8.3. $\mathrm{PIH}_{\mathrm{category}}(\mathrm{ESPACE}, \leq_T^{\mathrm{PSPACE}})$ holds.

Conjecture 8.4. $\mathrm{PIH}_{\mathrm{measure}}(\mathrm{ESPACE}, \leq_T^{\mathrm{PSPACE}})$ holds.

If any of these conjectures hold, then the results of the following two sections do indeed increase the power of the reducibility method.

9. Information Accessible in Polynomial Space

Theorem 1.4 says that, if L is 2^{n^c} close to P for every $c > o$, then $P_T(L)$ does not contain all of ESPACE. In this section we prove the stronger result that, if L is 2^{n^c} close to DSPACE(2^{n^c}) for every $c > 0$, then $PSPACE_T(L)$ is meager and has measure 0 in ESPACE. That is, a language which is approximable in feasible space does not contain significant polynomial-space-accessible information about ESPACE.

The key to Huynh's proof of Theorem 1.4 is Theorem 1.2, the existence of ESPACE languages with high space-bounded Kolmogorov complexity. Theorem 6.1, which says that most languages in ESPACE have this property, plays an analogous role in this section.

We first prove that almost all languages in ESPACE are very hard to approximate.

Theorem 9.1. If $c > 0$ and $b > 1$, then the set of languages which are $\frac{2^{n+1}}{bn}$ far from DSPACE(2^{cn}) is pspace-comeager and has pspace-measure 1.

Proof. Fix such c and b and suppose L is in the complement of this set, i.e., that there is an $O(2^{cn})$ space-bounded machine M such that $|(L \triangle L(M))_{\leq n}| < \frac{2^{n+1}}{bn}$ a.e. Fix $0 < c' < c$ and $\frac{1}{b} < b' < 1$. Then for each n we can combine a description of M with a listing of $(L \triangle L(M))_{\leq n}$ to get a program $\pi^{(n)}$ such that the following conditions hold for almost all n.

(i) $|\pi^{(n)}| < b' \cdot 2^{n+1}$

(ii) $U\left[\left\langle \pi^{(n)}, s_i \right\rangle\right]_{i=0}^{2^n-2} = L_{\leq n}$ in $\leq 2^{c'n}$ space.

That is, the programs $\pi^{(n)}$ testify that $KS[2^{c'n}](L_{\leq n}) < b' \cdot 2^{n+1}$ a.e. By Theorem 6.1, the set of all L with this property is pspace-meager and has pspace-measure 0. □

Corollary 9.2. If $c > 0$ and $b > 1$, then the set of languages which are $\frac{2^{n+1}}{bn}$ far from DSPASCE(2^{cn}) is comeager and has measure 1 in ESPACE. □

We now give our improvement of Theorem 1.4.

Theorem 9.3. If L is 2^{n^c} close to DSPACE(2^{n^c}) for every $c > 0$, then $PSPACE_T(L)$ is pspace-meager and has pspace-measure 0.

Proof. Let X be the set of languages L such that $KS[2^{n^c}](L_{\leq n}) < 2^{n^c}$ holds a.e. for every $c > 0$. If the hypothesis holds, then an argument like that in the proof of Theorem 9.1 shows that $L \in X$. Since Theorem 6.1 says that X is pspace-meager and has pspace-measure 0, it thus suffices to prove that X is closed under $\leq_T^{PSPACE}$.

Assume $L' \leq_T^{PSPACE} L \in X$. Fix a q(n)-space-bounded oracle machine M such that $L' = L(M^L)$, where q is a polynomial. Also, for each $c > 0$ and $n \in \mathbb{N}$, fix a program $\pi(c,n)$ testifying to the value of $KS[2^{n^c}](L_{\leq n})$.

Now for each c and n, consider a program $\pi'(c,n)$ which simulates M, using $\pi(c,q(n))$ to answer oracle queries. Then there is a constant $d > 0$, not depending on c or n, such that for almost all c, for almost all n, the following conditions hold.

(i) $|\pi'(c,n)| \leq 2^{q(n)^c} + d$.

(ii) For all $x \in \{0,1\}^{\leq n}$, $U(\langle\pi'(c,n),x\rangle)$ decides whether $x \in L'$ in $\leq q(n) + 2^{q(n)^c} + d$ space.

Now let $c_1 > 0$ be arbitrary and choose $c > 0$ such that $q(n) + 2^{q(n)^c} < 2^{n^{c_1}}$ a.e. Then the programs $\pi'(c,n)$ testify that $KS[2^{n^{c_1}}](L'_{\leq n}) < 2^{n^{c_1}}$ a.e. Thus $L' \in X$, whence X is indeed closed under $\leq_T^{PSPACE}$ and the proof is complete. □

Corollary 9.4. If L is 2^{n^c} close to DSPACE(2^{n^c}) for every $c > 0$, then $PSPACE_T(L)$ is meager and has measure 0 in ESPACE. □

Thus, if a language can be shown to contain significant polynomial-space-accessible information about ESPACE, it will follow that the language is not very close to PSPACE.

10. Information Accessible in Polynomial Time

If Conjecture 8.3 or Conjecture 8.4 holds, then section 9 already extends the class of languages which can be proven intractable by the reducibility method. However, any language which is susceptible to the methods of section 9 still must lie "well outside" of PSPACE. Since most languages that we would like to prove intractable are elements of PSPACE, it follows that we need a finer method, i.e., a method which applies to a finer reducibility.

Theorem 1.3 says that, if L is 2^{n^c} close to P for every $c > 0$, then $P_m(L)$ does not contain all of E. In this section we show that, if L is 2^{n^c} close to DTIME(2^{n^c}) for every $c > 0$, then $P_m(L)$ is actually meager in E. That is, a language which is approximable in feasible time contains only meager $\leq_m^P$-accessible information about E. If Conjecture 8.2 holds, this strengthens Theorem 1.3.

The basis of section 9 is Theorem 6.1, which says that most ESPACE languages are incompressible in a space-bounded algorithmic sense. Similarly, the present section is based on Theorem 10.2 below, a technical result which says, in part, that most languages in E are incompressible in a time-bounded, many-one sense. The following definition, which specifies this sense, uses the function $m_g(n) = |\{x \in \{0,1\}^{\leq n} \mid \exists y \in \{0,1\}^{\leq n}[x \neq y \text{ and } g(x) = g(y)]\}|$ to quantify the rate at which a function $g:\{0,1\}^* \to \{0,1\}^*$ fails to be one-to-one.

Definition 10.1.

1) A language L is f(n)-*incompressible* by $\leq_m^P$-reductions if every $\leq_m^P$-reduction g of L has $m_g(n) \leq f(n)$ for infinitely many n. If this holds for some constant function f(n), then L is *strongly incompressible* by $\leq_m^P$-reductions.

2) A language L is *simultaneously* f(n)-*incompressible* by $\leq_m^P$-reductions *and* h(n) *far* from a set X of languages if for each $\leq_m^P$-reduction g of L and each language $L' \in X$ there are infinitely many n for which $m_g(n) \leq f(n)$ and $|(L \Delta L')_{\leq n}| \geq h(n)$ both hold.

Theorem 10.2. For any $\epsilon > 0$, any $a > 0$, and any nondecreasing, unbounded function f(n) which is computable in $2^{O(n)}$ time, the set of all languages which are simultaneously $2^{\epsilon n}$-incompressible by $\leq_m^P$-reductions and $\frac{2^{n+1}}{f(n)}$ far from DTIME(2^{an}) is p-comeager.

Proof. Let $X = X(\epsilon,a,f)$ be the set of all such languages, where we assume without loss of generality that $\epsilon \leq 1$ and $f(n) \leq 2^n$ for all n. Let $M_0, M_1, \ldots$, and $T_0, T_1, \ldots$ be standard enumerations of the Turing machine acceptors and transducers, respectively.

If T_k is a transducer and $x,y \in \{0,1\}^*$, we say that $T_k(x) = T_k(y)$ *in time* t if $T_k(x) = T_k(y)$ and T_k halts in $\leq t$ steps on each of the inputs x and y. We then say that x *defies* T_k if there exist $i < j < |x|$ such that the i^{th} and j^{th} bits of x are different but $T_k(s_i) = T_k(s_j)$ in time $|x|$.

If x defies T_k and $x \sqsubseteq L$ it is clear that T_k is not a many-one reduction of L. It is also clear that the predicate "x defies T_k" can be evaluated in time polynomial in $|x|$.

Now let $\Theta(k,i,j,x,n)$ be a predicate asserting that $k < |x|$, x does not defy T_k, $i < j$, $|x| \leq j < 2^{n+1} - 1$, and $T_k(s_i) = T_k(s_j)$ in time 2^{n+1}. Note that the condition $(\exists k,i,j)\ \Theta(k,i,j,x,n)$ can be tested in time polynomial in $|x| + 2^n$. Consider the strategy $\beta = \beta(\epsilon,a,f)$ defined as follows.

begin

input x;

$z, n, \ell := x, \lceil \epsilon^{-1} \log(2 + |x|) \rceil, 0;$

if $(\exists k, i, j)\ \Theta(k, i, j, x, n)$

then fix such i, j with k minimum

else $i, j := 0, 1;$

while $|z| < 2^{n+1} - 1$ do

cases

$|z| = i$: $z := z0$

[] $|z| = j$: $z := zb$, where b is the negation of the i^{th} bit of z

[] else: if M_ℓ accepts $s_{|z|}$ in $\leq 2^{(a+1)n}$ time

then $z, \ell := z0, (\ell+1) \bmod \lfloor \frac{f(n)}{4} \rfloor$

else $z, \ell := z1, (\ell+1) \bmod \lfloor \frac{f(n)}{4} \rfloor$

end cases and while;

output z

end β.

Since 2^n is polynomial in $|x|$ here, it is easy to check that $\beta \in p$. In fact, we will show here that β is a winning strategy for player II in $G[X^c; \text{all}, p]$, where $X^c = \mathcal{P}(\{0,1\}^*) \backslash X(\epsilon, a, f)$. To this end, let $L = R(\alpha, \beta)$, where α is an arbitrary strategy for player I. It suffices to show that $L \in X$.

Fix a $\leq_m^P$-reduction g of L and a language $L' \in \text{DTIME}(2^{an})$, with witnesses T_k and M_ℓ, respectively.

Since $g = T_k$ is a reduction of L, no initial segment of L defies T_k. This implies that the first if-test in β is not true with k as the least witness during any move by player II. Since T_k runs in polynomial time, this in turn

implies that, for all but finitely many of player II's moves, $m_g(n) \leq |x| < 2^{\epsilon n}$.

The machine M_ℓ runs in $O(2^{an})$ time on any input in $\{0,1\}^{\leq n}$, so the while-loop in β ensures that, for all but finitely many of player II's moves, $|(L \triangle L')_{\leq n}| \geq \lfloor (2^{n+1} - |x| - 3)/\lfloor \frac{f(n)}{4} \rfloor \rfloor \geq \frac{2^{n+3} - 4|x| - 12}{f(n)} - 1 \geq \frac{2^{n+2} - 12}{f(n)} \geq \frac{2^{n+1}}{f(n)}$. Since g and L' are arbitrary here, we have now shown that $L \in X$. Thus β does indeed win $G[X^c; \text{all}, p]$ for player II, so Theorem 4.3 tells us that X is p-comeager. □

Corollary 10.3. For any $\epsilon > 0$, any $a > 0$, and any nondecreasing, unbounded function f(n) which is computable in $2^{O(n)}$ time, the set of all languages which are simultaneously $2^{\epsilon n}$-incompressible by $\leq_m^P$-reductions and $\frac{2^{n+1}}{f(n)}$ far from $DTIME(2^{an})$ is comeager in E. □

If the game strategy β used in the above proof is played against itself, where $\epsilon = c = 1$ and $f(n) = n$, then we get the following result, which is the basis of Schöning's proof of Theorem 1.3.

Corollary 10.4. There is a language $L \in E$ which is strongly incompressible by $\leq_m^P$-reductions and $\frac{2^n}{n}$ far from P. □

If we ignore the incompressibility in Theorem 10.2, then we immediately get the following.

Theorem 10.5. If $c > 0$ and f(n) is any nondecreasing, unbounded function which is computable in $2^{O(n)}$ time, then the set of languages which are $\frac{2^{n+1}}{f(n)}$ far from $DTIME(2^{cn})$ is comeager in E. □

Since f(n) may be an extremely slow-growing function, this is a very strong non-approximability theorem. It says that, in the sense of category, most languages in E cannot be feasibly approximated with an error rate that

converges to 0 in any feasible way.

We finally come to our improvement of Theorem 1.3.

Theorem 10.6. If L is 2^{n^c} close to DTIME(2^{n^c}) for every $c > 0$, then $P_m(L)$ is p-meager.

Proof. Assume that $P_m(L)$ is not p-meager and let X be the set of languages in Theorem 10.2, where $\epsilon = \frac{1}{2}$, $a = 1$, and $f(n) = n$. Since X is p-comeager, there is a language $A \in X \cap P_m(L)$. Fix such, let g be a $\leq_m^P$-reduction of A to L, let q be a polynomial such that $|g(x)| \leq q(|x|)$ for all x, and choose $0 < c < b$ such that $q(n)^b \leq n$ a.e. We will show that L is 2^{n^c} far from DTIME(2^{n^c}).

Let $L' \in$ DTIME(2^{n^c}). Then $g^{-1}(L') \in$ DTIME(2^n), so there exist infinitely many n such that $m_g(n) \leq 2^{\frac{n}{2}}$ and $|(A \Delta g^{-1}(L'))_{\leq n}| \geq \frac{2^{n+1}}{n}$. For any sufficiently large such n we thus have $|(L \Delta L')_{\leq q(n)}| \geq \frac{2^{n+1}}{n} - 2^{\frac{n}{2}} \geq \frac{2^n}{n} \geq 2^{n^{\frac{c}{b}}} \geq 2^{q(n)^c}$. Since $L' \in$ DTIME(2^{n^c}) is arbitrary here, it follows that L is 2^{n^c} far from DTIME(2^{n^c}). □

Corollary 10.7. If L is 2^{n^c} close to DTIME(2^{n^c}) for every $c > 0$, then $P_m(L)$ is meager in E.

Thus, if a language L can be shown to contain non-meager $\leq_m^P$-accessible information about E, it will follow that the language is not very close to P. In fact, the proofs of Theorems 10.2 and 10.6 show that this will follow if it is just shown that player II does not have a winning strategy for the game $G[P_m(L); \text{all}, p]$. □

It is not known whether Theorem 10.6 holds with $P_T(L)$ in place of $P_m(L)$ or with "measure 0" in place of "meager," but it is now easy to get the

following much weaker result.

Theorem 10.8. If $i \geq 1$ and L is G_i close to E_{i+1}, then $P_{iT}(L)$ is p_{i+2}-meager and has p_{i+2}-measure 0.

Proof. By Lemma 6.5, the hypothesis implies that $P_{iT}(L) \subseteq KE_{i+1}$, so this follows immediately from Theorem 6.6. □

Corollary 10.9. If L is polynomially close to E_2, then $P_T(L)$ is meager and has measure 0 in E_3. □

11. Conclusion

Resource-bounded category and measure have been introduced and shown to reveal new structure in many complexity classes. This structure has been used to refine known relationships between uniform and nonuniform complexity measures. It has also been used as the basis for a new formulation of the reducibility method.

The important open questions here concern the partial information hypotheses. If any of Conjectures 8.2, 8.3, or 8.4 hold, then the newly formulated reducibility method is indeed more powerful than the old one. Of course it would be ideal for these conjectures to be shown to hold with interesting, natural problems as witnesses, since then the work here gives lower bounds for such problems.

In any case, it is already clear that resource-bounded category and measure interact in interesting ways with resource-bounded reducibilities, nonuniform complexity measures, approximation, and other much-studied structural aspects of complexity classes. It is expected that the study of such interactions will continue to yield clarifying insights.

References

J. L. Balcázar, R. V. Book, and U. Schöning [1986], Sparse sets, lowness, and highness, *SIAM Journal on Computing* 15, pp. 739-747.

S A. Cook [1971], The complexity of theorem proving procedures, *Proceedings of the* 3^{rd} *ACM Symposium on the Theory of Computing*, pp. 151-158.

R. I. Friedzon [1972], Families of recursive predicates of measure zero, translated in *Journal of Soviet Mathematics* 6 (1976), No. 4, pp. 449-455.

P. R. Halmos [1950], *Measure Theory*, Springer-Verlag.

J. Hartmanis [1983], Generalized Kolmogorov complexity and the structure of feasible computations, *Proceedings of the* 24^{th} *IEEE Symposium on the Foundations of Computer Science*, pp. 439-445.

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

D. T. Huynh [1986a], Some observations about the randomness of hard problems, *SIAM Journal on Computing* 15, pp. 1101-1105.

D. T. Huynh [1986b], Resource-bounded Kolmogorov complexity of hard languages, *Structure in Complexity Theory*, Lecture Notes in Computer Science, Vol. 223, pp. 184-195.

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

R. M. Karp [1972], Reducibility among combinatorial problems, in R. E. Miller and J. W. Thatcher (eds.), *Complexity of Computer Computations*, Plenum Press, pp. 85-104.

R. M. Karp and R. J. Lipton [1980], Some connections between nonuniform and uniform complexity classes, *Proceedings of the* 12^{th} *ACM Symposium on the Theory of Computing*, pp. 302-309.

A. N. Kolmogorov [1933], *Foundations of the Theory of Probability*, English translation, Chelsea Publishing, 1950.

L. A. Levin [1973], Universal sorting problems, *Problems of Information Transmission* 9, 265-266.

L. R. Lisagor [1979], The Banach-Mazur game, translated in *Math. USSR Sbornik*, Vol. 38 (1981), No. 2, pp. 201-206.

J. H. Lutz [1987], Resource-bounded Baire category and small circuits in exponential space, *Proceedings of the Second Annual Structure in Complexity Theory Conference*, to appear.

K. Mehlhorn [1973], On the size of sets of computable functions, *Proceedings of the* 14th *IEEE Symposium on the Foundations of Computer Science*, pp. 190-196.

K. Mehlhorn [1974], The "almost all" theory of subrecursive degrees is decidable, *Proceedings of the Second Colloquium on Automata, Languages and Programming*, Springer Lecture Notes, pp. 317-325.

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

J. C. Oxtoby [1971], *Measure and Category*, Springer-Verlag.

U. Schöning [1983], A low and a high hierarchy within NP, *Journal of Computer and System Sciences* 27, pp. 14-28.

U. Schöning [1986], Complete sets and closeness to complexity classes, *Mathematical Systems Theory* 19, pp. 29-41.

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

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

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

Y. Yesha [1983], On certain polynomial-time truth-table reducibilities of complete sets to sparse sets, *SIAM Journal on Computing* 12, pp. 411-425.