Contributions to the Study of Resource-Bounded Measure

Elvira Mayordomo Cámara

Barcelona, abril de 1994

Contributions to the Study of Resource-Bounded Measure

Tesis doctoral presentada en el
Departament de Llenguatges i Sistemes Informàtics
de la Universitat Politècnica de Catalunya

para optar al grado de
Doctora en Ciencias (Matemáticas)

por
Elvira Mayordomo Cámara

Dirigida por el doctor
José Luis Balcázar Navarro

Barcelona, abril de 1994

This dissertation was presented on June 21st 1994
in the Departament de Llenguatges i Sistemes Informàtics,
Universitat Politècnica de Catalunya.

The committee was formed by the following members

Prof. Ronald V. Book, University of California at Santa Barbara
Prof. Ricard Gavaldà, Universitat Politècnica de Catalunya
Prof. Mario Rodríguez, Universidad Complutense de Madrid
Prof. Marta Sanz, Universitat Central de Barcelona
Prof. Jacobo Torán, Universitat Politècnica de Catalunya

Acknowledgements

I want to thank specially two people, José Luis Balcázar and Jack Lutz. Without their help and encouragement this dissertation would never have been written. José Luis Balcázar was my advisor and taught me most of what I know about Structural Complexity, showing me with his example how to do research. I visited Jack Lutz in Iowa State University in several occasions from 1991. His patience and enthusiasm in explaining to me the intuition behind his theory and his numerous ideas were the starting point of most of the results in this dissertation.

Several coauthors have collaborated in the publications included in the different chapters: Jack Lutz, Steven Fenner, Ronald Book, José Luis Balcázar, Monsterrat Hermo, and Harry Buhrman. In addition, a number of other people have contributed with different ideas, among others Peter Miltersen, Eric Allender, and Martin Strauss.

Thanks to all the people that helped me during my visits to Iowa State University, the University of Ulm, the University of California at Santa Barbara and the University of Aarhus. I specially appreciate the hospitality of Robyn Lutz and Celia Wrathall.

The Spanish Ministerio de Educacion y Ciencia supported this work through grant FPI-PN90 25434951. I received partial support from Acción Integrada HA-047, by the ESPRIT EC project 3075 (ALCOM) and by National Science Foundation Grant CCR-9157382.

The people in the Department de Llenguatges i Sistemes Informàtics have given me their friendship and help through these years. In the section of Theoretical Computer Science I found lots of interesting discussions and people that listened to my often still incoherent ideas.

Finally, I thank my family for being there. And Juanjo, for his love and support.

Contents

Chapter 1: Introduction and preliminaries

1.1 Introduction

The notion of "effective procedure" or algorithm was born in the early thirties, building on the work of Church, Gödel, Kleene, Post and Turing ([Chur33], [Chur36], [Göde], [Klee], [Post36], [Turi36], [Turi37]). They developed several formalizations of this concept, such as λ-calculus, partial recursive functions and the Turing Machine formalism. This was the base of Recursive Function Theory, where recursive problems were defined as those that are solvable by an algorithm.

The construction of actual computers led to the consideration of feasibly solvable problems instead of recursive or theoretically solvable ones. This distinction was related to the explosive growth of the exponential function, which implies that algorithms based on exhaustive search may be infeasible in practice. Therefore an increasing attention was paid in the sixties to the amount of computational resources used in the solution of a recursive problem. Specifically, the resources considered were mainly time and space.

With the work of Hartmanis, Stearns and Lewis ([HartSt], [LewiStH], [SteaHaL]) Complexity Theory started a division of recursive problems into complexity classes according to the amount of resources used in their resolution. The computing model used here was the Turing Machine, which corresponds to a simple mathematical representation of a computer (see [HopcUl] for a complete description). The problems that can be solved in time polynomial in the length of the input are considered feasibly solvable, and form the class denoted P. But there exist many problems for which no polynomial time algorithm is known; many important ones, among them, have the property of being easy to check, that is, once a solution is found, it can be checked in polynomial time that it is indeed a solution. This leads to the definition of the class NP as the class of "easy-to-check" problems; there are many important problems in this class, for instance those dealing with the satisfability of boolean formulas or with the existence of a hamiltonian path in a graph, and some practical operation research problems such as the distribution of crews into planes. It would be very interesting to know whether P and NP coincide. In the seventies, some techniques analogous to those in Recursive Function Theory, for instance the concept of "completeness" ([Cook], [Karp]), started to be developed and then used to attack the P versus NP problem. This constitutes the beginning of the field of Structural Complexity, which we develop next.

Structural Complexity describes complexity classes using various types of resources including time, space, nondeterministic time and space, Boolean circuit size and depth, and alternating time and space. We will not define here all the mentioned resources, let us just say that the word "nondeterministic" refers to the use of nondeterministic algorithms, that are a generalization of the usual algorithms with the extra possibility of choosing among several instructions that follow a given one; and that the word "alternating" is (indirectly) related to the use of parallel algorithms, in which several instructions can be run at the same time. The problems considered are mainly decisional ones, which are denoted as languages, and we say that an algorithm recognizes a language when it solves the corresponding decisional problem. The main open problems in Structural Complexity have the form 'Is the class of languages that can be recognized with an amount f of resource α included in the class of those recognized with an amount g of resource β?' The above mentioned P versus NP problem can be formulated as 'Is the class of languages that can be recognized with nondeterministic polynomial time included in the class of the polynomial time recognizable languages?' Other examples involve comparisons of polynomial time (P) versus polylogarithmic parallel time with polynomial size hardware (NC), exponential time (E) versus polynomial size circuits (P/poly), and polynomial space (PSPACE) versus polynomial time (P).

The notions of oracle Turing Machine, reduction and complete language are introduced in order to compare the complexity of specific languages. An oracle Turing Machine is an ordinary Turing Machine equipped with direct access to a particular language A, which is called oracle. The oracle Turing Machine operates as an ordinary one, with the extra possibility of, given a string q, computing in a single step the answer to '$q \in A$?'. For each oracle A, we can define complexity classes according to the resources used to recognize a language, when we can access oracle A. This means that, for each oracle A, we have a particular computation universe where the solution of A is given for free. A great effort was done to find out which answers to open problems of the form $\mathcal{C} \subseteq \mathcal{D}$? hold when translated into some such universe, trying to get some light on the solution of the open problem (see [BakeGiS] for the first work in this line). Given a problem such as $\mathcal{C} \subseteq \mathcal{D}$?, we say that it is nonrelativizable when there exist oracles A and B such that $\mathcal{C} \subseteq \mathcal{D}$ using oracle A and $\mathcal{C} \nsubseteq \mathcal{D}$ using oracle B, that is, the solution of the problem is different in the contexts of oracles A and B. A nonrelativizable problem is considered difficult because most of the techniques used in Structural Complexity are independent of the oracle used. (Just a few new results have shown that nonrelativizable problems are not impossible to solve; for instance, Shamir has shown in [Sham] that PSPACE $\subseteq$ IP, while there exist oracles for which the opposite holds [FortSi].)

If we can recognize a language A with an oracle B, this means that B is at least as hard to recognize as A, since an algorithm for B would produce an algorithm for A. This defines a partial preorder of languages, denoted $\leq_{\mathrm{T}}$ and called Turing reducibility, with the meaning that $A \leq_{\mathrm{T}} B$ if A can be recognized using B as oracle. Polynomial time reducibilities appear when considering only oracle Turing Machines that work in polynomial time. In general, given a restriction r on the oracle access, we say that a language L $\leq_r^{\mathrm{p}}$-reduces to a language A when L can be recognized in polynomial time using A as oracle with the access restrictions indicated by r. We say that a language A is $\leq_r^{\mathrm{p}}$-hard for a class $\mathcal{C}$ when every language in $\mathcal{C}$ $\leq_r^{\mathrm{p}}$-reduces to A, and that A is $\leq_r^{\mathrm{p}}$-complete for $\mathcal{C}$ when A is $\leq_r^{\mathrm{p}}$-hard and $A \in \mathcal{C}$. Intuitively a complete language A for a class $\mathcal{C}$ is the most difficult language in the class, since an easy algorithm for A would give an easy algorithm for any language in $\mathcal{C}$. The most common polynomial time reducibilities are $\leq_{\mathrm{T}}^{\mathrm{p}}$, which means no restriction on oracle access, $\leq_{\mathrm{tt}}^{\mathrm{P}}$, which means that each query does not depend on the answers to previous queries, and $\leq_{\mathrm{m}}^{\mathrm{P}}$, which allows only one query per input and with the additional restriction that the input is accepted if and only if the query is in the oracle. There is a whole range of polynomial time reducibilities of the form $\leq_{q(n)-\mathrm{tt}}^{\mathrm{P}}$ and $\leq_{q(n)-\mathrm{T}}^{\mathrm{P}}$, where $q(n)$ is a function bounding the number of queries allowed on inputs of length n.

As Schöning explains in the introduction of [Schö86], the first approach to the theory of complexity was mainly quantitative, since it corresponds to examining the amount of resources used in the solution of a particular problem; Structural Complexity became qualitative with the abstraction to complexity classes. This qualitative aspect seems inherent because a complexity class is or is not contained in another, a language is or is not complete for a class, etc. Despite that, a quantitative view can be also introduced in the study of complexity classes, as we explain below.

Consider a random experiment in which a language A is chosen by using an independent toss of a fair coin to decide whether each string is in A. This experiment defines Lebesgue probability distribution, usually referred to as Lebesgue measure. A probabilistic distribution on X can be viewed as a way of size classification of subsets of X, where probability 0 subsets are the smallest ones and probability 1 subsets are the largest ones. Bennett and Gill start in 1981 to use Lebesgue distribution to add a probabilistic quantitative aspect to Structural Complexity with results of the form 'a language is in the class $\mathcal{C}$ with probability μ'.

Let us briefly examine their results. In [BennGi], they study the class of oracles for which the class defined by polynomial time is different from that defined by nondeterministic polynomial time, showing that an oracle A separates P from NP with probability one. In the same paper they prove that with probability one for an oracle A, P equals the class of probabilistic polynomial time, denoted as BPP. After this, other similar results were obtained; for instance, an oracle A separates the polynomial time hierarchy from polynomial space with probability one (Cai [Cai], Babai [Baba]). But there is still something missing in this approach, since our main interest are recursive languages, and a language is recursive with probability 0 using Lebesgue probability distribution. Thus we know that most oracles separate P from NP, but we can infer nothing about the behaviour of recursive oracles from this result.

In 1987 Lutz started to remedy this situation. He defined resource-bounded measure as a way to provide size distinction for recursive classes. Lutz takes two main classes, exponential time, denoted E, and exponential space, denoted ESPACE, as comparison patterns, and, for each class X, tries to establish a size comparison between $X \cap \mathrm{E}$ and E, or between $X \cap \mathrm{ESPACE}$ and ESPACE.

The main concepts in Lutz's theory are measure 0 in $\mathcal{C}$ and measure 1 in $\mathcal{C}$, the class $\mathcal{C}$ being either E or ESPACE. Intuitively, a class X has measure 0 in $\mathcal{C}$ when $X \cap \mathcal{C}$ is negligibly small compared to $\mathcal{C}$, and a class X has measure 1 in $\mathcal{C}$ when $X \cap \mathcal{C}$ and $\mathcal{C}$ have similar sizes. (We will give the precise definitions in section 1.4.) Intermediate values $0 < \mu < 1$ of measure in E (ESPACE) could also be defined, but it is not necessary because all the complexity classes we are interested in, if at all measurable, have always either measure 0 or 1. This is a consequence of a variant of the Kolmogorov 0-1 law, which states that a class that is closed under finite variations can only be in one of three cases for both measure in E and in ESPACE, namely being non-measurable, having measure 0 and having measure 1 [Lutz92]. Indeed, all the classes studied in Complexity Theory are closed under finite variations, since membership of a language into a class is not affected by adding or deleting a finite number of strings.

Lutz gave a first definition of resource-bounded measure in his Ph.D. dissertation in 1987 (see [Lutz90]), and a new one in 1991 generalizing it (see [Lutz92]). Due to some technical inconveniences, the first formulation was mainly useful to prove results in ESPACE, while with the second one results in both E and ESPACE are easier to obtain.

The first goal of Lutz's approach is to extend existence results, of the form "there is a language in $\mathcal{C}$ that is not in X", to abundance results of the form "most languages in $\mathcal{C}$ are not in X", formally expressed as "the class X has measure 0 in $\mathcal{C}$". The interest of an abundance result is that it shows the typical behaviour of languages in a class, and therefore is more informative than an existence result that could as well correspond to an exception in the class. For instance the results in [Lutz92] extend Kannan's result that "there is a language in ESPACE that does not have polynomial size circuits" [Kann] to "the class of languages with polynomial size circuits has measure 0 in ESPACE", which means that most languages in ESPACE do not have polynomial size circuits. Abundance results in E are treated in Chapters 3 and 4 of this dissertation.

Another application of resource-bounded measure is in relation with the probabilistic method (developed in [AlonSp], [Erdö], [ErdöSp], [Spen]). Let A be a set where a probability distribution has been defined. If we want to prove an existence result of the form 'there exists $x \in A$ such that property Π holds for x', it may be easier to prove that the subset of the elements of A for which the property holds does not have probability (i.e. measure) 0. The easiness here comes from the use of powerful measure techniques that involve proving abundance, as opposed to constructing a particular object.

We can consider resource-bounded measure as a probabilistic method for a class $\mathcal{C}$. In order to prove that there exists a language in $\mathcal{C}$ with property Π, it may be easier to prove that the class $\{L \bigm| L \text{ has property } \Pi\}$ does not have measure 0 in $\mathcal{C}$. We will see a case where this is indeed true in Chapter 4.

A third aspect of resource-bounded measure is as a formal tool in Structural Complexity for the construction of new working hypothesis, characterization of complexity classes, etc. A first example is Lutz's characterization of the class BPP in terms of measure in ESPACE in [Lutz91a]. There exist resource-bounded measure hypothesis implying widely believed results that could not be obtained from reasonable classical complexity hypothesis. For instance Lutz shows in [Lutz91b] that if E does not have measure 0 in ESPACE then P = BPP. In Chapter 5 we discuss another useful resource-bounded measure hypothesis and its consequences.

The objective of this work is to study in deep resource-bounded measure, its possible generalizations to other complexity classes and its applications in the three exposed ways, namely the extension from existence to abundance results, the probabilistic method, and the identification of useful Structural Complexity hypothesis.

These applications concern mainly measure in E, for which very few results existed before the new formulation of Lutz's measure in 1991. The classes E (defined by linear exponential time) and E_2 (polynomial exponential time) have a rich and well studied reducibility structure, and are known to contain intractable problems, which makes them very suitable as base of comparison to other less known classes, such as NP. There is an interesting survey of measure in these classes in [Lutz93]. There, the results in Chapters 3, 4 and 5 are described in a broader context. A great part of the results to be described from now on are joint work with J.H. Lutz, as indicated in the references.

In this chapter we start by summarizing the main contributions of this Ph.D. dissertation. Then we review some common notation from Structural Complexity and finally we give a complete introduction to resource-bounded measure in sections 1.4, 1.5 and 1.6.

1.2 Main contributions

Extension to new classes

We have already mentioned Lutz's measure for the classes E and ESPACE as base classes, to which the other classes are compared. In the introduction we explained the interest of comparing with E, as developed in [Lutz93]. But there is also a technical point in the definition of resource bounded measure that makes it nontrivial to define a measure for any class below E. This difficulty is related to the use of characteristic sequences. Given a language A and a string x, the partial characteristic sequence $\chi_{A^{<x}}$ contains the answer to $y \in A$? for every y smaller than x. The definition of resource-bounded measure for a class $\mathcal{C}$ assumes that, given $A \in \mathcal{C}$, for each string x, the initial segment $\chi_{A^{<x}}$ can be computed within the resources allowed in $\mathcal{C}$ for an input of the length of x. Remark that this last condition requires at least exponential time.

We study in Chapter 2 the technical difficulties of translating Lutz's definition into PSPACE, the class of languages that can be recognized with polynomial space. We prove that the natural candidate of measure in PSPACE is not valid unless the unlikely consequence PSPACE = E_2 holds. We then propose a valid definition based on on-line computable functions, and use it to prove that a class of self-reducible languages has measure 0 in PSPACE. This chapter describes and extends results from [Mayo92b].

Measure versus Category: the P-bi-immune languages

A language A is P-bi-immune if neither A nor its complement has an infinite subset in P. We investigate in Chapter 3 the abundance of P-bi-immune languages in E. We prove that the class of P-bi-immune languages has measure 1 in E. This implies that almost every language in E is P-bi-immune, which extends the existence result in [BermHa].

Baire Category is a topological theory where there exist a concept of small class (denoted as meager or first-category) and a concept of big class (co-meager). This classification is incomparable with Lebesgue measure in the sense that there exist measure 0 classes that are co-meager and vice versa. Lutz defines in [Lutz90] a resource-bounded version of Baire Category. We prove that category in E and measure in E are incomparable as in the classical case, since the class of P-bi-immune languages is not meager or co-meager in E, while it has measure 1 in E, as indicated above. Notice that in this case the incomparability example is a naturally defined class, while in the classical case the examples were more artificial. The results described in this chapter appear in [Mayo92a].

Application to nonuniform models

Structural Complexity also studies nonuniform complexity classes as a way of comparing uniform and nonuniform computation models. A nonuniform computing model, for example a Boolean circuit, works only with inputs of a fixed length. In order to recognize an infinite language A, a countable family of nonuniform devices is needed, such that for each natural n, an element of the family recognizes exactly the words of length n in A. We can then define nonuniform complexity classes by measuring the resources used by these families, for instance we can consider Boolean circuit size or depth, number of states in finite automata, branching program depth, etc.

In this context, we start by studying the class $\mathrm{P}/\log$, defined as the class of languages that can be recognized in polynomial time with a nonuniform advice of logarithmic length. We characterize this class in terms of Boolean circuits and then show that it has measure 0 in E.

The class P/poly contains those languages that can be recognized by a family of polynomial size circuits. In [KarpLi], P/poly is also characterized as the class of languages that are $\leq^{\mathrm{P}}_{\mathrm{T}}$-reducible to a sparse language, where a language is sparse when it has at most a polynomial number of strings for each length.

The open problem of whether exponential time is included in P/poly is hard since it does not relativize. In Chapter 4 we first study the relation between E and a subclass of P/poly, namely the subclass of languages that are $\leq^{\mathrm{P}}_{n^\alpha-\mathrm{tt}}$-reducible to a sparse set for $\alpha < 1$ (denoted as $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{SPARSE})$). This class is almost the largest subclass of P/poly for which we can use techniques that relativize in order to investigate its relation with E, since the question $\mathrm{E} \subseteq \mathrm{P}_{n-\mathrm{tt}}(\mathrm{SPARSE})$? is already nonrelativizable. In fact we prove that $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{SPARSE})$ has measure 0 in E, that is, almost every language in E is not in $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{SPARSE})$. Applying the probabilistic method, this shows that there exists a language in E that is not in $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{SPARSE})$, thus E does not have sparse $\leq^{\mathrm{P}}_{n^\alpha-\mathrm{tt}}$-hard languages. This result, which had not been proven before and that strengthens Watanabe's 1987 result for $\leq^{\mathrm{P}}_{O(\log n)-\mathrm{tt}}$-complete languages [Wata87c] is, to our knowledge, the first application of resource-bounded measure as a probabilistic method.

We also study P/poly inside the exponential time hierarchy that lies between the classes E and ESPACE, and is defined in [HartImS] as a family of classes with an increasing nondeterministic power. We use the 'approximate counting' techniques from Stockmeyer [Stoc85], to obtain the result that P/poly has measure 0 in the third level of the exponential time hierarchy. Some of the results involving $\mathrm{P}/\log$ appear in [HermMa]; the results involving the class $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{SPARSE})$ appear in [LutzMa94a].

Measure of the class NP

The hypothesis "NP does not have measure 0 in E" (roughly, that NP contains more than a negligible subset of exponential time), cannot be proven or refuted from our present knowledge. Even more, both by proving and by refuting it one would obtain solutions to nonrelativizable open problems on the relations between NP, P and E. In Chapter 5 we present evidence that "NP does not have measure 0 in E" is a reasonable hypothesis with many credible consequences.

The first such consequence deals with the difference in NP of the completeness notions corresponding to the reducibilities $\leq_{\mathrm{T}}^{\mathrm{p}}$ (Cook) and $\leq_{\mathrm{m}}^{\mathrm{p}}$(Karp-Levin). Since these are respectively the least and most restrictive reductions, the corresponding complete languages are believed to be different for many classes, and indeed it is known that there are $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete languages for E and NE that are not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete (see [BuhrHoT], [KoMo] and [Wata87b]).

Under the hypothesis that NP does not have measure 0 in E, we show in Chapter 5 that there is a language that is $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete but not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete, for NP. This conclusion, widely believed to be true (see [LongYo]), is not known to follow from P $\neq$ NP or other traditional complexity-theoretic hypotheses.

We prove additional consequences of NP does not have measure 0 in E, including the separation of many truth-table reducibilities in NP (e.g., k queries versus $k+1$ queries), the class separation E $\neq$ NE, and the existence of NP search problems that are not reducible to the corresponding decision problems.

Our results in Chapter 4 give us yet another consequence of the hypothesis that NP does not have measure 0 in E, namely that, for every real $\alpha < 1$, no $\leq_{n^{\alpha}-\mathrm{tt}}^{\mathrm{p}}$-hard language for NP is sparse. All this chapter is from [LutzMa94b].

R-Cones

Given a reducibility R, we can picture the upper semi-lattice defined by the preorder relation R on the class of all languages. Fix a language A and look at the two classes formed respectively by languages that are R-reducible to A and languages to which A is R-reducible. These two classes can be viewed as the two parts of the cone starting in vertex A. We call the first one the R-lower cone of A, and the second one the R-upper cone of A. We want to study the size of the upper and lower cones of a language A as a way of having information on the usefulness of A as oracle and on the amount of oracles A reduces to. In this line, Tang and Book study in [TangBo] the Lebesgue measure of R-cones for various reducibilities R, and Juedes and Lutz study in [JuedLu94a] the resource-bounded measure of $\leq_{\mathrm{m}}^{\mathrm{p}}$-cones in E.

We say that a language A is R-weakly-hard for a class $\mathcal{C}$ when the R-lower cone of A does not have measure 0 in $\mathcal{C}$, and that A is R-weakly-complete when A is R-weakly-hard and $A \in \mathcal{C}$. Intuitively, A is weakly-hard when a non-negligible subclass of $\mathcal{C}$ is reducible to A. Clearly every complete set A is weakly-complete, since its lower cone contains the whole $\mathcal{C}$. It is interesting to know whether the opposite holds, that is, whether every weakly-complete problem is complete. Since complete problems are considered the most intractable in a class, a negative answer would imply the existence of a third level of intractability in $\mathcal{C}$, between the lowest level and the level of complete sets. Lutz's new technique of 'martingale diagonalization' [Lutz94a] gives a construction of a language that is weakly-complete but not complete in the usual sense for the class E with reducibility $\leq_{\mathrm{m}}^{\mathrm{p}}$. In joint work with S. Fenner and J.H. Lutz we have extended this technique to the class of all recursive languages with reducibility $\leq_{\mathrm{T}}$ [FennLuM].

Given a reducibility R, the class ALMOST-R is defined as the class of languages A such that the R-upper cone of A has Lebesgue probability 1. The "ALMOST-R" formalism, studied for instance in [Book94] and [BookLuW], provides characterizations of some interesting complexity classes, among others, P=ALMOST-$\leq^{\mathrm{p}}_{\mathrm{m}}$[Ambo], P=ALMOST-$\leq^{\mathrm{p}}_{\mathrm{btt}}$ [TangBo], BPP=ALMOST-$\leq^{\mathrm{p}}_{\mathrm{T}}$ ([Ambo], [BennGi]), BPP=ALMOST-$\leq^{\mathrm{p}}_{\mathrm{tt}}$ [TangBo], AM=ALMOST-$\leq^{\mathrm{NP}}_{\mathrm{T}}$ ([Cai], [NisaWi]), PH=ALMOST-$\leq^{\mathrm{PH}}_{\mathrm{T}}$ ([Cai], [NisaWi]) and IP=ALMOST-IP [Breu]. The notion of Martin-Löf algorithmically random language is the strongest definition of random language that is considered to represent randomness of individual infinite sequences. Book, Lutz and Wagner ([Book94], [BookLuW]) have characterized the classes of the form ALMOST-R as the class of recursive languages that can be R-reduced to Martin-Löf algorithmically random languages. For each natural n, we consider a subclass of Martin-Löf random languages, denoted n-random languages, and obtain new characterizations of the ALMOST-R classes (joint work with R. Book). These characterizations have the form 'A language A in Δ^0_n (the nth level of the Kleene arithmetical hierarchy) is in ALMOST-R if and only if A is R-reducible to an n-random language'. This gives us an idea of, for instance, how difficult can $\leq^{\mathrm{p}}_{\mathrm{T}}$-oracles for BPP be. We also see that n-random oracles are useless for the class Δ^0_n − REC. These results are described in [BookMa].

There is an active ongoing research on the topics in this chapter ([AmboNeT], [AmboTeZ], [JuedLu94b]); we include a summary of the new results and a description of the open problems.

1.3 Preliminaries

We start by fixing some notation on strings and languages. We will use the alphabet $\Sigma = \{0,1\}$. A *string* is a finite sequence $x \in \{0,1\}^*$. We write $|x|$ for the length of x. The unique string of length 0 is λ, the *empty string.* If x and y are two strings, then $x \leq y$ if $|x| < |y|$ or $|x| = |y|$ and x precedes y in alphabetical order. We call this order relation on strings lexicographical order. Let $s_0, s_1, s_2, \ldots$ be the standard enumeration of the strings in $\{0,1\}^*$ in lexicographical order. A *sequence* is an element of $\{0,1\}^\infty$. If x is a string and y is a string or sequence, then xy is the concatenation of x and y. If x is a string and $k \in \mathbb{N} \cup \{\infty\}$, then x^k is the k-fold concatenation of x with itself. If x is a string and y is a string or sequence, then $x \sqsubseteq y$ iff there exists a string or sequence z such that $y = xz$, and $x \sqsubsetneq y$ if $x \sqsubseteq y$ and $x \neq y$. If w is a string or sequence and $0 \leq i < |w|$ then $w[i]$ denotes the ith bit of w.

A *language* is a set of strings. A *class* is a set of languages. For each language A and $n \in \mathbb{N}$ we denote as $A^{=n}$ the set of all strings in A of length n, and as $A^{\leq n}$ the set of all strings in A of length less or equal to n.

Given a set A, we denote as $\mathcal{P}(A)$ the power set of A, that is, the set of all subsets of A.

We will use the *characteristic sequence* χ_L of a language L, defined as follows:

$$\chi_L \in \{0,1\}^\infty \text{ and } \chi_L[i] = 1 \text{ iff } s_i \text{ belongs to } L.$$

We identify through characteristic sequences the class $\mathcal{P}(\{0,1\}^*)$ of all languages over $\{0,1\}$ with the set $\{0,1\}^\infty$ of all sequences. Let $w \in \{0,1\}^*$. We define $\mathbf{C}_w$, the *cylinder generated by* w, as the class of languages $\{x \in \{0,1\}^\infty \mid w \sqsubseteq x\}$.

The *complement of a class* of languages X is $X^c = \{0,1\}^\infty - X$. The *complement of a language* $L \subseteq \{0,1\}^*$ is $\bar{L} = \{0,1\}^* - L$; using characteristic sequence notation, if $L \in \{0,1\}^\infty$ then $\bar{L} \in \{0,1\}^\infty$ is such that for each $i \in \mathbb{N}$, $\bar{L}[i] \neq L[i]$. For a class $X \subseteq \{0,1\}^\infty$ we define the *class of complements* as co-$X = \{\bar{L} \bigm| L \in X\}$.

The *symmetric difference* of two sets A and B, denoted $A\Delta B$, is defined by $A\Delta B = (A \cup B) - (A \cap B)$.

Next we introduce Lebesgue measure on $\{0,1\}^\infty$. Consider a random experiment in which a language A is chosen by using independent tosses of a fair coin to decide whether each string $x \in \{0,1\}^*$ is in A. This experiment defines *Lebesgue probability distribution* on $\{0,1\}^\infty$. Given a class $X \subseteq \{0,1\}^\infty$, we denote as $\Pr(X)$ the probability associated to the event $A \in X$, when A is randomly chosen according to Lebesgue distribution. The value $\Pr(X)$ is not defined for every subset of $\{0,1\}^\infty$, and we say that a set X is Lebesgue-measurable if $\Pr(X)$ is defined. The partial function $\Pr\colon \mathcal{P}(\{0,1\}^\infty) \to [0,1]$ is called Lebesgue measure on $\{0,1\}^\infty$. In the next section we give an equivalent constructive definition of Lebesgue measure on $\{0,1\}^\infty$ to be used in the formulation of resource-bounded measure.

Although Lebesgue measure is usually defined on subsets of real numbers, notice that we can identify $\{0,1\}^\infty$ with the unit interval $[0,1]$ by associating to each $x \in \{0,1\}^\infty$ the real number that has $0.x$ as its standard binary representation. Via this identification Lebesgue measure on $\{0,1\}^\infty$ can be translated into Lebesgue measure on $[0,1]$.

Given two properties of languages $\mathcal{Q}$, $\mathcal{R}$, we will denote as $\Pr_C[\mathcal{Q}(C)]$ the Lebesgue measure of the class $\{C \bigm| \mathcal{Q}(C)\}$, that is

$$\Pr_C[\mathcal{Q}(C)] = \Pr(\{C \bigm| \mathcal{Q}(C)\}),$$

and we will denote as $\Pr_C[\mathcal{Q}(C) \bigm| \mathcal{R}(C)]$ the conditional probability of $\mathcal{Q}(C)$ given $\mathcal{R}(C)$, that is,

$$\Pr_C[\mathcal{Q}(C) \mid \mathcal{R}(C)] = \frac{\Pr(\{C \mid \mathcal{Q}(C)\} \cap \{C \mid \mathcal{R}(C)\})}{\Pr(\{C \bigm| \mathcal{R}(C)\})}.$$

Let X be a class of languages. We say that X is *closed under finite variations* if when $A \in X$ and $|A\Delta B| < \infty$ then $B \in X$. We say that X is *closed under finite translations* if $B \in X$ when $A \in X$ and there exists $w \in \{0,1\}^*$ such that $A = w \cdot B$.

Next we fix some notation on Complexity Classes. For a complete introduction to Turing Machines and Complexity Classes see for instance [BalcDíG].

Our computation model is the multi-tape oracle Turing machine, with a read-only input tape and a write-only oracle tape. We will work with oracle Turing machines that halt on every oracle and every input. For a Turing machine M and a language A, $L(M)$ denotes the set accepted by M with the empty oracle, and $L(M,A)$ stands for the set accepted by machine M with oracle A. Given $t\colon \mathbb{N} \to \mathbb{N}$, we say that a Turing machine M recognizes a language L in time t when on each input x, M halts with output $L(x)$ in time less or equal than $t(|x|)$. Analogously, M recognizes a language L in space t when on each input x, M halts with output $L(x)$ using memory space less or equal than $t(|x|)$. $\{M_i \mid i \in \mathbb{N}\}$ is a standard enumeration of all deterministic oracle Turing machines.

For each nondecreasing function $t\colon \mathbb{N} \to \mathbb{N}$, we denote as DTIME$(t)$ the class of all languages that can be recognized by a deterministic machine in time t, and as DSPACE(t) the class of all languages that can be recognized by a deterministic machine in space t. Let NTIME(t) be the class of languages than can be recognized by a nondeterministic machine in time t, and let NSPACE(t) be the class of languages that can be recognized by a nondeterministic machine in space t. DTIMEF(t) and DSPACEF(t) are the corresponding classes of functions that can be computed in time t and space t, respectively. Unless indicated otherwise, when we bound the space used in the computation of a function we are also bounding the output space. For each language A, let DTIMEF$^A(t)$ be the class of all fuctions that can be computed by a deterministic machine in time t when having access to oracle A, and analogously we define DSPACEF$^A(t)$.

For each class $\mathcal{F}$ of functions from $\mathbb{N}$ to $\mathbb{N}$, we write DTIME$(\mathcal{F})$ for $\bigcup_{t\in\mathcal{F}}$ DTIME(t), and analogously for NTIME$(\mathcal{F})$, DSPACE$(\mathcal{F})$, NSPACE$(\mathcal{F})$, DTIMEF$(\mathcal{F})$ and DSPACEF$(\mathcal{F})$. For each language A, DTIMEF$^A(\mathcal{F})$ denotes $\bigcup_{t\in\mathcal{F}}$ DTIMEF$^A(t)$, and in the same way we have DSPACEF$^A(\mathcal{F})$. Let $\mathcal{C}$ be a class of languages. Then

$$\mathrm{DTIMEF}^{\mathcal{C}}(\mathcal{F}) = \bigcup_{A\in\mathcal{C}} \mathrm{DTIMEF}^{A}(\mathcal{F})$$

and with a similar meaning DSPACEF$^{\mathcal{C}}(\mathcal{F})$ is defined.

Let RE be the class of recursively enumerable languages, and REC be the class of recursive languages. We use the following notation for classes of languages

$$\begin{array}{ll}
\mathrm{P} = \bigcup_{k\in\mathbb{N}} \mathrm{DTIME}(n^k) & \mathrm{E} = \bigcup_{c>0} \mathrm{DTIME}(2^{cn}) \\
 & \mathrm{E}_2 = \bigcup_{k\in\mathbb{N}} \mathrm{DTIME}(2^{n^k}) \\
\mathrm{NP} = \bigcup_{k\in\mathbb{N}} \mathrm{NTIME}(n^k) & \mathrm{NE} = \bigcup_{c>0} \mathrm{NTIME}(2^{cn}) \\
\mathrm{LINSPACE} = \bigcup_{c>0} \mathrm{DSPACE}(cn) & \mathrm{ESPACE} = \bigcup_{c>0} \mathrm{DSPACE}(2^{cn}). \\
\mathrm{PSPACE} = \bigcup_{k\in\mathbb{N}} \mathrm{DSPACE}(n^k) & \mathrm{E_2SPACE} = \bigcup_{k\in\mathbb{N}} \mathrm{DSPACE}(2^{n^k})
\end{array}$$

Let **all** be the class of all functions $f\colon \{0,1\}^* \to \{0,1\}^*$, and rec be the class of recursive functions in **all**. We will denote different classes of functions as follows,

$$\begin{array}{ll}
\mathrm{p} = \bigcup_{k\in\mathbb{N}} \mathrm{DTIMEF}(n^k) & \mathrm{pspace} = \bigcup_{k\in\mathbb{N}} \mathrm{DSPACEF}(n^k) \\
\mathrm{p}_2 = \bigcup_{k\in\mathbb{N}} \mathrm{DTIMEF}(2^{(\log n)^k}) & \mathrm{p_2space} = \bigcup_{k\in\mathbb{N}} \mathrm{DSPACEF}(2^{(\log n)^k}).
\end{array}$$

For each class $\mathcal{C}$, $\mathrm{p}(\mathcal{C}) = \bigcup_{k \in \mathbb{N}} \mathrm{DTIMEF}^{\mathcal{C}}(n^k)$.

We fix a one to one pairing function $\langle , \rangle$ from $\{0,1\}^* \times \{0,1\}^*$ onto $\{0,1\}^*$ such that the pairing function and its associated projections, $\langle x, y \rangle \mapsto x$ and $\langle x, y \rangle \mapsto y$ are computable in polynomial time, and such that for $x, y \in \{0,1\}^*$, $x \leq \langle x, y \rangle$, $y \leq \langle x, y \rangle$. For $k \geq 2$ and strings $y_1, \ldots, y_k$, $\langle y_1, \ldots, y_k \rangle$ stands for $\langle\langle\langle \ldots y_1, y_2 \rangle, \ldots \rangle, y_k \rangle$.

For a function $f \colon \{0,1\}^* \to \{0,1\}^*$, we write f^n for the n-fold composition of f with itself.

The boolean value of a condition γ is denoted with $[\![\gamma]\!]$.

A *relativized class* is a function $\mathcal{C} : \{0,1\}^\infty \longrightarrow \mathcal{P}(\{0,1\}^\infty)$. A *recursive presentation* of a relativized class $\mathcal{C}$ of languages is a total recursive function $f : \mathbb{N} \longrightarrow \mathbb{N}$ such that for every language A and every $i \in \mathbb{N}$, every computation of $M_{f(i)}(A)$ is halting and $\mathcal{C}(A) = \{L(M_{f(i)}, A) \mid i \in \mathbb{N}\}$. A relativized class is *recursively presentable* if it has a recursive presentation.

A *reducibility* is a relativized class. A *bounded reducibility* is a relativized class that is recursively presentable. If R is a reducibility, then we use the notation $A \leq^R B$ to indicate that $A \in R(B)$.

If R is a reducibility and $\mathcal{C}$ is a set of languages, write $R(\mathcal{C})$ for $\bigcup_{A \in \mathcal{C}} R(A)$.

Given a reducibility R, we say that a language A is *R-hard* for a class $\mathcal{C}$ if $\mathcal{C} \subseteq R(A)$, and that A is *R-complete* for $\mathcal{C}$ if $A \in \mathcal{C}$ and A is R-hard for $\mathcal{C}$.

We will discuss a variety of specialized polynomial-time reducibilities, in addition to the well-known reducibilities $\leq^{\mathrm{p}}_{\mathrm{T}}$ and $\leq^{\mathrm{p}}_{\mathrm{m}}$. These include $\leq^{\mathrm{p}}_{q(n)-\mathrm{T}}$ (*Turing reducibility with $q(n)$ queries* on inputs of length n), $\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}}$ (*truth-table reducibility with $q(n)$ queries* on inputs of length n, where $q \colon \mathbb{N} \to \mathbb{N}$ is a query-counting function), $\leq^{\mathrm{p}}_{\mathrm{tt}}$(*truth-table* reducibility), and $\leq^{\mathrm{p}}_{\mathrm{btt}}$ (*bounded truth-table* reducibility). We now indicate the meanings of these specialized reducibilities.

Let $A, B \subseteq \{0,1\}^*$. The condition $A \leq^{\mathrm{p}}_{\mathrm{T}} B$ means that there is a polynomial time-bounded oracle Turing machine M such that $A = L(M, B)$. For $q \colon \mathbb{N} \to \mathbb{N}$, the condition $A \leq^{\mathrm{p}}_{q-\mathrm{T}} B$ means that there is a polynomial time-bounded Turing machine M such that $A = L(M, B)$ and M makes $\leq q(|x|)$ oracle queries on each input $x \in \{0,1\}^*$.

Given a query-counting function $q \colon \mathbb{N} \to \mathbb{N}$, a *q-query function* is a function f with domain $\{0,1\}^*$ such that, for all $x \in \{0,1\}^*$,

$$f(x) = (f_1(x), ..., f_{q(|x|)}(x)) \in (\{0,1\}^*)^{q(|x|)}.$$

Each $f_i(x)$ is called a *query* of f on input x. A *q-truth table function* is a function g with domain $\{0,1\}^*$ such that, for each $x \in \{0,1\}^*$, $g(x)$ is the encoding of a boolean function $g(x) \colon \{0,1\}^{q(|x|)} \to \{0,1\}$. A $\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}}$*-reduction* is an ordered pair (f, g) such that f is a q-query function, g is a q-truth table function, and f and g are computable in polynomial time.

Let $A, B \subseteq \{0,1\}^*$. A $\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}}$*-reduction of A to B* is a $\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}}$-reduction (f, g) such that, for all $x \in \{0,1\}^*$,

$$[\![x \in A]\!] = g(x)([\![f_1(x) \in B]\!] ... [\![f_{q(|x|)}(x) \in B]\!]).$$

In this case we say that $A \leq^{\mathrm{p}}_{q(n)-\mathrm{tt}} B$ via (f,g), and denote it $A = (f,g)(B)$. We say that A is $\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}}$*-reducible to* B, and write $A \leq^{\mathrm{p}}_{q(n)-\mathrm{tt}} B$, if there exists (f,g) such that $A = (f,g)(B)$.

The condition $A \leq^{\mathrm{p}}_{\mathrm{tt}} B$ means that there exists a polynomial q such that $A \leq^{\mathrm{p}}_{q(n)-\mathrm{tt}} B$. The condition $A \leq^{\mathrm{p}}_{\mathrm{btt}} B$ means that there exists a constant k such that $A \leq^{\mathrm{p}}_{k-\mathrm{tt}} B$. (This is equivalent to saying that there exists a possibly different constant k such that $A \leq^{\mathrm{p}}_{k-\mathrm{T}} B$.)

PH is the polynomial time hierarchy, defined as follows

(i) $\Sigma^{\mathrm{p}}_1 = \mathrm{NP}$,

(ii) for every $n > 0$, $\Sigma^{\mathrm{p}}_{n+1} = \mathrm{NP}(\Sigma^{\mathrm{p}}_n)$,

(iii) for every $n > 0$, $\Pi^{\mathrm{p}}_n = \text{co-}\Sigma^{\mathrm{p}}_n$,

(iv) for every $n > 0$, $\Delta^{\mathrm{p}}_n = \mathrm{P}(\Sigma^{\mathrm{p}}_{n-1})$,

(v) PH$= \bigcup_{n>0} \Sigma^{\mathrm{p}}_n$.

We will denote with AH the arithmetical hierarchy of languages, that is,

(i) $\Sigma^0_1 = \mathrm{RE} = \{A \subseteq \{0,1\}^* \mid A \text{ is recursively enumerable}\}$,

(ii) for every $n > 0$, $\Sigma^0_{n+1} = \mathrm{RE}(\Sigma^0_n)$,

(iii) for every $n > 0$, $\Pi^0_n = \text{co-}\Sigma^0_n$,

(iv) for every $n > 0$, $\Delta^0_n = \Sigma^0_n \cap \Pi^0_n$,

(v) $AH = \bigcup_{n>0} \Sigma^0_n$.

We use the following form of the Chernoff bound.

Lemma 1.1. *[Cher], [HageRü].* Let $\epsilon > 0$, let $N \in \mathbb{N}$. Then

$$\sum_{k=0}^{\epsilon \frac{N}{2}} \binom{N}{k} \leq 2^N \cdot e^{-\frac{\epsilon^2 N}{6}}.$$

In particular, taking $\epsilon = \frac{2}{j+1}$, where $j \in \mathbb{N}$,

$$\sum_{k=0}^{\frac{N}{j+1}} \binom{N}{k} \leq 2^N \cdot e^{-\frac{N}{2(j+1)^2}}.$$

Proof . See [HageRü]. ∎

Finally, we introduce the concept of Kolmogorov complexity. We fix a Universal Turing Machine U. Using it we can denote the unbounded Kolmogorov Complexity of a word w as follows

Definition 1.2. $\mathrm{K}(w) = \min\{|x| / U(x) = w\}$,

The Kolmogorov complexity of a string w is the length of the shortest program, which, when given as input to U, will lead U to write down w as output. The choice of U as the base Universal Machine is irrelevant, as long as a Universal Machine is used, since the Kolmogorov complexity would change only by an additive constant.

Hartmanis introduces in [Hart] a tool we will use in Chapter 4: time-bounded Kolmogorov Complexity. We follow the notation in [Hart]. Using U and functions f and $g : N \to N$ define the class of time-bounded Kolmogorov complexity sets $\mathrm{K}[f, g]$ as follows

Definition 1.3. $L \in \mathrm{K}[f, g]$ iff $\forall x \in L\ \exists w$, $|w| \le f(|x|)$ such that $U(w) = x$ in time $\le g(|x|)$.

Thus, $\mathrm{K}[f, g]$ is the class of the sets whose strings can be compressed by a factor of f, and which can also be recovered from their compressed form within the time bound g.

Observe that, despite the similarity of notation, $\mathrm{K}(w)$ denotes a function from words to nonnegative integers, while $\mathrm{K}[f, g]$ is a class of languages.

For families of functions $\mathcal{F}$, $\mathcal{G}$ we have

Definition 1.4. $L \in \mathrm{K}[\mathcal{F}, \mathcal{G}]$ iff there exists $f \in \mathcal{F}$, $g \in \mathcal{G}$ such that $L \in \mathrm{K}[f, g]$.

We will use the class $\mathrm{K}[\log, \mathrm{poly}]$, where $\log = O(\log n)$ and $\mathrm{poly} = \{n^{O(1)}\}$.

1.4 Resource-bounded measure

In this section we present resource-bounded measure, a method to classify complexity classes depending on their size. Resource-bounded measure was introduced by Lutz in [Lutz92]. (The earlier formulation of [Lutz90] has a number of technical inconveniences, and is not used anymore.) This theory is a generalization of a powerful mathematical tool, Lebesgue measure. Let us explain the meaning of ‘generalization’ here.

Our goal is to define a measure in $\mathcal{C}$, where $\mathcal{C}$ can take one of the following values E, E_2, ESPACE, E_2SPACE and REC. Intuitively, a measure in $\mathcal{C}$ is a function $\mu\colon \mathcal{P}(\mathcal{C}) \to [0, 1]$ with some additivity properties, whose main purpose is to classify by size criteria the subclasses of $\mathcal{C}$. Given a recursive class $\mathcal{C}$, we could define a measure μ in $\mathcal{C}$ as a restriction of Lebesgue measure to $\mathcal{P}(\mathcal{C})$. But this would be useless, because since every countable class has Lebesgue measure 0 (that is, minimal size) and recursive classes are always countable, μ would be identically 0.

In order to obtain a non-trivial measure on the mentioned recursive classes, Lutz takes a constructive definition of Lebesgue measure and bounds the resources allowed in the process. Intuitively, we restrict the measurable sets to those from the Lebesgue measurable ones that can be ‘feasibly measured’. We next give this constructive definition of Lebesgue measure by using betting games, where we will be able to bound the resources used by the player.

We consider a game in which there is a player with starting capital $0 < c_0 \in \mathbf{R}$ and a hidden language L. The player bets part of his money on the successive bits of χ_L, making money on a double or nothing fashion. The game goes as follows

Step 0: The player bets a_0, a part of c_0, either that $\lambda \in L$ or that $\lambda \notin L$. If he wins, he gets double, that is $2 \times a_0$, and his capital is now $c_1 = c_0 + a_0$. If he loses, he gets nothing and his capital is now $c_1 = c_0 - a_0$.

*Step n, $n > 0$:*With the information $[\![s_0 \in L]\!] \dots [\![s_{n-1} \in L]\!]$, the player bets a_n, a part of c_n, either that $s_n \in L$ or that $s_n \notin L$. If he wins, he gets double, that is $2 \times a_n$, and his capital is now $c_{n+1} = c_n + a_n$. If he loses, he gets nothing and his capital is now $c_{n+1} = c_n - a_n$.

The game goes on eternally, and we say that the player *succeeds* if he gets infinite money, that is to say, if the upper limit of $\{c_n\}$ is infinite as n goes to infinity.

The player tries to find a betting strategy that is always useful. A *strategy* for this game is a function $a\colon \{0,1\}^* \to \{0,1\} \times [0,\infty)$ that tells the player how much to bet, depending on the information the player has. That is, if $[\![s_0 \in L]\!] \dots [\![s_{n-1} \in L]\!] = w$, $w \in \{0,1\}^*$, and $a(w) = (b\,,u)$, the player should bet an amount of $a_n = u$ that $[\![s_n \in L]\!] = b$, according to the strategy a.

We can now compute the capital a player has when using this strategy a and represent it via a function $d_a\colon \{0,1\}^* \to [0,\infty)$, with the meaning that, if $[\![s_0 \in L]\!] \dots [\![s_{n-1} \in L]\!] = w$, $w \in \{0,1\}^*$, then the player's capital, after having bet on $s_0, \dots, s_{n-1}$ according to a, is $c_n = d_a(w)$. The value $d_a(\lambda)$ thus represents the starting capital c_0.

From a we can compute d_a and vice versa:

$$a(w) \quad = \begin{cases} \big(0, d_a(w0) - d_a(w)\big) & \text{if } d_a(w0) \geq d_a(w) \\ \big(1, d_a(w1) - d_a(w)\big) & \text{if } d_a(w1) \geq d_a(w) \end{cases}$$

Let $b \in \{0,1\}$:

$$d_a(wb) = \begin{cases} d_a(w) + u & \text{if } a(w) = (b\,,u) \\ d_a(w) - u & \text{if } a(w) = (1 - b\,,u). \end{cases}$$

From now on we will represent a strategy a by its capital function d_a, which we call a martingale.

Definition 1.5. A *martingale* is a function $d\colon \{0,1\}^* \to [0,\infty)$ satisfying

$$d(w) = \frac{d(w0) + d(w1)}{2} \tag{1.1}$$

for all $w \in \{0,1\}^*$.

Martingales were extensively used by Schnorr ([Sch70], [Sch71a], [Sch71b], [Sch73]) in his investigation of random and pseudorandom sequences.

(1.1) is the only condition that a function must fulfill to be a martingale and it is imposed by the double or nothing fashion in which we defined the game. Notice that if d is a martingale then for each $w \in \{0,1\}^*$, $d(w) \leq 2^{|w|} \cdot d(\lambda)$.

A martingale will be successful for a language L if the player using this martingale is successful when playing with L as the hidden language.

Definition 1.6. A martingale d is *successful* for a language $x \in \{0,1\}^\infty$ iff

$$\limsup_{n\to\infty} d(x[0 \dots n]) = \infty.$$

For each martingale d, we denote the set of all languages for which d is successful as $\mathrm{S}^\infty[d]$, that is

$$\mathrm{S}^\infty[d] = \{x \mid \limsup_{n\to\infty} d(x[0\dots n]) = \infty\}.$$

(This notation is chosen for consistency with other measure values; see section 1.6.)

We are now ready to define Lebesgue measure.

Definition 1.7. A class $X \subseteq \{0,1\}^\infty$ has *Lebesgue-measure 0* iff there exists a martingale d such that $X \subseteq \mathrm{S}^\infty[d]$, that is, for any $L \in X$, d is successful for L.

Intuitively, a class X has measure 0 when there exists a single strategy that is good for predicting any language in the class X.

Definition 1.8. A class $X \subseteq \{0,1\}^\infty$ has *Lebesgue-measure 1* iff X^c (the complement of X) has Lebesgue measure 0.

We only define measure 0 and measure 1 because we are always interested in classes that are closed under finite variations, and from the Kolmogorov 0-1 law (Theorem 21.3 in [Oxto]), these classes can only have measure 0 or measure 1, if they are measurable at all.

The definition we just introduced is just a restatement of more classical formulations of Lebesgue measure, for instance the one we sketched in the preliminaries.

Going back to the initial problem of defining a non trivial measure inside REC, E, E_2, ESPACE or E_2SPACE, what we do next is to restrict the martingales that can witness that a class has measure 0. We will require the martingales to be recursive and computable within certain time and space bounds, depending on the class where we are defining a measure.

Since martingales are real-valued functions and we want to use restrictions based on computing resources, we start by showing that rational valued martingales are sufficient to define Lebesgue measure. In fact we use dyadic rationals, that is, rational numbers with a finite binary expansion.

Let $\mathbf{D} = \{m2^{-n} \mid m,n \in \mathbb{N}\}$ be the set of *nonnegative dyadic rational numbers*. For purposes of computational complexity we represent each $q \in \mathbf{D}$ as $\langle u,v\rangle$, where u and v are the binary representations of the integer and fractional parts of q, respectively. In the same way, when we consider $k \in \mathbb{N}$, we are assuming the unary representation 0^k.

We will use the next auxiliary lemma in the proof of Lemmas 1.10 and 1.31. The lemma states that if c is a function that is very close to a martingale d, then we can define from c a martingale d' that acts exactly as d.

Lemma 1.9. Let d be a martingale. Let $c\colon \{0,1\}^* \to [0,\infty)$ be a function such that for each $w \in \{0,1\}^*$ $|c(w) - d(w)| \le 2^{-|w|}$. Let d' be recursively defined as follows

$$\begin{aligned} d'(\lambda) &= c(\lambda) + 2 \\ d'(wb) &= d'(w) + \frac{c(wb) - c(w\bar{b})}{2}. \end{aligned}$$

Then d' is a martingale and $\mathrm{S}^\infty[d] = \mathrm{S}^\infty[d']$.

Proof . Let d, c and d' be as in the hypothesis. d' fulfills trivially equality (1.1). In order to see that d' is a martingale, we have to show that it takes only nonnegative values. For this we prove by induction on $|w|$ that $d'(w) \geq d(w) + 2^{-|w|}$ for every $w \in \{0,1\}^*$.
For $w = \lambda$, $d'(\lambda) = c(\lambda) + 2 \geq d(\lambda) - 1 + 2$. For $w \in \{0,1\}^*$, $b \in \{0,1\}$ we have that, by induction hypothesis,

$$d'(wb) = d'(w) + \frac{c(wb) - c(w\bar{b})}{2} \geq d(w) + 2^{-|w|} + \frac{c(wb) - c(w\bar{b})}{2}$$

thus by our hypothesis on c,

$$d'(wb) \geq d(w) + 2^{-|w|} + \frac{d(wb) - d(w\bar{b})}{2} - 2^{-|w|-1} = d(wb) + 2^{-|w|-1},$$

the last equality following from d being a martingale.
Next we show by induction on $|w|$ that for every w,

$$|d(w) - d'(w)| \leq 4 - 2^{-|w|},$$

once this is done, the result follows immediately, since then for each $x \in \{0,1\}^\infty$

$$\left| \limsup_{m\to\infty} d(x[0..m]) - \limsup_{m\to\infty} d'(x[0..m]) \right| \leq 4$$

which implies that $\mathrm{S}^\infty[d] = \mathrm{S}^\infty[d']$.
For $w = \lambda$, $|d(\lambda) - c(\lambda) - 2| \leq 2^{-0} + 2$. For $w \in \{0,1\}^*$, $b \in \{0,1\}$,

$$\begin{gathered} |d'(wb) - d(wb)| \leq \\ |d'(w) - d(w)| + \left| d(w) + \frac{c(wb) - c(w\bar{b})}{2} - d(wb) \right| = \\ |d'(w) - d(w)| + \left| \frac{d(w\bar{b}) - c(w\bar{b})}{2} + \frac{c(wb) - d(wb)}{2} \right| \leq \\ |d'(w) - d(w)| + \left| \frac{d(w\bar{b}) - c(w\bar{b})}{2} \right| + \left| \frac{c(wb) - d(wb)}{2} \right| \leq \\ |d'(w) - d(w)| + 2^{-|w|-1} \end{gathered}$$

and by induction hypothesis this implies that

$$|d(wb) - d'(wb)| \leq 4 - 2^{-|w|} + 2^{-|w|-1} = 4 - 2^{-|w|-1}.$$

■

Lemma 1.10. For each martingale d, there exists a martingale $d'\colon \{0,1\}^* \to \mathbf{D}$ such that $\mathrm{S}^\infty[d] = \mathrm{S}^\infty[d']$.

Proof. Let d be a martingale. Using the fact that $\mathbf{D}$ is dense in $\mathbf{R}$, we define $c\colon \{0,1\}^* \to \mathbf{D}$ a function with values in $\mathbf{D}$ that is very close to d. For each $w \in \{0,1\}^*$ we fix $c(w) \in \mathbf{D}$ such that $|c(w) - d(w)| \leq 2^{-|w|}$.

We define recursively d' as follows

$$d'(\lambda) = c(\lambda) + 2$$
$$d'(wb) = d'(w) + \frac{c(wb) - c(w\bar{b})}{2}.$$

Notice that d' takes only values in $\mathbf{D}$. By Lemma 1.9, d' is a martingale and $\mathrm{S}^\infty[d] = \mathrm{S}^\infty[d']$, which completes the proof. ■

We now define the concept of measure resource-bounds, that are classes of recursive functions. By requiring the martingales to be in a certain measure resource-bound we will define measures for different classes.

We say that a set $\mathcal{F}$ of functions from $\mathbb{N}$ to $\mathbb{N}$ is a *family of bounds* if all functions in $\mathcal{F}$ are non-decreasing and for each $f, g \in \mathcal{F}$, $f \circ g$ is also in $\mathcal{F}$.

Definition 1.11. A class $\Gamma \subseteq \mathbf{all}$ is a *measure resource-bound* if $\mathrm{p} \subseteq \Gamma$ and Γ is in one of the following cases

a) $\Gamma = \mathbf{all}$,

b) $\Gamma = \mathrm{DTIMEF}^{\mathcal{C}}(\mathcal{F})$ for $\mathcal{F}$ a family of bounds and $\mathcal{C}$ a family of languages,

c) $\Gamma = \mathrm{DSPACEF}^{\mathcal{C}}(\mathcal{F})$ for $\mathcal{F}$ a family of bounds and $\mathcal{C}$ a family of languages.

We are specially interested in the following measure resource-bounds: p, p_2, pspace, $\mathrm{p}_2$space and rec, as we will see below.

We use Γ to denote a measure resource-bound in this dissertation, with the exception of Chapter 2 where Γ can be any class inside rec.

Now for each measure resource-bound Γ, we define μ_Γ as a restriction of Lebesgue measure to martingales in Γ. We then use μ_Γ to define a nontrivial measure on a suitable recursive class $\mathcal{C}$.

Definition 1.12. A class $X \subseteq \{0,1\}^\infty$ has Γ-*measure 0* (and we denote it $\mu_\Gamma(X) = 0$) iff there exists a martingale $d \in \Gamma$ such that, $X \subseteq \mathrm{S}^\infty[d]$.

Thus a class X has Γ-measure 0 when there exists a strategy in Γ that is good for predicting any language in the class X.

Definition 1.13. A set $X \subseteq \{0,1\}^\infty$ has Γ-*measure 1* (and we denote it $\mu_\Gamma(X) = 1$) iff X^c has Γ-measure 0.

Originally Lutz [Lutz92] defined Γ-measure using a type of Γ-approximable martingales. We will see in section 1.5 that his definition is equivalent to the one we just introduced.

Notice that taking $\Gamma = \mathbf{all}$ we again obtain Lebesgue measure.

As in the case of Lebesgue measure, there exists a resource-bounded generalization of the Kolmogorov 0-1 law [Lutz94b] by which classes that are closed under finite variations can only be in one of three cases, namely being Γ-measure 0, being Γ-measure 1 and being non-Γ-measurable. For this reason we only define Γ-measure 0 and Γ-measure 1. For the sake of completeness, we give a proof of the resource-bounded Kolmogorov 0-1 law in section 1.6.

The following step is to find the appropriate Γ such that from μ_Γ we can define a non-trivial measure in each of the classes E, E_2, ESPACE, E_2SPACE and REC. For $\mathcal{C}$ each of these classes, it is enough to find Γ such that $\mathcal{C}$ does not have Γ-measure 0, because then the restriction of μ_Γ to $\mathcal{P}(\mathcal{C})$ will be non-trivial. Since we want to have the biggest possible amount of measurable subclasses of $\mathcal{C}$, we are looking for the largest measure resource-bound Γ such that $\mathcal{C}$ does not have Γ-measure 0.

Notice that the complexity of a martingale is given in terms of the length of initial parts of characteristic sequences, while the complexity of a language is given in terms of the length of strings. We next develop the constructor formalism that establishes a relationship between both approaches. We associate with each measure resource-bound Γ a class of languages $\mathrm{R}(\Gamma)$.

Definition 1.14. $f \in \Gamma$ is a *constructor* iff $\forall w \in \{0,1\}^*$, $w \sqsubsetneq f(w)$.

Definition 1.15. If h is a constructor in Γ, then $\mathrm{R}(h)$ is the unique element in $\{0,1\}^\infty$ such that $\forall i\ h^i(\lambda) \sqsubseteq \mathrm{R}(h)$.

Definition 1.16. $\mathrm{R}(\Gamma)$ is the class of languages $\{\mathrm{R}(h) \mid h \text{ a constructor in } \Gamma\}$.

From the measure resource-bounds we mentioned, we obtain well-known classes as proven in the next lemma from [Lutz90].

Lemma 1.17. [Lutz90].

$$\begin{array}{ll} \mathrm{R}(\mathbf{all}) = \{0,1\}^\infty, & \mathrm{R}(\mathrm{p}_2) = \mathrm{E}_2, \\ \mathrm{R}(\mathrm{rec}) = \mathrm{REC}, & \mathrm{R}(\mathrm{pspace}) = \mathrm{ESPACE}, \\ \mathrm{R}(\mathrm{p}) = \mathrm{E}, & \mathrm{R}(\mathrm{p}_2\mathrm{space}) = \mathrm{E}_2\mathrm{SPACE}. \end{array}$$

Proof . We show that $\mathrm{R}(\mathrm{p}) = \mathrm{E}$, the rest of the cases being analogous.

Let δ be a constructor in p. Let $c > 0$ be such that $\delta \in \mathrm{DTIMEF}(n^c)$. The next algorithm recognizes $\mathrm{R}(\delta)$. On input $x = s_i$ the algorithm computes $\delta^k(\lambda)$ for successive values of k, until $|\delta^k(\lambda)| > i$. In this moment the algorithm outputs $\delta^k(\lambda)[i]$ that is exactly $\mathrm{R}(\delta)(x)$.

```
BEGIN
    INPUT $x = s_i$
    $w := \lambda$
    WHILE $|w| \leq i$ DO
        $w := \delta(w)$
    END WHILE
    OUTPUT $w[i]$
END
```

This algorithm on input x computes $\delta^k(\lambda)$ for k such that $|\delta^{k-1}(\lambda)| \leq i < |\delta^k(\lambda)|$. Since by the definition of constructor, $w \sqsubsetneq \delta(w)$ for every w, we know that $k \leq i+1$. Thus the algorithm computes at most $i+1$ values of the form $\delta(w)$ with $|w| \leq i$, taking time less than $(i+1) \cdot i^c$. Since $i+1 < 2^{|x|+1}$, then $\mathrm{R}(\delta) \in \mathrm{DTIME}(2^{(c+1)(|x|+1)})$. We have shown that $\mathrm{R}(\mathrm{p}) \subseteq \mathrm{E}$.

To see the converse, let $L \in \mathrm{E}$. Let $c > 0$ be such that $L \in \mathrm{DTIME}(2^{cn})$. Consider the constructor δ computed by the following algorithm

```
BEGIN
    INPUT $w$
    $n := |w|$
    $b := L(s_n)$
    OUTPUT $wb$
END
```

Clearly $\mathrm{R}(\delta) = L$. The time used by the algorithm on input w is $2^{c|s_n|}$, for $n = |w|$. Since $|w| \geq 2^{s_n}$, then $\delta \in \mathrm{DTIMEF}(n^c)$ and we have finished the proof of $\mathrm{R}(\mathrm{p}) = \mathrm{E}$. ■

We will use now Γ-measure to define a non trivial measure on the class $\mathrm{R}(\Gamma)$. The justification of why it is a non trivial measure is given by next theorem, which states that $\mathrm{R}(\Gamma)$ does not have Γ-measure 0.

Theorem 1.18. [Lutz92] Measure Conservation Theorem. For every martingale $d \in \Gamma$, there exists a language $L \in \mathrm{R}(\Gamma)$ such that d is not successful for L.

Proof . Let d be a martingale in Γ.

We define δ, a constructor in Γ, such that $\mathrm{R}(\delta) \notin \mathrm{S}^\infty[d]$ as follows

$$\delta(x) = \begin{cases} x0 & \text{if } d(x0) \leq d(x) \\ x1 & \text{otherwise.} \end{cases}$$

Then for each x, $d(\delta(x)) \leq d(x)$, and $d(\delta^{k+1}(\lambda)) \leq d(\delta^k(\lambda))$ for every $k \in \mathbb{N}$. This implies that

$$\limsup_{m\to\infty} d(\chi_{\mathrm{R}(\delta)}[0..m]) \leq d(\lambda),$$

and $\mathrm{R}(\delta) \notin \mathrm{S}^\infty[d]$. ■

After technical Lemma 1.35 in section 1.5, we will be able to show that Γ is, in a precise sense, the largest measure resource-bound such that $\mathrm{R}(\Gamma)$ does not have Γ-measure 0.

We finally define a meaningful measure in $\mathrm{R}(\Gamma)$ that is based on the restriction of Γ-measure to $\mathrm{R}(\Gamma)$.

Although we were looking for a measure in $\mathrm{R}(\Gamma)$, in order to simplify notation what we really do is to define a measure on $\{0,1\}^\infty$. For each $X \subseteq \{0,1\}^\infty$ we look at the subclass $X \cap \mathrm{R}(\Gamma)$.

Definition 1.19. A set $X \subseteq \{0,1\}^\infty$ has *measure 0 in* $\mathrm{R}(\Gamma)$ iff $X \cap \mathrm{R}(\Gamma)$ has Γ-measure 0. This is denoted as $\mu(X \mid \mathrm{R}(\Gamma)) = 0$.

Definition 1.20. A set $X \subseteq \{0,1\}^\infty$ has *measure 1 in* $\mathrm{R}(\Gamma)$ iff X^c has measure 0 in $\mathrm{R}(\Gamma)$. This is denoted as $\mu(X \mid \mathrm{R}(\Gamma)) = 1$.

Since taking $\Gamma = \mathrm{p}$, p_2, pspace, $\mathrm{p}_2$space and rec we obtain $\mathrm{R}(\Gamma) = \mathrm{E}$, E_2, ESPACE, E_2SPACE and REC, respectively, we have defined a nontrivial measure on those classes.

The following lemmas contain the first elementary properties of resource-bounded measure. Their proofs are straightforward from the above definitions.

Lemma 1.21. Let X, $Y \subseteq \{0,1\}^\infty$.

a) If $Y \subseteq X$ and X has Γ-measure 0 then Y has Γ-measure 0.

b) If $Y \subseteq X$ and X has measure 0 in $\mathrm{R}(\Gamma)$ then Y has measure 0 in $\mathrm{R}(\Gamma)$.

c) If X has Γ-measure 0 then X has measure 0 in $\mathrm{R}(\Gamma)$.

We show next that a finite union of Γ-measure 0 sets has Γ-measure 0. We will generalize this result to more general unions in section 1.5.

Lemma 1.22. Let $n \in \mathbb{N}$. If $X_1, \ldots, X_n$ have Γ-measure 0 then $\bigcup_{i=1}^{n} X_i$ has Γ-measure 0. If $X_1, \ldots, X_n$ have measure 0 in $\mathrm{R}(\Gamma)$ then $\bigcup_{i=1}^{n} X_i$ has measure 0 in $\mathrm{R}(\Gamma)$.

Proof. Given $d_1, \ldots, d_n$ martingales in Γ witnessing that $X_1, \ldots, X_n$ have Γ-measure 0, let d be the martingale in Γ defined as $d(w) = \sum_{i=1}^{n} d_i(w)$, for each $w \in \{0,1\}^*$. Clearly $\bigcup_{i=1}^{n} X_i \subseteq \mathrm{S}^\infty[d]$. ∎

Lemma 1.23. Let $X \subseteq \{0,1\}^\infty$. Let Γ, Γ' be two measure resource-bounds such that $\Gamma \subseteq \Gamma'$. If X has Γ-measure 0 then X has Γ'-measure 0, and if X has Γ-measure 1 then X has Γ'-measure 1.

In general, the implication

$$\mu(X \mid \mathrm{R}(\Gamma)) = 0 \overset{?}{\Longrightarrow} \mu(X \mid \mathrm{R}(\Gamma')) = 0.$$

is false. A counterexample is provided by Corollary 1.39 in section 1.5, stating that if Γ' contains a universal function for Γ then $\mu(\mathrm{R}(\Gamma) \mid \mathrm{R}(\Gamma')) = 0$. In this case if $X = \mathrm{R}(\Gamma)^c$, then $\mu(X \mid \mathrm{R}(\Gamma)) = 0$ and $\mu(X \mid \mathrm{R}(\Gamma')) = 1$. The implication

$$\mu(X \mid \mathrm{R}(\Gamma)) = 1 \overset{?}{\Longrightarrow} \mu(X \mid \mathrm{R}(\Gamma')) = 1$$

is also false. (For a counterexample, take $X = \mathrm{R}(\Gamma)$ if Γ' contains a universal function for Γ.)

In particular, for the classes we are more interested in we have the following corollary

Corollary 1.24. Let $X \subseteq \{0,1\}^\infty$. The following implications hold

$$\mu_{\mathrm{p}}(X) = 0 \Longrightarrow \mu(X \mid \mathrm{E}) = 0,$$
$$\mu_{\mathrm{p}_2}(X) = 0 \Longrightarrow \mu(X \mid \mathrm{E}_2) = 0,$$
$$\mu_{\mathrm{pspace}}(X) = 0 \Longrightarrow \mu(X \mid \mathrm{ESPACE}) = 0,$$
$$\mu_{\mathrm{pspace}_2}(X) = 0 \Longrightarrow \mu(X \mid \mathrm{E_2SPACE}) = 0,$$

$$\mu_{\mathrm{p}}(X) = 1 \Longrightarrow \mu(X \mid \mathrm{E}) = 1;$$
$$\mu_{\mathrm{p}_2}(X) = 1 \Longrightarrow \mu(X \mid \mathrm{E}_2) = 1;$$
$$\mu_{\mathrm{pspace}}(X) = 1 \Longrightarrow \mu(X \mid \mathrm{ESPACE}) = 1;$$
$$\mu_{\mathrm{pspace}_2}(X) = 1 \Longrightarrow \mu(X \mid \mathrm{E_2SPACE}) = 1.$$

The implications summarized by the next two diagrams hold

$$\begin{tikzcd}
\mu_{\mathrm{p}}(X) = 0 \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \mu_{\mathrm{p}_2}(X) = 0 \arrow[d, Rightarrow] & \\
\mu_{\mathrm{pspace}}(X) = 0 \arrow[r, Rightarrow] & \mu_{\mathrm{pspace}_2}(X) = 0 \arrow[r, Rightarrow] & \Pr(X) = 0;
\end{tikzcd}$$

$$\begin{tikzcd}
\mu_{\mathrm{p}}(X) = 1 \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \mu_{\mathrm{p}_2}(X) = 1 \arrow[d, Rightarrow] & \\
\mu_{\mathrm{pspace}}(X) = 1 \arrow[r, Rightarrow] & \mu_{\mathrm{pspace}_2}(X) = 1 \arrow[r, Rightarrow] & \Pr(X) = 1.
\end{tikzcd}$$

By the observation after Lemma 1.23, implications such as

$$\mu(X \mid \mathrm{E}) = 0 \overset{?}{\Longrightarrow} \mu(X \mid \mathrm{E}_2) = 0$$

or

$$\mu(X \mid \mathrm{E}) = 1 \overset{?}{\Longrightarrow} \mu(X \mid \mathrm{E}_2) = 1$$

are both false in general.

As a curiosity, we include a very recent result by Juedes and Lutz [JuedLu94b] showing an interesting relationship between measure in E and measure in E_2.

Theorem 1.25. (Lemma 4.3 in [JuedLu94b].) Let X be a class of languages. The following holds

$$\mu(\mathrm{P_m}(X) \mid \mathrm{E}_2) = 0 \Longrightarrow \mu(X \mid \mathrm{E}) = 0$$

and

$$\mu(\mathrm{P_m}(X) \mid \mathrm{E}_2) = 1 \Longrightarrow \mu(X \mid \mathrm{E}) = 1.$$

Notice that E_2 is the closure of E under polynomial-time many-one reductions.

Corollary 1.26. If X is closed downwards under polynomial-time many-one reductions, that is, $X = \mathrm{P_m}(X)$, then

$$\mu(X \mid \mathrm{E}_2) = 0 \Longrightarrow \mu(X \mid \mathrm{E}) = 0$$

and

$$\mu(X \mid \mathrm{E}_2) = 1 \Longrightarrow \mu(X \mid \mathrm{E}) = 1.$$

We now give easy examples of classes that are measure 0 and measure 1 in E. More interesting and elaborated proofs of measure 0 and measure 1 in different classes require the additivity lemmas we will prove in section 1.5.

Example 1 The class

$$X = \{A \mid \text{ there exist } n \text{ such that } |A^{\leq n}| \text{ is not a multiple of } 3\}$$

has measure 1 in E.

Proof. By Definition 1.20, we have to show that the class $Y = X^c$ has measure 0 in E. Let A be a language in Y. For every $n \in \mathbb{N}$, $|A^{\leq n}|$ is a multiple of 3. If we know the value of $|A^{\leq n} - \{1^n\}|$ we can guess $A(1^n)$ because if $|A^{\leq n} - \{1^n\}|$ is a multiple of 3 then 1^n must be out of A; if $|A^{\leq n} - \{1^n\}|$ is a multiple of 3 plus two, then 1^n must be in A. The case when $|A^{\leq n} - \{1^n\}|$ is a multiple of 3 plus two is impossible for A a language in Y.

Thus a successful strategy for Y will be to bet only on the bits corresponding to strings of the form 1^n, if $|A^{\leq n} - \{1^n\}|$ is a multiple of 3 we bet all our money to $1^n \notin A$, else we bet all our money to $1^n \in A$.

Notice that s_i is of the form 1^n if and only if i is of the form $2^m - 2$.

We define a martingale d that corresponds to the described strategy. Let $d(\lambda) = 1$. For each $w \in \{0,1\}^*$, assume that $d(w)$ has been already defined, then let $d(w0)$ and $d(w1)$ be as follows

If $|w| = 2^m - 2$ for some m then

$$d(w0) = \begin{cases} 2 \cdot d(w) & \text{if } \sum_{i=0}^{|w|-1} w[i] \text{ is a multiple of } 3 \\ 0 & \text{otherwise} \end{cases}$$

$$d(w1) = \begin{cases} 0 & \text{if } \sum_{i=0}^{|w|-1} w[i] \text{ is a multiple of } 3 \\ 2 \cdot d(w) & \text{otherwise} \end{cases}$$

Else, if $|w|$ is not of the form $2^m - 2$ then $d(w0) = d(w1) = d(w)$.

Let us see that d is successful on all languages in Y. Let $A \in X$, $n \in \mathbb{N}$. Then $|A^{\leq n}| = \sum_{i=0}^{2^{n+1}-2} A[i]$ is a multiple of 3. If $|A^{\leq n} - \{1^n\}| = \sum_{i=0}^{2^{n+1}-3} A[i]$ is a multiple of three, then $1^n \notin A$, and $A[2^{n+1} - 2] = 0$. By the definition of d then $d(A[0..2^{n+1} - 2]) = 2 \cdot d(A[0..2^{n+1} - 3]$. If $|A^{\leq n} - \{1^n\}|$ is not a multiple of three then $1^n \in A$ and $d(A[0..2^{n+1} - 2]) = 2 \cdot d(A[0..2^{n+1} - 3])$.

Since we only bet on bits of the form $2^m - 2$, then for each $n \geq 1$, $d(A[0..2^{n+1} - 3]) = d(A[0..2^n - 2])$. Thus $d(A[0..2^{n+1} - 2]) = 2 \cdot d(A[0..2^n - 2])$, $\limsup_m d(A[0..m]) = \infty$ and $X \subseteq \mathrm{S}^\infty[d]$.

Also, d is a martingale in p, because for each input w we can compute $d(w)$ from $d(w[0..|w| - 2])$ just by checking whether $|w|$ is of the form $2^m - 2$, and computing $\sum_{i=0}^{|w|-1} w[i]$, all of which which can be done in time linear in $|w|$; computing $d(w)$ requires computing $d(u)$

for each u prefix of w, and can thus be done in time quadratic in $|w|$. This proves that Y has p-measure 0 and by Lemma 1.21 c) we have that $X = Y^c$ has measure 1 in E. ■

Example 2

The class

$$X = \left\{ A \mid \text{ for every } n \in \mathbb{N},\ |A^{=n}| \geq \frac{2}{3}2^n \right\}$$

has measure 0 in E.

Notice that for every A, $|A^{=n}| \leq 2^n$ for every n.

Proof . Let us show that X has p-measure 0. This time the betting strategy cannot concentrate on certain bits for which we can guess the answer, since all we know about the languages in X is that they have many strings. The trick is that for each $i \in \mathbb{N}$, we bet more money on $s_i \in A$ than on $s_i \notin A$, thus making more money if $s_i \in A$ happens more often.

We define a martingale d as follows. $d(\lambda) = 1$ For each $w \in \{0,1\}^*$,

$$d(w0) = \frac{1}{2}d(w) \qquad d(w1) = \frac{3}{2}d(w).$$

Let us see that d is successful on every language in X. If $A \in X$ then for each $n \in \mathbb{N}$ we have that

$$d(A[0..2^{n+1}-2]) = \left(\frac{3}{2}\right)^{|A^{=n}|} \left(\frac{1}{2}\right)^{|(A^c)^{=n}|} \cdot d(A[0..2^n-2]) \geq$$

$$\geq \left(\frac{3}{2}\right)^{\frac{2}{3}2^n} \left(\frac{1}{2}\right)^{\frac{1}{3}2^n} \cdot d(A[0..2^n-2]) = \left(\frac{9}{8}\right)^{\frac{1}{3}2^n} \cdot d(A[0..2^n-2]).$$

This implies that $\limsup_m d(A[0..m]) = \infty$.

Clearly d is computable in linear time, thus X has p-measure 0. Therefore X has measure 0 in E. ■

As a last measure concept, we introduce the pseudo-random languages, which represent the notion of 'typical' language in this setting. Lutz uses this concept in [Lutz91a] to characterize the class BPP.

Definition 1.27. A language L is Γ-*random* iff it belongs to every class that has Γ-measure 1. We denote as Γ-rand the class of all Γ-random languages.

There exist several definitions of "random language", for individual languages. Each of them intuitively tries to capture those languages whose characteristic sequences have been obtained by some random process, for instance independent tosses of a fair coin. The strongest notion of randomness that is widely accepted is Martin-Löf randomness, discussed in Chapter 6. But every Martin-Löf random language is nonrecursive, and the interest of Γ-random languages is that they can be recursive and still be useful as a source of random bits; in fact if we have computing power Γ to check whether a certain language is random, then a Γ-random language looks truly random to us.

We remark next that languages in $\mathrm{R}(\Gamma)$ cannot be Γ-random. We will show in section 1.5 that most languages in E_2 are p-random and that most languages in E_2SPACE are pspace-random.

Proposition 1.28. If $A \in \mathrm{R}(\Gamma)$, then $\{A\}$ has Γ-measure 0 and thus A is not Γ-random.

Proof . Use martingale d, where $d(\lambda) = 1$ and for $w \in \{0,1\}^*$, $b \in \{0,1\}$

$$d(wb) = \begin{cases} 2 \cdot d(w) & \text{if } A(s_{|w|}) = b \\ 0 & \text{otherwise.} \end{cases}$$

■

Notice that every singleton set has Lebesgue-measure 0, so we cannot define in this way a Lebesgue concept of randomness for individual languages.

1.5 Some technical lemmas

In this section we develop some technical tools that will help us in the proofs that a given class has measure 0 or measure 1 in $\mathrm{R}(\Gamma)$.

The formulation and proof of these lemmas will be simplified by the use of 'Γ-approximable' martingales, in the place of martingales in Γ. For instance, the use of 'Γ-approximable' functions helps us to deal with an infinite sum of martingales in Γ. This sum may not be in Γ, but is Γ-approximable if the martingales have a uniform enumeration in Γ.

We start by proving that, for our purposes, using martingales that are 'Γ-approximable' is equivalent to using martingales in Γ, that is, if a class has measure 0 using a 'Γ-approximable' martingale then it has Γ-measure 0. Let us formalize our definition of Γ-approximable.

Notation. Given two sets X, Y, we consider each function $f\colon \mathbb{N}\times X \to Y$ as an enumeration of the functions f_k, $k \in \mathbb{N}$ where for each $k \in \mathbb{N}$, $f_k\colon X \to Y$ is defined as $f_k(x) = f(k,x)$ for every $x \in X$. In the same way we consider each function $f\colon \mathbb{N}^n \times X \to Y$ as an enumeration of the functions $f_{\vec{k}}$ for $\vec{k} \in \mathbb{N}^n$.

Definition 1.29. Let X be the cartesian product of a finite number of factors of the form $\mathbb{N}$ and $\{0,1\}^*$. A function $\widehat{f} \in \Gamma$, $\widehat{f}\colon \mathbb{N} \times X \to \mathbf{D}$ is a Γ-*computation* of a function $f\colon X \to [0,\infty)$ iff

$$|\widehat{f}_k(w) - f(w)| \leq 2^{-k}$$

for all $w \in X$ and $k \in \mathbb{N}$.

Definition 1.30. A function $f\colon X \to [0,\infty)$ is Γ-*computable* iff there exists a Γ-computation of f.

Notice that if f takes only values in $\mathbf{D}$ and $f \in \Gamma$ then f is trivially Γ-computable. This simple case will often happen in our applications.

Γ-computable martingales do not give additional measuring-power, as shown in the next lemma. The proof uses the techniques for p-computations developed by Lutz in [Lutz94a]. A similar result has been independently proven by Juedes and Lutz in [JuedLu94b], where they adopt the name Exact Computation Lemma.

Lemma 1.31. Exact Computation Lemma. For each Γ-computable martingale d there exists a martingale d' in Γ such that $\mathrm{S}^\infty[d] = \mathrm{S}^\infty[d']$.

Proof . Let $\widehat{d}$ be a Γ-computation of d. We define $c\colon \{0,1\}^* \to \mathbf{D}$ a function that is very close to d as $c(w) = \widehat{d}_{|w|}(w)$ for each $w \in \{0,1\}^*$. Since $\widehat{d}$ is a Γ-computation of d we have that $c \in \Gamma$ and $|d(w) - c(w)| \leq 2^{-|w|}$ for each $w \in \{0,1\}^*$.

We define recursively d' as follows

$$d'(\lambda) = c(\lambda) + 2$$

$$d'(wb) = d'(w) + \frac{c(wb) - c(w\bar{b})}{2}.$$

Since c is in Γ, then d' is also in Γ.

By Lemma 1.9, d' is a martingale and $\mathrm{S}^\infty[d] = \mathrm{S}^\infty[d']$, which finishes our proof. ■

Using the lemma we just proved, to see that a class X has Γ-measure 0 it will be enough to find a Γ-computable martingale d such that $X \subseteq \mathrm{S}^\infty[d]$. This will be useful mainly in the proofs of more sophisticated tools in this section.

Corollary 1.32. Let X be a class of languages. X has Γ-measure 0 if and only if there exists a Γ-computable martingale d such that $X \subseteq \mathrm{S}^\infty[d]$.

In Lebesgue measure, a countable union of measure 0 sets has measure 0. This additivity property is a useful tool when proving that a certain set has Lebesgue measure 0. Notice that a countable union of Γ-measure classes is not necessarily Γ-measure 0, because for each $A \in \mathrm{R}(\Gamma)$, $\{A\}$ has Γ-measure 0 and still $\mathrm{R}(\Gamma)$ does not have Γ-measure 0. Therefore, we want to find a weaker additivity property for Γ-measure that helps us to prove that some classes have measure 0 in $\mathrm{R}(\Gamma)$.

We find next a weak version of countable additivity that corresponds to the idea of uniformity and is useful in Γ-measure. This additivity notion implies uniform families of martingales that are called martingale systems.

Definition 1.33. An *n-dimensional martingale system* (n-MS) is a function

$$d\colon \mathbb{N}^n \times \{0,1\}^* \to [0,\infty)$$

such that $d_{\vec{k}}$ is a martingale for every $\vec{k} \in \mathbb{N}^n$.

We now define a restricted notion of countable union, that is called Γ-union. This concept is only defined for Γ-measure 0 sets.

Definition 1.34. A set X is a Γ-*union* of the Γ-measure 0 sets $X_0, X_1, X_2, \ldots$ iff

$$X = \bigcup_{j=0}^{\infty} X_j$$

and there exists a Γ-computable 1-MS d such that for every j, $X_j \subseteq \mathrm{S}^\infty[d_j]$.

Notice that by Corollary 1.32, each X_i in the definition has Γ-measure 0, because each d_j is Γ-computable.

Lemma 1.35. [Lutz92] Γ-*additivity Lemma.* If X is a Γ-union of Γ-measure 0 sets, then X has Γ-measure 0.

Proof . Let d be given by the definition of Γ-union. Let $\widehat{d}$ be a Γ-computation of d.

We first construct a Γ-computable 1-MS D such that

i) For all $j \in \mathbb{N}$, $\mathrm{S}^\infty[D_j] = \mathrm{S}^\infty[d_j]$.

ii) For all $j \in \mathbb{N}$, $D_j(\lambda) \leq 2^{-j}$.

For $j \in \mathbb{N}$, $w \in \{0,1\}^*$ we define

$$D_j(w) = 2^{\min\{0, -\log(\widehat{d}_{j,1}(\lambda)) - 2 - j\}} \cdot d_j(w)$$

that clearly fulfills i) and ii).

To see that D is Γ-computable we define $\widehat{D} \in \Gamma$ as follows. For $j, k \in \mathbb{N}$, $w \in \{0,1\}^*$

$$\widehat{D}_{j,k}(w) = 2^{\min\{0, -\log(\widehat{d}_{j,1}(\lambda)) - 2 - j\}} d_{j,k}(w).$$

For $j, k \in \mathbb{N}$, $w \in \{0,1\}^*$ we have that

$$|D_j(w) - \widehat{D}_{j,k}(w)| = 2^{\min\{0, -\log(\widehat{d}_{j,1}(\lambda)) - 2 - j\}} |d_j(w) - \widehat{d}_{j,k}(w)| \leq |d_j(w) - \widehat{d}_{j,k}(w)| \leq 2^{-k},$$

which shows that $\widehat{D}$ is a Γ-computation of D.

It is straightforward to show that condition i) above holds, since for each $j \in \mathbb{N}$, the function D_j is just d_j multiplied by a constant $c > 0$, thus for each $x \in \{0,1\}^\infty$, $\limsup_m D_j(x[0..m]) = c \cdot \limsup_m d_j(x[0..m])$, and $\mathrm{S}^\infty[D_j] = \mathrm{S}^\infty[d_j]$.

Since $d_j(\lambda) \leq \widehat{d}_{j,1}(\lambda) + 2^{-1}$ we have that

$$D_j(\lambda) = 2^{\min\{0, -\log(\widehat{d}_{j,1}(\lambda)) - 2 - j\}} d_j(\lambda) \leq 2^{-j},$$

and thus condition ii) above holds.

To prove that X has Γ-measure 0 we define the martingale $d' \colon \{0,1\}^* \to [0, \infty)$ by

$$d'(w) = \sum_{j=0}^{\infty} D_j(w).$$

d' is well defined because $D_j(\lambda) \leq 2^{-j}$ which implies that $d'(\lambda) < \infty$, and for other values $w \in \{0,1\}^*$ $d'(w) \leq \sum_{j=0}^\infty 2^{|w|} D_j(\lambda) \leq d'(\lambda) 2^{|w|}$. d' is trivially a martingale such that $X \subseteq \mathrm{S}^\infty[d']$, since $X_j \subseteq \mathrm{S}^\infty[D_j]$.

All that remains to be shown is that d' is Γ-computable. By Corollary 1.32 this will show that X has Γ-measure 0. Define a function $\widehat{d'}: \mathbb{N} \times \{0,1\}^* \to \mathbf{D}$ such that $\widehat{d'} \in \Gamma$ by

$$\widehat{d'_k}(w) = \sum_{j=0}^{k+|w|+1} \widehat{D}_{j,j+k+2}(w).$$

Let us see that $\widehat{d'}$ is a Γ-computation of d'. For each $k \in \mathbb{N}$, $w \in \{0,1\}^*$

$$|\widehat{d'_k}(w) - d'(w)| \leq \sum_{j=0}^{k+|w|+1} |\widehat{D}_{j,j+k+2}(w) - D_j(w)| \; + \sum_{j=k+|w|+2}^{\infty} D_j(w) \leq$$

$$\leq \sum_{j=0}^{k+|w|+1} 2^{-j-k-2} \; + \sum_{j=k+|w|+2}^{\infty} 2^{|w|} \cdot D_j(\lambda) \leq$$

$$\leq \sum_{j=0}^{\infty} 2^{-j-k-2} \; + 2^{|w|} \cdot 2^{-k-|w|-1} = 2^{-k-1} + 2^{-k-1} = 2^{-k}.$$

This completes the proof that X has Γ-measure 0. ■

As a corollary we have a similar result for measure in $\mathrm{R}(\Gamma)$.

Definition 1.36. A set X is a Γ-*union* of the measure 0 in $\mathrm{R}(\Gamma)$ sets $X_0, X_1, X_2, \ldots$ iff $X \cap \mathrm{R}(\Gamma)$ is a Γ-*union* of the Γ-measure 0 sets $X_0 \cap \mathrm{R}(\Gamma), X_1 \cap \mathrm{R}(\Gamma), X_2 \cap \mathrm{R}(\Gamma), \ldots$

Corollary 1.37. If X is a Γ-union of measure 0 in $\mathrm{R}(\Gamma)$ sets, then X has measure 0 in $\mathrm{R}(\Gamma)$.

With these Γ-additivity lemmas we can show now more elaborated results. We start with an easy example of application and then prove a number of interesting consequences.

Example 3

The class

$$X = \left\{ A \;\middle|\; \text{for almost every } n \in \mathbb{N},\ |A^{=n}| \geq \frac{2}{3} 2^n \right\}$$

has measure 0 in E.

Proof. We start by writing X as a countable union of classes. For each $i \in \mathbb{N}$ let

$$X_i = \left\{ A \;\middle|\; \text{for every } n \geq i,\ |A^{=n}| \geq \frac{2}{3} 2^n \right\}.$$

It is clear that $X = \bigcup_i X_i$.

We want to show that X has p-measure 0 by proving that X is a p-union of the measure 0 sets X_i and then using Lemma 1.35. Therefore we have to define a p-computable 1-MS d such that for each i, $X_i \subseteq \mathrm{S}^\infty[d_i]$.

Our definition of d is based in the martingale in Example 2. For each $i \in \mathbb{N}$, $d_i(\lambda) = 1$, and for each $w \in \{0,1\}^*$,

If $|w| \geq 2^i - 1$ then

$$d_i(w0) = \frac{1}{2} d_i(w) \qquad d_i(w1) = \frac{3}{2} d_i(w),$$

else, if $|w| < 2^i - 1$ then $d_i(w0) = d_i(w1) = d_i(w)$.

(We remind the reader that $d_i(w)$ denotes $d(i, w)$.)

For each $i \in \mathbb{N}$, d_i is a martingale that works as the one in Example 2 on inputs of length bigger than $2^i - 1$, that is, on bits corresponding to strings of length at least i. By the same reasoning of that example we can show that for each i, $X_i \subseteq \mathrm{S}^\infty[d_i]$.

To check whether $|w| < 2^i - 1$ we just need to write $|w|$ in binary and count the number of bits used, comparing it with i. Thus d can be computed in time linear in $|w| + i$, is trivially p-computable and X has p-measure 0. ■

The next Theorem has a number of interesting corollaries.

Theorem 1.38. Let Γ and Γ' be two measure resource-bounds such that Γ' contains a universal function for Γ, that is, there exist $f \in \Gamma'$ with $\Gamma = \{f_i \mid i \in \mathbb{N}\}$. Then the class $X = \bigcup_{\mu_\Gamma(Y)=0} Y$ has Γ'-measure 0.

Proof. Let Γ and Γ' be as in the hypothesis. Let $f \in \Gamma'$ be a universal function for Γ. We define $g\colon \mathbb{N} \times \{0,1\}^* \to \mathbf{D}$ as follows. For each $i \in \mathbb{N}$, $g_i(\lambda) = f_i(\lambda)$; for $w \in \{0,1\}^*$, and $b \in \{0,1\}$,

$$g_i(wb) = \begin{cases} f_i(wb) & \text{if } g_i(w) = f_i(w) \text{ and } \frac{f_i(w0)+f_i(w1)}{2} = f_i(w) \\ g_i(w) & \text{otherwise.} \end{cases}$$

Notice that if f_i is a martingale then $f_i \equiv g_i$, and that for every i, g_i is a martingale. It is also clear that $g \in \Gamma'$. Then g is a 1-MS in Γ', thus trivially Γ'-computable.

By Lemma 1.35, $\bigcup_i \mathrm{S}^\infty[g_i]$ has Γ'-measure 0. We finish the proof by seeing that $X \subseteq \bigcup_i \mathrm{S}^\infty[g_i]$.

Let Y be such that $\mu_\Gamma(Y) = 0$, there exists a martingale d in Γ such that $Y \subseteq \mathrm{S}^\infty[d]$. Since f is universal for Γ, $d \equiv f_i$ for some i. Thus f_i is a martingale and $f_i \equiv g_i$, which implies $Y \subseteq \bigcup_i \mathrm{S}^\infty[g_i]$. ■

We can show now that Γ is in a sense the largest measure resource-bound such that $\mathrm{R}(\Gamma)$ does not have Γ-measure 0.

Corollary 1.39. Let Γ and Γ' be two measure resource-bounds such that Γ' contains a universal function for Γ. Then $\mathrm{R}(\Gamma)$ has Γ'-measure 0.

Proof. From Proposition 1.28 we know that for each $L \in \mathrm{R}(\Gamma)$, $\{L\}$ has Γ-measure 0. Thus this is a direct consequence of the last theorem. ■

Corollary 1.40. The class

$$X = \bigcup_{\mu_{\mathrm{p}}(Y)=0} Y$$

has p_2-measure 0. The class

$$X = \bigcup_{\mu_{\mathrm{pspace}}(Y)=0} Y$$

has $\mathrm{p}_2$space-measure 0.

Corollary 1.41. E has measure 0 in E_2. ESPACE has measure 0 in E_2SPACE. The class of p-random languages has measure 1 in E_2. The class of pspace-random languages has measure 1 in E_2SPACE.

Proposition 1.42. For every $c > 0$,

$$\mu(\mathrm{DTIME}(2^{cn}) \mid \mathrm{E}) = 0$$

and

$$\mu(\mathrm{DSPACE}(2^{cn}) \mid \mathrm{ESPACE}) = 0.$$

Proof . We show the first part, the second part being analogous.

Let $c > 0$. Let $\{M_i \mid i \in \mathbb{N}\}$ be a recursive enumeration of the Turing Machines that work in time 2^{cn}. We can assume that the enumeration is efficient, and for each $i \in \mathbb{N}$, $x \in \{0,1\}^*$ we can compute $M_i(x)$ in time $2^{c|x|} \cdot i$.

For each $i \in \mathbb{N}$ we define $X_i = \{L(M_i)\}$, the class containing only the language $L(M_i)$. Then $\mathrm{DTIME}(2^{cn}) = \bigcup_i X_i$. Let us see that $\mathrm{DTIME}(2^{cn})$ is a p-union of the p-measure 0 classes X_i.

Let d be the following. For each $i \in \mathbb{N}$, $d_i(\lambda) = 1$. For $w \in \{0,1\}^*$, $b \in \{0,1\}$

$$d_i(wb) = \begin{cases} 2 \cdot d_i(w) & \text{if } M_i(s_{|w|}) = b \\ 0 & \text{otherwise.} \end{cases}$$

We leave to the reader to see that d is the 1-MS in p that witnesses the result. ■

The classical first Borel-Cantelli Lemma deals with countable families of sets $\{X_n\}$ such that $\Pr(X_n)$ decreases very quickly and goes to 0 in the limit. For those families the lemma states that the set of all x that belong to X_n for infinite many n's has Lebesgue measure 0. The exact formulation follows:

Lemma 1.43. Classical first Borel-Cantelli Lemma. Let $\{X_j \subseteq \{0,1\}^\infty \mid j \in \mathbb{N}\}$ be a sequence of Lebesgue-measurable sets such that

$$\sum_{j=0}^{\infty} \Pr(X_j)$$

is convergent, then

$$\Pr\left(\bigcap_{t=0}^{\infty}\bigcup_{j=t}^{\infty} X_j\right) = 0.$$

Notice that the class

$$\bigcap_{t=0}^{\infty}\bigcup_{j=t}^{\infty} X_j$$

consists exactly of those x that belong to X_n for an infinite number of n.

We are interested in classes of languages that can be represented with this kind of expressions. To study their measure we want an appropriate resource-bounded formulation of the Borel-Cantelli Lemma. Since countable unions of Γ-measure 0 sets are not always measure 0, it will be useful to have a more elaborated property that for each family $\{X_{i,j} \mid i, j \in \mathbb{N}\}$ fulfilling certain restrictions derives a consequence for

$$\bigcup_{i=0}^{\infty}\bigcap_{t=0}^{\infty}\bigcup_{j=t}^{\infty} X_{i,j}.$$

For a translation of the classical Borel-Cantelli Lemma to Γ-measure we need a resource-bounded version of the idea of a family of classes with Lebesgue measure decreasing quickly to 0. To do this we introduce a way of saying 'X has Γ-measure smaller than μ', for a class X and $\mu > 0$.

For each martingale d and $r > 0$, we define the set $\mathrm{S}^r[d]$ with those languages for which d succeeds in multiplying by at least r the starting capital $d(\lambda)$. We interpret $X \subseteq \mathrm{S}^r[d]$ as 'X has Γ-measure smaller than $1/r$'.

Definition 1.44. Let d be a martingale, and $r > 0$. We define the class

$$\mathrm{S}^r[d] = \{A \mid \lim_{m\to\infty} d(A[0..m]) \geq d(\lambda)\cdot r\}.$$

Notice that if X is a class that is closed under finite variations,

$$X = \bigcap_{t=0}^{\infty}\bigcup_{j=t}^{\infty} X_j,$$

the corresponding X_j are not necessarily closed under finite variations, thus the 0-1 law stating that Γ-measurable classes that are closed under finite variations can only have Γ-measure 0 and Γ-measure 1 does not apply to them. For instance let X be the class of A such that for infinitely many n, $A^{=n}$ has less that 2^{n-2} strings; if for each n we define $X_n = \{A \mid |A^{=n}| < 2^{n-2}\}$ then

$$X = \bigcap_{t=0}^{\infty}\bigcup_{n=t}^{\infty} X_n,$$

and each X_n is not closed under finite variations.

For our resource-bounded version of the Borel-Cantelli Lemma we also need to substitute in the classical formulation the usual convergence of series by a more restrictive notion.

Definition 1.45. Let $\{a_n \mid n \in \mathbb{N}\}$ be a sequence of nonnegative real numbers. A *modulus* for the series $\sum_{n=0}^{\infty} a_n$ is a function $m\colon \mathbb{N} \to \mathbb{N}$ such that

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

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

Definition 1.46. Let $\{a_{j,k} \mid j,k \in \mathbb{N}\}$ be a sequence of nonnegative real numbers. A sequence

$$\sum_{k=0}^{\infty} a_{j,k} \qquad (j = 0, 1, 2, \ldots)$$

of series is *uniformly* Γ-*convergent* if there exists a function $m\colon \mathbb{N}^2 \to \mathbb{N}$ such that $m \in \Gamma$ and for each $j \in \mathbb{N}$, m_j is a modulus for the series $\sum_{k=0}^{\infty} a_{j,k}$.

Finally, we state the following uniform, resource-bounded generalization of the classical first Borel-Cantelli Lemma that will greatly simplify the proof of several measure results in the following chapters.

Lemma 1.47. *[Lutz92].* Let $\{X_{i,j} \subseteq \{0,1\}^\infty \mid i,j \in \mathbb{N}\}$ be a sequence of classes. If there exists d a Γ-computable 2-MS such that

(i) $\forall i,j \in \mathbb{N}$, $X_{i,j} \subseteq \mathrm{S}^{\frac{1}{d_{i,j}}(\lambda)}[d_{i,j}]$. and

(ii) the series

$$\sum_{j=0}^{\infty} d_{i,j}(\lambda) \qquad (i = 0, 1, 2, \ldots)$$

are uniformly Γ-convergent,

then

$$\mu_\Gamma \left(\bigcup_{i=0}^{\infty} \bigcap_{t=0}^{\infty} \bigcup_{j=t}^{\infty} X_{i,j} \right) = 0.$$

Proof of Lemma 1.47. Assume the hypothesis. Fix a function $m\colon \mathbb{N}^2 \to \mathbb{N}$ witnessing that the series

$$\sum_{j=0}^{\infty} d_{i,j}(\lambda) \qquad (i = 0, 1, 2, \ldots)$$

are uniformly Γ-convergent. Without loss of generality, assume that m_i is nondecreasing and $m_i(n) \geq 2$ for all $i, n \in \mathbb{N}$. Define

$$Y_i = \bigcap_{t=0}^{\infty} \bigcup_{j=t}^{\infty} X_{i,j},$$

$$Y = \bigcup_{i=0}^{\infty} Y_i.$$

Our task is to prove that $\mu_\Gamma(Y) = 0$. For this we will use the Γ-additivity Lemma (Lemma 1.35) defining $d' \colon \mathbb{N} \times \{0,1\}^* \to [0, \infty)$ by

$$d'(i, w) = \sum_{j=m_i(i)}^{\infty} d_{i,j}(w)$$

for all $i \in \mathbb{N}$, $w \in \{0,1\}^*$. We show that d' testifies that Y is a Γ-union of the Γ-measure 0 sets $Y_0, Y_1, Y_2, \ldots$, whence $\mu_\Gamma(Y) = 0$ by the Γ-additivity Lemma.

Each d'_i is trivially a martingale, so d' is a 1-MS. We want to see that $Y_i \subseteq \mathrm{S}^\infty[d'_i]$. Fix $i \in \mathbb{N}$, $x \in Y_i$. Since $X_{i,j} \subseteq \mathrm{S}^{\frac{1}{d_{i,j}}(\lambda)}[d_{i,j}]$ for every $j \in \mathbb{N}$ and there are infinitely many j for which $x \in X_{i,j}$, then for each $k \in \mathbb{N}$ there exist $n \in \mathbb{N}$, $j_1, \ldots, j_{k+1}$ bigger than $m_i(i)$, with $d_{j_r}(x[0..n]) > 1 - \frac{1}{k+1}$ for $r = 1, \ldots, k+1$, and

$$d'_i(x[0..n]) = \sum_{j=m_i(i)}^{\infty} d_{i,j}(x[0..n]) > k.$$

We conclude that $\limsup_n d'_i(x[0..n]) = \infty$ and $x \in \mathrm{S}^\infty[d'_i]$.

Next we have to show that d' is Γ-computable. For this we use the Γ-convergence of the cited series. Let $\tilde{d} \colon \mathbb{N}^3 \times \{0,1\}^* \to \mathbf{D}$ be a Γ-computation of d. We define $\widehat{d} \colon \mathbb{N}^2 \times \{0,1\}^* \to \mathbf{D}$ by

$$\widehat{d}(i, n, w) = \sum_{j=0}^{m_i(n+|w|+1)-1} \tilde{d}_{i,j,n+2+j}(w)$$

for all $i, n \in \mathbb{N}$, $w \in \{0,1\}^*$. Let's see that $\widehat{d}$ is a Γ-computation of d'.

$$|\widehat{d}(i, n, w) - d'(i, w)| \leq$$

$$\leq \sum_{j=0}^{m_i(n+|w|+1)-1} |d(i, j, w) - \tilde{d}(i, j, n+2+j, w)| + \sum_{j=m_i(n+|w|+1)}^{\infty} |\widehat{d}(i, j, w)| \leq$$

$$\leq 2^{-n-1} + 2^{-n-1} = 2^{-n}.$$

■

In order to take full advantage of this lemma we will use the following sufficient condition for uniform Γ-convergence. (This well-known lemma is easily verified by routine calculus, remarking that polynomials can be computed in Γ, for Γ any measure resource-bound.)

Lemma 1.48. Let $a_{j,k} \in [0, \infty)$ for all $j, k \in \mathbb{N}$. If there exist a real $\varepsilon > 0$ and a polynomial $g\colon \mathbb{N} \to \mathbb{N}$ such that $a_{j,k} \leq e^{-k^{\varepsilon}}$ for all $j, k \in \mathbb{N}$ with $k \geq g(j)$, then the series

$$\sum_{k=0}^{\infty} a_{j,k} \qquad (j = 0, 1, 2, \ldots)$$

are uniformly Γ-convergent.

We finish this chapter with an application of the resource-bounded Borel-Cantelli Lemma (Lemma 1.47).

Example 4 The class

$$X = \left\{ A \mid |A^{=n}| \geq 2^n \left(\frac{1}{2} + \frac{1}{n} \right) \text{ for infinitely many } n \right\}$$

has measure 0 in E.

Proof . For each $n \in \mathbb{N}$, let

$$X_{2^n} = \left\{ A \mid |A^{=n}| \geq 2^n \left(\frac{1}{2} + \frac{1}{n} \right) \right\},$$

and let $X_j = \emptyset$ if j is not a power of 2. Then by definition of X,

$$X = \bigcap_{t=0}^{\infty} \bigcup_{j=t}^{\infty} X_j.$$

We want to apply Lemma 1.47 to this expression of X. Notice that we do not have the outermost union. It is enough to define a 1-MS d such that for each j, $X_j \subseteq \mathrm{S}^{\frac{1}{d_j(\lambda)}}[d_j]$. For each $j \in \mathbb{N}$ such that j is a power of 2, $w \in \{0,1\}^*$ let

$$d(j, w) = \Pr_x[x \in X_j \mid x \in \mathbf{C}_w],$$

for the rest of j let $d_j \equiv 2^{-j}$.

By definition of conditional probability, d is a 1-MS. We have to show that d is p-computable and that conditions (i) and (ii) in Lemma 1.47 hold.

To see that condition (i) holds, fix $j \in \mathbb{N}$ a power of 2 and $x \in X_j$. Since the condition $x \in X_j$ is only based on the prefix $x[0..2j-2]$, any $y \in \{0,1\}^{\infty}$ such that $y \in \mathbf{C}_{x[0..2j-2]}$ is also in X_j, and $d(j, x[0..2j-2]) = 1$, thus $x \in \mathrm{S}^{\frac{1}{d_j(\lambda)}}[d_j]$.

To see that condition (ii) holds, we have to look at the series

$$\sum_{j=0}^{\infty} d_j(\lambda) = \sum_{j=0}^{\infty} \Pr_x[x \in X_j].$$

For each $n \in \mathbb{N}$,

$$\Pr_x[x \in X_{2^n}] = \sum_{i=\lceil 2^n(1/2+1/n)\rceil}^{2^n} \binom{2^n}{i} = \sum_{i=0}^{2^n(1/2+1/n)} \binom{2^n}{i}$$

The Chernoff bound (Lemma 1.1) tells us that, if $j = 2^n$

$$\Pr_x[x \in X_j] \leq e^{-\frac{j}{6n^2}},$$

thus there exists $c > o$ such that if $j > c$ then

$$\Pr_x[x \in X_j] \leq e^{-j^{0.5}}.$$

This, together with Lemma 1.48, tells us that $\sum\limits_{j=0}^{\infty} d_j(\lambda)$ is p-convergent and condition (ii) holds.

We need to check that d is p-computable. We can use binomial coefficients to exactly compute $\Pr_x[x \in X_j \mid x \in \mathbf{C}_w]$ in time polynomial in $|w| + j$, thus $d \in \mathrm{p}$ and we have the result. ■

1.6 Γ-measurability and the Kolmogorov 0-1 law

In this section we develop the concept of Γ-measurability and consider classes that have Γ-measure μ, for μ any value between 0 and 1. Then we prove that these new concepts are not useful for classes that are closed under finite variations, for instance most of the classes defined in Structural Complexity. This is stated as the resource-bounded version of the Kolmogorov 0-1 law, which is a consequence of the classical Kolmogorov 0-1 law.

The material in this section has been included for the sake of completeness. The whole section can be skipped without losing continuity with the rest of the chapters.

In order to define Γ-measurability, we start by defining a function μ^*_Γ that associates to each class of languages X the set of upper bounds of its possible Γ-measure. A class X will be Γ-measurable when one of these bounds is tight.

Definition 1.49. Let $\mu^*_\Gamma \colon \{0,1\}^\infty \to \mathcal{P}([0,1])$ be the function that for each $X \in \{0,1\}^\infty$ is defined as follows

$$\mu^*_\Gamma(X) = \left\{\gamma \geq 0 \;\middle|\; \text{there exists a 1-MS } d \in \Gamma \text{ such that for each } k \in \mathbb{N}, X \subseteq \mathrm{S}^{\frac{1}{\gamma+2^{-k}}}[d_k]\right\}.$$

Remember from section 1.5 that for each martingale d and $r > 0$, the class $\mathrm{S}^r[d]$ contains those languages for which d succeeds in multiplying by at least r the starting capital $d(\lambda)$.

The next lemma states some basic properties of μ^*_Γ. Since this section is not critical, we prefer to state strictly the properties needed in the proof of Theorem 1.54.

Lemma 1.50.

(i) If $X \subseteq Y$ then $\mu^*_\Gamma(Y) \subseteq \mu^*_\Gamma(X)$.

(ii) Let $\{Y_1, \ldots, Y_n\}$ be a finite sequence of pair-wise disjoint classes. If $\gamma_i \in \mu^*_\Gamma(Y_i)$ for each i then

$$\sum_{i=1}^{n} \gamma_i \in \mu^*_\Gamma\Big(\bigcup_{i=1}^{n} Y_i\Big).$$

(iii) For very X, $\mu^*_\Gamma(X) + \mu^*_\Gamma(X^c) \geq 1$.

(iv) Let $\{Y_i \mid i \in \mathbb{N}\}$ be a sequence of pair-wise disjoint classes. If $\gamma_i \in \mu^*_{\mathbf{all}}(Y_i)$ for each i then

$$\sum_{i \in \mathbb{N}} \gamma_i \in \mu^*_{\mathbf{all}}\Big(\bigcup_{i \in \mathbb{N}} Y_i\Big).$$

(v) If $\{a_n \mid n \in \mathbb{N}\}$ is a decreasing sequence such that $\{a_n \mid n \in \mathbb{N}\} \subseteq \mu^*_{\mathbf{all}}(X)$ and $l = \lim_{n\to\infty} a_n$ then $l \in \mu^*_{\mathbf{all}}(X)$.

Proof . Part (i) is straight-forward from the definition. For (ii), let $\{Y_1, \ldots, Y_n\}$ be as in the hypothesis. For each $i \in \mathbb{N}$, let $\gamma_i \in \mu^*_\Gamma(Y_i)$. All we need to do is to define a 1-MS d such that for each $k \in \mathbb{N}$, $\bigcup_i Y_i \subseteq \mathrm{S}^{\frac{1}{\Sigma_i \gamma_i + 2^{-k}}}[d_k]$.

For each $i \in \mathbb{N}$, let d^i be a 1-MS in Γ such that for each k $Y_i \subseteq \mathrm{S}^{\frac{1}{\gamma_i + 2^{-k}}}[d^i_k]$. For each $w \in \{0,1\}^*$, $k \in \mathbb{N}$,

$$d_k(w) = \sum_i \frac{\gamma_i + 2^{-k-\log n}}{d^i_{k+\log n}(\lambda)} d^i_{k+\log n}(w).$$

By definition $d_k(\lambda) = \sum_i \gamma_i + 2^{-k}$, and for each i, $Y_i \subseteq \mathrm{S}^{\frac{1}{d_k(\lambda)}}[d_k]$.

For (iii), assume that there exist $\gamma \in \mu^*_\Gamma(X)$, $\gamma' \in \mu^*_\Gamma(X^c)$ such that $\gamma + \gamma' < 1$. Then by (ii) $\gamma + \gamma' \in \mu^*_\Gamma(\{0,1\}^\infty)$, and for each $k \in \mathbb{N}$ there exists a martingale d_k such that $\{0,1\}^\infty \subseteq \mathrm{S}^{\frac{1}{\gamma + \gamma' + 2^{-k}}}$, thus there exists a martingale d' and $r > 1$ such that $\{0,1\}^\infty \subseteq \mathrm{S}^r[d']$. But this contradicts property (1.1) in the definition of martingale.

For (iv), let $\{Y_i \mid i \in \mathbb{N}\}$ be as in the hypothesis. For each $i \in \mathbb{N}$, let $\gamma_i \in \mu^*_{\mathbf{all}}(Y_i)$. We define a 1-MS d such that for each $k \in \mathbb{N}$, $\bigcup_i Y_i \subseteq \mathrm{S}^{\frac{1}{\Sigma_i \gamma_i + 2^{-k}}}[d_k]$.

For each $i \in \mathbb{N}$, let d^i be a 1-MS such that for every k, $Y_i \subseteq \mathrm{S}^{\frac{1}{\gamma_i + 2^{-k}}}[d^i_k]$. For each $w \in \{0,1\}^*$, $k \in \mathbb{N}$

$$d_k(w) = \sum_i \frac{\gamma_i + 2^{-i-k}}{d^i_{k+i}(\lambda)} d^i_{k+i}(w).$$

By definition $d_k(\lambda) = 2^{-k} + \sum_i \gamma_i$, and for each i, $Y_i \subseteq \mathrm{S}^{\frac{1}{d_k(\lambda)}}[d_k]$.

Part (v). For each n let d^n be a 1-MS such that for each k, $X \subseteq \mathrm{S}^{\frac{1}{a_n+2^{-k}}}[d^n_k]$. We construct d' such that $X \subseteq \mathrm{S}^{\frac{1}{l+2^{-k}}}[d'_k]$ as follows. For each $k \in \mathbb{N}$, let n_k be such that for each $n \geq n_k$, $a_n - l \leq 2^{-2k}$. Let $d'_k \equiv d^{n_k}_{2k}$. Then for each $x \in X$,

$$\lim_{m\to\infty} d'_k(x[0..m]) = \lim_{m\to\infty} d^{n_k}_{2k}(x[0..m]) \geq \frac{1}{2^{-2k}+a_{n_k}} \cdot d'_k(\lambda) \geq \frac{1}{2^{-k}+l} \cdot d'_k(\lambda),$$

and $l \in \mu^*_{\mathbf{all}}(X)$. ■

From μ^*_Γ we define Γ-measurability.

Definition 1.51. Let X be a class of languages. We say that X is Γ-measurable if there exists $\gamma \in \mu^*_\Gamma(X)$, $\gamma' \in \mu^*_\Gamma(X^c)$ such that $\gamma + \gamma' = 1$.

Notice that if X is Γ-measurable, then by part (iii) of Lemma 1.50 there exist a unique $\gamma \in \mu^*_\Gamma(X)$ and a unique $\gamma' \in \mu^*_\Gamma(X^c)$ such that $\gamma + \gamma' = 1$. We denote $\mu_\Gamma(X) = \gamma$.

For $\Gamma = \mathbf{all}$ this definition corresponds to classical Lebesgue-measurability.

Remark that if X is Γ-measurable then X^c is also Γ-measurable.

We have the following elemental property

Lemma 1.52. Let Γ and Γ' be two measure resource-bounds such that $\Gamma \subseteq \Gamma'$. If X is Γ-measurable then X is Γ'-measurable and $\mu_\Gamma(X) = \mu_{\Gamma'}(X)$.

Proof. By Definition 1.49, $\mu^*_\Gamma(X) \subseteq \mu^*_{\Gamma'}(X)$, and $\mu^*_\Gamma(X^c) \subseteq \mu^*_{\Gamma'}(X^c)$, Thus if $\gamma + \gamma' = 1$ for some $\gamma \in \mu^*_\Gamma(X)$, $\gamma' \in \mu^*_\Gamma(X^c)$, X is Γ'-measurable. ■

Let us see that this definition is consistent with our definitions of Γ-measure 0 and Γ-measure 1 sets in section 1.4. Remember that in those definitions we used upper-limits instead of limits.

Lemma 1.53. X has Γ-measure 0 if and only if X is Γ-measurable and $\mu_\Gamma(X) = 0$.

Proof. Let X be a class that has Γ-measure 0, let $d \in \Gamma$ such that $X \subseteq \mathrm{S}^\infty[d]$.

By the definition of Γ-measurable, it is enough to see that there exists a 1-MS $d' \in \Gamma$ such that for every $k \in \mathbb{N}$, $\mathrm{S}^\infty[d] \subseteq \mathrm{S}^{2^k}[d'_k]$.

Fix $k \in \mathbb{N}$. d'_k is defined as follows

$$d'_k(w) = \begin{cases} d(w) & \text{if for every } i < |w|,\ d(w[0..i]) < 2^k \cdot d(\lambda), \\ d(w[0..i]) & \text{otherwise, for } i < |w| \text{, the first such that } d(w[0..i]) \geq 2^k \cdot d(\lambda). \end{cases}$$

Let $x \in \mathrm{S}^\infty[d]$. Since $\limsup_n d'(x[0..n]) = \infty$, for each $k \in \mathbb{N}$ there exists $n \in \mathbb{N}$ such that $d(w[0..n]) \geq 2^k \cdot d(\lambda)$. Let n_0 be the first such n. Then by definition of d'_k, $d'_k(x[0..n]) \geq 2^k \cdot d'_k(\lambda)$ for every $n \geq n_0$, and $\lim_n d'_k(x[0..n]) \geq 2^k \cdot d'_k(\lambda)$. Thus $x \in \mathrm{S}^{2^k}[d'_k]$, which finishes the first part.

Let X be a class that is Γ-measurable with $\mu_\Gamma(X) = 0$. There exists a 1-MS d such that for each k, $X \subseteq \mathrm{S}^{2^k}[d_k]$.

Let d' be the following martingale $d'(w) = \sum_k \frac{2^{-k}}{d_k(\lambda)} d_k(w)$. Then $X \subseteq \mathrm{S}^{\infty}[d']$ and d' is Γ-approximable by $\widehat{d}(i,w) = \sum_{k=1}^{i+|w|} \frac{2^{-k}}{d_k(\lambda)} d_k(w)$, thus X has Γ-measure 0 by Lemma 1.31. ■

Finally we state the resource-bounded version of the Kolmogorov 0-1 law. We argue that it immediately follows from the Kolmogorov 0-1 law for Lebesgue measure, and, for the sake of completeness, we give a proof of it in terms of martingales. A classical classical proof, using the definition by covers of Lebesgue measure can be found for instance in [Oxto].

Theorem 1.54. Let X be a class of languages that is closed under finite variations. If X is Γ-measurable then either X has Γ-measure 0 or X has Γ-measure 1.

Proof .

A key observation is that, by Lemma 1.52, if X is Γ-measurable, then X is **all**-measurable and $\mu_\Gamma(X) = \mu_{\mathbf{all}}(X)$. Therefore we just need to prove this property for $\Gamma = \mathbf{all}$, that is, for Lebesgue measure.

For each set of strings $U \subseteq \{0,1\}^*$, we denote as $\mathbf{C}_U$ the class of languages $\bigcup\limits_{y \in U} \mathbf{C}_y$.

This proof is based on the following claim:

Claim. Let X be a class of languages that is closed under finite variations. For each class Y, if $\gamma \in \mu^*_{\mathbf{all}}(X)$ and $\gamma' \in \mu^*_{\mathbf{all}}(X^c)$ then

$$\gamma \cdot \gamma' \in \mu^*_{\mathbf{all}}(X \cap Y).$$

Assuming that the claim is true, take $Y = X$. We have that $\mu_{\mathbf{all}}(X)^2 \in \mu^*_{\mathbf{all}}(X)$, which implies that $\mu_{\mathbf{all}}(X)$ can only be 0 or 1. Thus if X is measurable then $\mu_{\mathbf{all}}(X)$ must be either 0 or 1.

We finish this proof with the cited claim.

Proof of Claim. Let X be as in the hypothesis. Let $\gamma \in \mu^*_{\mathbf{all}}(X)$.

Consider the case $Y = \mathbf{C}_y$, for $y \in \{0,1\}^*$. It is an easy exercise to see that $\mu_{\mathbf{all}}(\mathbf{C}_w) = 2^{-|w|}$. Let d be a 1-MS such that $X \subseteq \mathrm{S}^{\frac{1}{\gamma+2^{-k}}}[d_k]$ for each k. Define a 1-MS d' as follows

$$d'_k(y) = \begin{cases} 0 & \text{if } w \not\sqsubseteq y \text{ and } y \not\sqsubseteq w \\ \sum\limits_{|z|=|y|} d_k(z) & \text{if } w = yt \\ \sum\limits_{|z|=|w|} d_k(zt) & \text{if } y = wt. \end{cases}$$

Then for each $x \in \mathbf{C}_w$, $d'_k(x[0..n]) = \sum\limits_{|z|=|w|} d_k(zx[|w|..n])$.

Let $x \in X \cap \mathbf{C}_w$. For each z with $|z| = |w|$, since X is closed under finite variations we know that $zy \in X$, where $y[i] = x[i+|w|]$ for every $i \geq 0$. Since $X \subseteq \mathrm{S}^{\frac{1}{\gamma+2^{-k}}}[d_k]$, then $\lim_n d_k(zy[0..n]) \geq \frac{d_k(\lambda)}{\gamma+2^{-k}}$. Thus $\lim_n d'_k(x[0..n]) \geq 2^{|w|}\frac{d_k(\lambda)}{\gamma+2^{-k}} = \frac{d_k(\lambda)}{2^{-|w|}(\gamma+2^{-k})}$. This shows that

$$\gamma \cdot \mu_{\mathbf{all}}(\mathbf{C}_w) \in \mu^*_{\mathbf{all}}(X \cap Y).$$

Now we consider the case $Y = \mathbf{C}_U$, where U is a prefix code (that is, U is a set of strings such that there are no $x, y \in U$ with $x \subsetneqq y$). We leave to the reader to see that $\mu_{\mathbf{all}}(\mathbf{C}_U) = \sum\limits_{w \in U} 2^{-|w|}$.

For each pair of strings $x, y \in U$ with $x \neq y$, $\mathbf{C}_x \cap \mathbf{C}_y = \emptyset$. By Lemma 1.50, if $\gamma_w \in \mu^*_{\mathbf{all}}(X\mathbf{C}_w)$ then $\sum\limits_{w \in U} \gamma_w \in \mu^*_{\mathbf{all}}(X \cap \mathbf{C}_U)$. Using the case $Y = \mathbf{C}_w$ we have that if $\gamma \in \mu^*_{\mathbf{all}}(X)$ then $\gamma \cdot \mu_{\mathbf{all}}(\mathbf{C}_U) \in \mu^*_{\mathbf{all}}(X \cap \mathbf{C}_U)$.

For the general case, let $\gamma' \in \mu^*_{\mathbf{all}}(Y)$. Let d be a 1-MS such that $Y \subseteq \mathrm{S}^{\frac{1}{\gamma' + 2^{-k}}}[d_k]$. Let

$$U_k = \left\{ w \;\middle|\; d_k(w) \geq \frac{d_k(\lambda)}{\gamma' + 2^{-k-1}} \text{ and for every } y \text{ with } y \subsetneqq w,\, d_k(w) < \frac{d_k(\lambda)}{\gamma' + 2^{-k-1}} \right\}.$$

Then U_k is a prefix code, $\gamma' + 2^{-k-1} \in \mu^*_{\mathbf{all}}(\mathbf{C}_{U_k})$ and $Y \subseteq \mathbf{C}_{U_k}$. Therefore,

$$\mu_{\mathbf{all}}(X) \cdot (\gamma' + 2^{-k-1}) \in \mu^*_{\mathbf{all}}(X \cap \mathbf{C}_{U_k}) \subseteq \mu^*_{\mathbf{all}}(X \cap Y).$$

Since this holds for every k, by Lemma 1.50 (v), $\mu_{\mathbf{all}}(X) \cdot \gamma' \in \mu^*_{\mathbf{all}}(X \cap Y)$ and we have finished the general case. ∎

Chapter 2: Measuring in PSPACE

2.1 Introduction

In Chapter 1 we have defined Lutz's resource-bounded measure for classes such as E, E_2, ESPACE and E_2SPACE by bounding resources in a constructive definition of Lebesgue measure, in this way the more restrictive is the resource-bound, the smaller is the class in which we have defined a nontrivial measure. The use of characteristic sequences in the definition of measure causes that, for instance if we impose a time bound $F(n)$ on Lebesgue measure, we obtain a meaningful measure in the class defined by the time bound $F(2^n)$. If we impose a linear time bound on Lebesgue measure we already reach the class of exponential time languages. Since dealing with sublinear bounds requires a more careful consideration, all of the classes for which we had the general definition of Lutz's measure contain E as a subclass.

However, there are interesting problems that can be formulated in terms of estimating the size of subclasses of P or PSPACE. For instance, we want to know whether most languages in P are efficiently parallelizable, or whether self-reducibility is a typical property for the languages in PSPACE. In this chapter we are interested in extending Lutz's measure to the class PSPACE of languages recognizable in polynomial space.

To do this, we explore the property that has endowed the mentioned exponential classes with a non-trivial measure structure, that is, the Measure Conservation Theorem in Chapter 1. We want to define a non-trivial measure inside PSPACE by looking for the same property. First we see that the natural candidate is not valid unless PSPACE $= E_2$, and then we get a valid formulation of a measure inside PSPACE.

There is another property of measure in exponential classes, namely Γ-additivity, that it is interesting to have for any measure. We prove this property for our measure in PSPACE and then use it to show that a class of self-reducible languages has measure 0 in PSPACE.

It remains open how to measure in P, which will probably require a different approach not dealing with characteristic sequences. This is because as in polynomial space we could not store a characteristic sequence, in polynomial time we cannot even compute it*.

The results in section 2.2 appear in [Mayo92b]. The results of section 2.3 are as yet unpublished.

2.2 Measure in PSPACE

* While revising the draft of this text, we have been informed that Allender and Strauss have obtained a reasonable definition of measure for P [AlleSt].

Our definition of resource bounded measure in Chapter 1 was restricted to classes of the form R(Γ), for Γ a measure resource-bound. In particular, since each measure resource-bound contains p, $\mathrm{E} \subseteq \mathrm{R}(\Gamma)$. The definition of measure in R(Γ) was based in Γ-measure, defined as a 'Γ-restriction' of Lebesgue-measure. The reason why R(Γ) inherits a non-trivial measure from this Γ-measure is that, as shown in Lutz's Measure Conservation Theorem (Theorem 1.18), R(Γ) does *not* have Γ-measure 0.

Following the same idea, in order to define a measure inside PSPACE we have to find solutions to the equation $\mathrm{R}(\Gamma) = \mathrm{PSPACE}$ and check that the corresponding Γ-measure fulfills the Measure Conservation Theorem, that is, PSPACE does not have Γ-measure 0. This time we cannot require Γ to be a measure resource-bound, and through this chapter we will use Γ to denote any class of functions inside rec.

Let us have a look at the solutions of the equations $\mathrm{R}(\Gamma) = \mathrm{DTIME}(\mathcal{F})$ and $\mathrm{R}(\Gamma) = \mathrm{DSPACE}(\mathcal{F})$ for different families $\mathcal{F}$ that we used in Chapter 1. For example for $\mathrm{E} = \bigcup_c \mathrm{DTIME}(2^{cn})$ the solution was $\Gamma = \mathrm{p}$ and for $\mathrm{ESPACE} = \bigcup_c \mathrm{DSPACE}(2^{cn})$, $\Gamma = \mathrm{pspace}$. In all cases, for $\mathrm{R}(\Gamma) = \mathrm{DTIME}(\mathcal{F})$ we use the solution $\Gamma = \mathrm{DTIMEF}(\mathcal{F} \circ \log)$, and for $\mathrm{R}(\Gamma) = \mathrm{DSPACE}(\mathcal{F})$, $\Gamma = \mathrm{DSPACEF}(\mathcal{F} \circ \log)$.

By analogy, the class of polylogarithmic space computable functions is the natural candidate to define a measure in PSPACE, and we would like to prove that R(polylogspace) = PSPACE. We bound polylogarithmically only the working space, and do not pose any restriction on the output space in this case.

The following lemma proves the first part of this equality.

Lemma 2.1. PSPACE ⊆ R(polylogspace).

Proof . Given L a language in PSPACE, we have to define a constructor h such that $\mathrm{R}(h) = L$. We use the simple idea of just adding a bit of the characteristic string of L as follows

$$h(w) = w\chi_L(s_{|w|}).$$

It is straightforward to check that $\mathrm{R}(h) = L$ and that h is in polylogspace. ■

The other inclusion is more complicated. Given $h \in \mathrm{p}$, in order to check whether an input $x = s_i$ is in $\mathrm{R}(h)$ we can simulate the computations λ, $h(\lambda)$, $h(h(\lambda))$, $h(h^2(\lambda))$, …, $h(h^m(\lambda))$ for successive m until $|h(h^m(\lambda))| > i$. But since the output of some of these computations will be too big to be kept in space polynomial in $|x|$, we cannot expect polynomial space algorithm for $\mathrm{R}(h)$ by using this simple approach. Another idea would be to simulate the computations of $h(h^m(\lambda))$ for successive m, but without writing the full output, that is, recalculating the bits in $h^m(\lambda)$ needed in the computation of $h(h^m(\lambda))$. In this case the stack of the recursion can be too big for PSPACE (e.g. in the cases where $|h(w)| = |w| + 1$ for every input), and the idea does not work in general.

We see in the next theorem what is really R(polylogspace). It corresponds to a class of self-reducible languages that is expected to be different from PSPACE.

Definition 2.2. A language A is PSPACE-*wdq-self-reducible* (where wdq stands for word-decreasing-queries) if $A = L(M, A)$, where M is a PSPACE Oracle Turing Machine that makes only queries strictly smaller than the input (in lexicographical order).

Balcázar defines in [Balc] two types of self-reducibility, namely wdq-self-reducibility and ldq-self-reducibility, ldq standing for length decreasing queries. The most restrictive one is ldq-self-reducibility, where all the queries must be strictly shorter than the input. Notice that wdq-self-reducibility allows exponentially long decreasing chains to exist, while only linearly long chains can appear for the ldq type. Definition 2.2 is obtained by substituting PSPACE computations for P computations in the definition of [Balc].

Theorem 2.3. R(polylogspace) is exactly the class of PSPACE-wdq-self-reducible languages.

Proof . We start by proving that every language in R(polylogspace) is PSPACE-wdq-self-reducible. Let h be a constructor in polylogspace.

We define a PSPACE-oracle machine M that on oracle A and input s_i computes $h(\chi_A[0 \dots m-1])$ for a certain $m < i+1$ such that $|h(\chi_A[0 \dots m-1])| > i$, and gives as output the $i+1$-th bit of $h(\chi_A[0 \dots m-1])$. In the case of $A = \mathrm{R}(h)$, we choose m such that $\chi_A[0 \dots m-1] = h^l(\lambda)$ for some l, and this ensures that $h(\chi_A[0 \dots m-1]) \sqsubseteq A$, thus $M(s_i, A) = A(s_i)$.

For each oracle A, we define the following sequence of natural numbers

$$\begin{aligned} a_0 =& |h(\lambda)| \\ a_{n+1} =& |h(\chi_A[0 \dots a_n - 1])|, \text{ for } n \geq 0. \end{aligned}$$

The machine M on oracle A works as follows

```
INPUT $s_i$
    Find $a_n$ such that $a_n \leq i < a_{n+1}$
    $b$ :=the $i+1$-th bit of the output of $h(\chi_A[0 \dots a_n - 1])$
OUTPUT $b$
```

The computations of the form $h(\chi_A[0 \dots m-1])$ are done by machine M by substituting each access to the input by a query to A. In this way all queries to A are strictly smaller than s_i, since $a_n \leq i$. The computation of M can be done in space polynomial in the length of the input, because the working space of the computations of h is polylogarithmic in i, which is polynomial in s_i, and for the outputs we only need to write their length and their $(i+1)$-th bit if it exists.

When $A = \mathrm{R}(h)$ we have seen that $M(s_i, A) = A(s_i)$, thus M performs a PSPACE-wdq-self-reduction in this case. This completes the first part of the theorem.

To see that every PSPACE-wdq-self-reducible language is in R(polylogspace), fix L PSPACE-wdq-self-reducible via a Turing Machine M. We can define h as in Lemma 2.1

$$h(w) = wL(s_{|w|}),$$

where $L(y)$ is 1 if $y \in L$, 0 otherwise.

It is clear that $\mathrm{R}(h) = L$. In the computation of h, to decide whether $s_{|w|}$ is in L we can use that L is PSPACE-wdq-self-reducible and simulate the computation of M on input $s_{|w|}$, answering to a query s_i, $(i < |w|)$ by checking the ith bit of w. This simulation can be done in space polynomial in the length of $s_{|w|}$, that is, polylogarithmic in the length of w, so h is in polylogspace. ■

In [Balc] it is proven that E_2 has $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete languages that are P-wdq-self-reducible. Since every P-wdq-self-reducible language is clearly PSPACE-wdq-self-reducible, E_2 has $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete languages that are PSPACE-wdq-self-reducible, and we have the following result.

Theorem 2.4. If PSPACE = R(polylogspace) then E_2 = PSPACE.

Proof. By the comment above, E_2 has a PSPACE-wdq-self-reducible complete language A. In the hypothesis that PSPACE = R(polylogspace) and using Theorem 2.3 it is clear that A is in PSPACE.

Since PSPACE is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction, if PSPACE contains A, it contains the whole class E_2. ■

There are other restrictions we can impose on polylogspace functions and still be able to construct the full PSPACE. For instance we can consider only functions that can be computed with on-line polylogspace machines, that is to say, machines that read the input only once and from right to left.

Our model of on-line machine is based on that of Hartmanis, Immerman and Mahaney in [HartImM].

Definition 2.5. An *on-line Turing Machine* is a machine that on input of length n

(a) starts with $\log n$ blank spaces marked on one of the working tapes,

(b) reads the input tape once from left to right, and

(c) writes the output from left to right on a write-only tape.

Definition 2.6. Let plogon be the class of functions that are computable by on-line machines with working and output space polylogarithmic in the size of the input. In this case and for constructor functions only, we do not bound the output space.

Theorem 2.7. PSPACE = R(plogon).

Proof. $\subseteq$) For this inclusion we use the constructor in Lemma 2.1, which can be clearly computed with an on-line machine.

$\supseteq$) Given h a constructor in plogon we construct an algorithm for R(h) that simulates the successive computations $h(\lambda)$, $h(h(\lambda))$, $h(h^2(\lambda)))$, …, $h(h^m(\lambda)))$ by couples. Since we are using an on-line machine, the computation of $h(h^{m-1}(\lambda))$ is identical to that of $h(h^m(\lambda))$ (except for the initially marked blank tapes), until $h(h^{m-1}(\lambda))$ finds the end of its input. We take advantage of this to simulate two of these computations in parallel. Notice that on-line constructors read the whole input before adding new bits to the output (new here means not a part of the copy of the input). We simulate the part of the computation of $h(h^{m-1}(\lambda))$ when the input has been read and new bits of the output are being produced. These output bits are being fed as input bits in the computation of $h(h^m(\lambda))$. In this way we do not need to keep long pieces of R(h), only one bit at a time that as soon as is produced as output is consumed as input.

The next algorithm recognizes R(h). On input s_i it works as follows

```
BEGIN
  $j := 0$;
  FOR $k := 1$ to $\log i$ DO
    $C_1^k$ :=initial configuration in the computation of $h(\lambda)$ with $k$ spaces marked on the output ta
    $C_2^k := C_1^k$;
  END FOR
  $s := 0$
  WHILE $j \leq i$ DO
    {First we simulate the part of the computation of $h(h^{m-1}(\lambda))$ after the whole
    input has been read, getting one bit of output at a time.}
    FOR $k := 1$ to $\log i$ DO in parallel
      simulate the computation of $h$ starting in $C_1^k$
        assuming that the rest of the input is blank,
        until getting the $j$th bit of the output, $b^k$
      $C_1^k$ :=last configuration reached in the simulation
      IF $b^k$ is not blank
        {Now we simulate the part of the computation of $h(h^m(\lambda))$
        after the prefix of the input $h^{m-1}(\lambda)$ has been read,
        reading one bit of input at a time.}
        THEN simulate the computation of $h$ starting in $C_2^k$, assuming that
         the $j$th bit of the input is $b^k$, stopping when accessing this $j$th bit
        $C_2^k$ :=last configuration reached in the simulation
        $j := j + 1$
      ELSE $C_1^k := C_2^k$
      IF $s = k$ THEN $s' = \log j$
    END FOR
  END WHILE
  OUTPUT $b^s$    {For this $b^s$, $R(h)[i] = b^s$}
END
```

In this way we only need space to keep $2 \log i$ configurations of size polylogarithmic in i, which is polynomial space in the length of the input s_i. Thus $\mathrm{R}(h) \in$ PSPACE and we finish the proof. ■

We have proved that plogon is a solution to the equation $\mathrm{R}(\Gamma) =$ PSPACE. In order to define a nontrivial measure in PSPACE, we have to check that the class plogon fulfills the Measure Conservation Theorem, that is, that PSPACE does not have plogon-measure 0. This is straight-forward from the proof of this theorem in Chapter 1. The proof consisted of, for each martingale $d \in \Gamma$, defining a constructor $\delta \in \Gamma$ such that $\mathrm{R}(\delta) \notin \mathrm{S}^\infty[d]$ as follows

$$\delta(w) = \begin{cases} w0 & \text{if } d(w0) \leq d(w1) \\ w1 & \text{otherwise.} \end{cases}$$

Notice that if $d \in$ plogon then the above defined δ is also in plogon. Thus there is no martingale in plogon that succeeds on every language in PSPACE, and we can define a measure in PSPACE from plogon-measure.

In the next section we show that Γ-additivity holds for plogon-measure, and use it for an application concerning self-reducible languages.

2.3 Γ-additivity in PSPACE

In Chapter 1 we introduced Lutz's property of Γ-additivity as a useful tool in the proof that a certain class has Γ-measure 0. All the measures we defined in that chapter had this property. We show here that the same holds for plogon-measure. We then use it to show that the class of LINSPACE-self-reducible-languages for a particular self-reducibility has measure 0 in PSPACE.

Let us remind the definition of Γ-union of Γ-measure 0 sets.

Definition 2.8. A set X is a Γ-*union* of the Γ-measure 0 sets X_j, $j \in \mathbb{N}$ iff $X = \bigcup_{j=0}^{\infty} X_j$ and there exists a Γ-computable 1-MS d such that, for every j, $X_j \subseteq \mathrm{S}^{\infty}[d_j]$.

Lemma 2.9. If X is a plogon-union of plogon-measure 0 sets, then X has plogon-measure 0.

Proof. Let d be given by the definition of Γ-union. Let $\widehat{d}$ be a plogon-computation of d. To prove that X has plogon-measure 0 we define the martingale $d'\colon \{0,1\}^* \to [0,\infty)$ by

$$d'(w) = \sum_{j=0}^{\infty} 2^{\min\{0, -\log(\widehat{d}_{j,1}(\lambda))-2-2^j\}} d_j(w).$$

d' is well defined because since $d_j(\lambda) < \widehat{d}_{j,1}(\lambda) + 2^{-1}$ we have that

$$2^{\min\{0, -\log(\widehat{d}_{j,1}(\lambda))-2-2^j\}} d_j(\lambda) < 2^{-2^j}$$

and thus $d'(\lambda) < \infty$. For other values $w \in \{0,1\}^*$, notice that $d'(w) \leq 2^{|w|} d'(\lambda)$. d' is trivially a martingale such that $X \subseteq \mathrm{S}^{\infty}[d']$.

By the same arguments as in the case of Γ a measure resource-bound (Corollary 1.32), defining plogon-measure using only martingales in plogon is equivalent to defining it with plogon-computable martingales. In order to see that X has plogon-measure 0, we then need only to show that d' is plogon-computable. For this, define a function $\widehat{d'}\colon \mathbb{N} \times \{0,1\}^* \to \mathbf{D}$ by

$$\widehat{d'}_k(w) = \sum_{j=0}^{\log(k+|w|+1)} 2^{\min\{0, -\log(\widehat{d}_{j,1}(\lambda))-2-2^j\}} \widehat{d}_{j,j+k+2}(w).$$

Let us see that $\widehat{d'}$ is a plogon-computation of d'. For each $k \in \mathbb{N}$, $w \in \{0,1\}^*$

$$|\widehat{d'}_k(w) - d'(w)| \leq \sum_{j=0}^{\log(k+|w|+1)} |\widehat{d}_{j,j+k+2}(w) - d_j(w)| + \sum_{j=\log(k+|w|+2)}^{\infty} 2^{-\log(\widehat{d}_{j,1}(\lambda))-2-2^j} d_j(w)$$

$$\leq \sum_{j=0}^{\log(k+|w|+1)} 2^{-j-k-2} + \sum_{j=\log(k+|w|+2)}^{\infty} 2^{|w|} 2^{-\log(\widehat{d}_{j,1}(\lambda))-2-2^j} d_j(\lambda) \leq 2^{-k-1} + 2^{-k-1} = 2^{-k}.$$

It is clear that $\widehat{d'} \in$ plogon, because it is defined as a sum of $O(\log n)$ functions, each of them computed in space $O(\log^l n)$, for a fixed l. ■

We have studied PSPACE-wdq-self-reducibility in section 2.2, showing that there are languages out of PSPACE that are PSPACE-wdq-self-reducible, unless PSPACE = E_2. We now look at a more restrictive form of wdq-self-reducibility, where the machine used has a linear bound on the space and a restriction on the order the queries are made.

Definition 2.10. A language A is LINSPACE-*oq-self-reducible* (where oq stands for ordered queries) if $A = L(M, A)$, where M is a LINSPACE-oracle-machine that for each input makes the queries in lexicographical ascending order, and all of them are strictly smaller than the input (in lexicographical order).

Remark than if A is LINSPACE-wdq-self-reducible via a $O(n)$-truth-table LINSPACE-machine, then A is LINSPACE-oq-self-reducible, because we can order the queries before making them. In particular, most of the known self-reductions for natural problems in PSPACE are $O(n)$-tt (even $O(1)$-tt) and computable in LINSPACE.

Theorem 2.11. The class of LINSPACE-oq-self-reducible languages has measure 0 in PSPACE.

Proof. Let Y be the class of LINSPACE-oq-self-reducible languages. We will prove that Y has plogon-measure 0.

Let $\{M_i \mid i \in \mathbb{N}\}$ be a recursive enumeration of all Oracle Turing Machines working in space $|z|^2$ on input z.

Let $X_i = \{B \mid B = L(M_i, B)\}$ if the queries of M_i on any input and any oracle are ordered and strictly smaller than the input, let X_i be empty otherwise. Let $X = \bigcup_i X_i$. It is clear that $Y \subseteq X$. In fact, the techniques in [Balc] show that each self-reduction corresponds to a single language, that is, for each $i \in \mathbb{N}$ either $|X_i| = 1$ or $|X_i| = 0$.

We are going to see that X is a plogon-union of the plogon-measure 0 sets X_i, $i \in \mathbb{N}$. Let us define d a plogon-computable 1-MS as required by Definition 2.8.

Let $i \in \mathbb{N}$ such that $X_i \neq \emptyset$. Let $w \in \{0,1\}^*, b \in \{0,1\}$.

$d_i(\lambda) = 1$

If $s_{|w|} \in \{0\}^*$ then let $s_{k_1}, s_{k_2}, \ldots, s_{k_l}$ be the queries of M_i on input $s_{|w|}$. Let $a \in \{0,1\}$ be the output of M_i on input $s_{|w|}$ when the queries are answered according to $w[k_1], \ldots, w[k_l]$.

$$d_i(wb) = \begin{cases} 2 * d_i(w), & \text{if } b = a; \\ 0, & \text{otherwise.} \end{cases}$$

Otherwise $d_i(wb) = d_i(w)$.

d is in plogon, because for any i we can compute $d_i(w)$ by simulating in parallel the computations of $M_i(0), M_i(0^2), \ldots, M_i(0^{\log(|w|)})$, answering to the different queries in lexicographical order while reading the input on-line.

More precisely, on input w, the following algorithm computes $d_i(w)$

```
**BEGIN**
$Q := \emptyset$
$d := 1$
FOR $m := 1$ to $\log(|w|)$ DO
    $C_m$ :=configuration of machine $M_i$ on input $0^m$ when making its first query, $q_m$
    $Q := Q \cup \{(m, q_m)\}$
END FOR
FOR $j := 0$ to $|w| - 1$ DO
    FOR $m := 1$ to $\log(|w|)$ DO
        IF $(m, s_j)$ is in $Q$
            THEN $C_m$ :=configuration of machine $M_i$ on input $0^m$ after answering query $s_j$
              according to $w[j]$ and making its next query, $q_m$, if it exists
            IF $C_m$ is a final configuration (1 for accepting, 0 for rejecting)
                THEN IF $w[2^m - 1] = C_m$
                THEN $d = 2d$
                ELSE $d = 0$
            END IF
            $Q := Q \cup \{(m, q_m)\} - \{(m, s_j)\}$
        END IF
    END FOR
END FOR
**END** .
```

The algorithm works in space plogon because in each step we only need to keep a logarithmic number of polylogarithmic size configurations.

If $B \in X_i$ then $d_i(B[0 \dots 2^m - 1]) = 2d_i(B[0 \dots 2^m - 2])$ for each $m \in \mathbb{N}$, so $X_i \subseteq \mathrm{S}[d_i]$.

Using Lemma 2.9 we have the result. ∎

We could also state a plogon version of the Borel-Cantelli Lemma in Chapter 1. But in this case we need a very restrictive notion of uniform convergence of series, which makes a plogon-formulation unattractive and difficult to apply.

We will use plogon-measure to study other classes inside PSPACE in the following chapters.

Chapter 3

Measure versus category: the P-Bi-immune sets

3.1 Introduction

In this chapter, we study the class of P-bi-immune languages that are in E. Informally, a set A is bi-immune for a complexity class $\mathcal{C}$ if no nontrivial part of A or of its complement can be 'attacked' by any algorithm of 'type $\mathcal{C}$'. More precisely, a set A is $\mathcal{C}$-bi-immune if no nontrivial subset of A or its complement is in $\mathcal{C}$.

The notion of immunity was first introduced by Post [Post44] in recursive function theory. Flajolet and Steyaert transformed it into the complexity theoretic setting in [FlajSt74a] and [FlajSt74b]. Hartmanis and Berman show that E contains a P-bi-immune set ([BermHa], observed in [KoMo]), and an application of [GeskHuS] yields that for all $c > 0$ there exists a DTIME(2^{cn})-bi-immune set in E. P-bi-immunity is also studied in detail by Balcázar and Schöning in [BalcSc], where several characterizations are presented; for instance, a recursive set $A \subseteq \{0,1\}^*$ is P-bi-immune if and only if $\{0,1\}^*$ is a complexity core for A.

Our goal here is to study the size of the class X of all P-bi-immune languages inside E, that is, to compare $X \cap \mathrm{E}$ and E by size criteria. We would like to generalize 'There exists a P-bi-immune set in E' to 'Almost every set in E is P-bi-immune'.

For this purpose we will use two ways of size-classification of classes within exponential time, namely measure in E and category in E. We have fully introduced the first one in Chapter 1. The second one is defined by Lutz in [Lutz90] by bounding resources in topological Baire category (see section 3.3 for a revision of resource-bounded category).

We will prove that the class of P-bi-immune languages has measure 1 in E. This implies that almost every language in E is P-bi-immune, and so it extends the previously mentioned result from [BermHa] (in fact, it extends [GeskHuS] since we will see that for any $c > 0$ almost every language in E is DTIME(2^{cn})-bi-immune). As a corollary, we show that the class of $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete sets for E has measure 0 in E.

We obtain generalizations of the above result, such as: E-bi-immunity defines a measure 1 class within E_2. So almost every language in E_2 is E-bi-immune.

Classical Baire Category differs drastically from Lebesgue measure in the sense that 'large' classes for Baire can be 'small' for Lebesgue, and vice versa [Oxto]. We prove here that the class of P-bi-immune languages is a natural example that witnesses the differences between category and measure for the resource-bounded formulation. The class of P-bi-immune sets is not 'measurable in E' in the category setting (formally, it does not have the property of Baire), whereas it has measure 1 in E.

The two different approaches of category and measure give us two different concepts of typical language, namely generic language and random language. We contrast these in the resource-bounded setting, observing that a pseudo-random language is necessarily P-bi-immune, while a pseudo-generic language can have an infinite subset in P.

The main results for the class of P-bi-immune languages inside E in this chapter can be restated for the class of DLOG-bi-immune languages inside PSPACE, as we remark at the end of each section.

The results in this chapter appear in [Mayo92a].

3.2 P-bi-immunity and resource-bounded measure

In this section we prove that the class of P-bi-immune sets has measure 1 in E. As a consequence, almost every set in E is P-bi-immune, that is to say, almost every set recognizable in linear exponential time has no algorithm that recognizes it and works in polynomial time on an infinite number of instances. Next we explain some consequences of this result for $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete languages.

First, we review the notion of immunity.

Definition 3.1. Let $\mathcal{C}$ be a class of languages, and L be a language. We say that L is *$\mathcal{C}$-immune* iff L does not have an infinite subset that belongs to $\mathcal{C}$.

Definition 3.2. Let $\mathcal{C}$ be a class of languages, and L be a language. We say that L is *$\mathcal{C}$-bi-immune* iff both L and the complement of L are $\mathcal{C}$-immune.

Next we prove our main result, by using the Γ-additivity property (Lemma 1.35).

Theorem 3.3. The class of P-bi-immune languages has p-measure 1, and thus measure 1 in E.

Proof . Let Y be the class of non-P-bi-immune languages. By Definition 1.13, if we prove that Y is a p-measure 0 class we have the theorem.

Let $A \in \mathrm{E}$ be a universal language for the class P, that is, if we define for each $i \in \mathbb{N}$, $A_i = \{x \bigm| \langle x, i\rangle \in A\}$, then $\mathrm{P} = \{A_i \bigm| i \in \mathbb{N}\}$.

For $i > 0$ we define the classes Y_{2i-1} and Y_{2i} as follows. If $|A_i| = \infty$ then

$$\begin{aligned} Y_{2i-1} &= \{L \mid A_i \subseteq L\}, \text{ and} \\ Y_{2i} &= \{L \mid A_i \subseteq L^c\}. \end{aligned}$$

If $|A_i| < \infty$ then $Y_{2i-1} = Y_{2i} = \emptyset$. It is easy to see that Y is contained in the union of all the classes Y_m.

Now we will use Lemma 1.35 to prove that Y has p-measure 0. For this we have to build a p-computable 1-MS d that witnesses that Y is a p-union of the p-measure 0 classes Y_m.

Let $m \in \mathbb{N}, w \in \{0,1\}^*$. d_m is defined as follows,

$$d_m(\lambda) \quad = \quad 1$$

If $m = 2i - 1$ then

When $s_{|w|} \in A_i$ we define

$$\begin{aligned} d_{2i-1}(w0) &= 0 \\ d_{2i-1}(w1) &= 2 * d_{2i-1}(w) \end{aligned}$$

and when $s_{|w|} \notin A_i$ for each $b \in \{0,1\}$

$$d_{2i-1}(wb) = d_{2i-1}(w)$$

If $m = 2i$ then

When $s_{|w|} \in A_i$ we set

$$\begin{aligned} d_{2i}(w0) &= 2 * d_{2i}(w) \\ d_{2i}(w1) &= 0 \end{aligned}$$

and when $s_{|w|} \notin A_i$ for each $b \in \{0,1\}$

$$d_{2i}(wb) = d_{2i-1}(w)$$

It is straightforward to check that d is a 1-MS. Since $A \in$ E, then there exists $c > 0$ such that d can be computed in time $2^{c(\log(|w|)+\log m)}$ on input $\langle m, w\rangle$, and thus $d \in$ p.

Let us see that for each $m \in \mathbb{N}$, $Y_m \subseteq \mathrm{S}^\infty[d_m]$. Let $i > 0$ such that $|A_i| = \infty$. Let $B \in Y_{2i-1}$. For each n such that $s_n \in A_i$ we know that, since $B \in Y_{2i-1}$, $s_n \in B$, and thus $d_{2i-1}(\chi_B[0..n]) = 2 \cdot d_{2i-1}(\chi_B[0..n-1])$, and for n such that $s_n \notin A_i$, $d_{2i-1}(\chi_B[0..n]) = d_{2i-1}(\chi_B[0..n-1])$. But the case $s_n \in A_i$ happens infinitely often, thus $\lim_n d_{2i-1}(\chi_B[0..n]) = \infty$, and $B \in \mathrm{S}^\infty[d_{i-1}]$. The proof of $Y_{2i} \subseteq \mathrm{S}^\infty[d_{2i}]$ is analogous.

Applying Lemma 1.35 we have that Y has p-measure 0, then Y^c has p-measure 1 and we have completed the proof of the theorem. ■

Recently, Juedes and Lutz [JuedLu94a] have improved Theorem 3.3 by looking at strong P-bi-immunity, a notion defined in [BalcSc] that is more restrictive than P-bi-immunity. They show that the class of strongly P-bi-immune languages has measure 1 in E. Since every strongly-P-bi-immune language is P-bi-immune, their result implies Theorem 3.3.

Next we look at the complexity cores of languages in E. A complexity core for a language L is a set of 'infeasible' inputs for every algorithm that recognizes L. Complexity cores were introduced by Lynch in [Lync].

Definition 3.4. An infinite set $U \subseteq \{0,1\}^*$ is a *complexity core* for a language A if for every machine M that accepts A and every polynomial p there are at most finitely many $z \in U$ such that the time of machine M on input z is smaller than $p(|z|)$.

A characterization of P-bi-immune sets in [BalcSc] says that a language is P-bi-immune if and only if it has $\{0,1\}^*$ as a complexity core. Thus we have the next corollary.

Corollary 3.5. Almost every set in E has $\{0,1\}^*$ as a complexity core.

In the next Theorem we extend Theorem 3.3 to the class of $\mathcal{C}$-bi-immune languages, for $\mathcal{C}$ any class such that E has a universal language for $\mathcal{C}$. The same kind of results hold for measure in ESPACE.

Theorem 3.6.

a) Let $\mathcal{C}$ be a complexity class such that there exists $A \in$ E with $\mathcal{C} \subseteq \{A_i \mid i \in \mathbb{N}\}$. Then the class of $\mathcal{C}$-bi-immune languages has p-measure 1, and thus measure 1 in E.

b) Let $\mathcal{C}$ be a complexity class such that there exists $A \in \mathrm{E}_2$ with $\mathcal{C} \subseteq \{A_i \mid i \in \mathbb{N}\}$. Then the class of $\mathcal{C}$-bi-immune languages has p_2-measure 1, and thus measure 1 in E_2.

c) Let $\mathcal{C}$ be a complexity class such that there exists $A \in$ ESPACE with $\mathcal{C} \subseteq \{A_i \mid i \in \mathbb{N}\}$. Then the class of $\mathcal{C}$-bi-immune languages has pspace-measure 1, and thus measure 1 in ESPACE.

The proof is similar to that of Theorem 3.3, and therefore is omitted.

Next we look at the class of complete sets in E. Complete sets are considered the most difficult in a class, and for instance in [StocCh], it is shown that a problem defined using a certain two-person combinatorial game is intractable because it is $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete for E. We want to know whether completeness is a typical property in E. We study $\leq^{\mathrm{p}}_{\mathrm{m}}$-completeness, that coincides with $\leq^{\mathrm{p}}_{1-\mathrm{tt}}$-completeness as proven in [HomeKuR].

Corollary 3.7. The class of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete languages for E has measure 0 in E. The class of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete languages for NE has measure 0 in E_2.

Proof . As proven in [Berm], no $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set for E is P-bi-immune, so the class of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets is included in a measure 0 in E class by Theorem 3.3, and from Lemma 1.21 b) it has measure 0 in E. The second part is analogous, using Theorem 3.6 and the fact that no $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set for NE is E-bi-immune (from [Berm]). ∎

Notice that it is not known whether NE $\subseteq \mathrm{E}_2$. Also, from [BuhrSpT] every $\leq^{\mathrm{p}}_{1-\mathrm{tt}}$-complete set for NE is $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete. Very recently, Ambos-Spies, Terwijn and Zheng [AmboTeZ] have shown that the class of $\leq^{\mathrm{p}}_{\mathrm{btt}}$-complete languages has measure 0 in E.

Next we see that the typical languages for resource-bounded measure are E-bi-immune. From Chapter 1 we know that most languages in E_2 are p-random and that most languages in E_2SPACE are pspace-random.

Corollary 3.8. Every p-random language is E-bi-immune. Every pspace-random language is ESPACE-bi-immune

Proof . For each $c > 0$, the class DTIME(2^{cn}) has a universal language in E. Thus Theorem 3.6 proves that the class of DTIME(2^{cn})-bi-immune sets has p-measure 1. Since by definition p-random languages belong to every p-measure 1 class, it follows that they are DTIME(2^{cn})-bi-immune for every c, and thus E-bi-immune. The same argument works in the proof of pspace-random languages being ESPACE-bi-immune. ∎

We finish this section by looking at the class of DLOG-bi-immune languages inside of PSPACE. We have a similar result to that of Theorem 3.3.

Theorem 3.9.

a) The class of DLOG-bi-immune languages has measure 1 in PSPACE.

b) Let $\mathcal{C}$ be a complexity class such that there exists $A \in$ PSPACE with $\mathcal{C} \subseteq \{A_i \mid i \in \mathbb{N}\}$. Then the class of $\mathcal{C}$-bi-immune languages has plogon-measure 1, and thus measure 1 in PSPACE.

The proof is similar to that of Theorem 3.3.

We also have the following corollary for $\leq^{\log}_{\mathrm{m}}$-complete languages for PSPACE, that coincide with $\leq^{\mathrm{log}}_{\mathrm{1-tt}}$-complete ones by [HomeKuR].

Corollary 3.10. The class of $\leq^{\mathrm{log}}_{\mathrm{m}}$-complete languages for PSPACE has measure 0 in PSPACE.

Proof . The results in [BuhrHoT] imply that no $\leq^{\mathrm{log}}_{\mathrm{m}}$-complete set for PSPACE is DLOG-bi-immune, so the class of $\leq^{\mathrm{log}}_{\mathrm{m}}$-complete sets has measure 0 in PSPACE. ∎

3.3 P-bi-immunity and resource-bounded category

In this section we introduce resource-bounded category, a topological based way of size distinction for subclasses of E, ESPACE, REC and other recursive classes. We show that the class of P-bi-immune languages is neither large nor small in E following resource-bounded category. We finish by proving that for a class that is closed under finite variations, such as the class of P-bi-immune languages, the fact of being neither large nor small in E in the category sense implies that it is nonmeasurable in E in the category setting (formally, it lacks the property of Baire in E). Since we have seen in the last section that the same class has measure 1 in E, this shows that resource-bounded measure and resource bounded category are incomparable.

Classical Baire category was introduced by R. Baire in 1899 (and reviewed for instance in [Oxto]). Lutz defines a resource-bounded category in [Lutz90], later studied by Fenner [Fenn], based on classical category in $\{0,1\}^{\infty}$ with the usual topology of cylinders. Both classical and resource-bounded category can be characterized in terms of Banach-Mazur games, which are a type of two person games. We present here resource-bounded category only through Banach-Mazur games, which are simpler to understand and to use for our purposes.

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

Definition 3.11. Let X be a class of languages, let Γ_1 and Γ_2 be two measure resource-bounds. A *Banach-Mazur game* $G[X;\Gamma_1,\Gamma_2]$ is a game with two players I and II such that player I has chosen a constructor $g \in \Gamma_1$ and player II has chosen a constructor $h \in \Gamma_2$. Starting from $w := \lambda$, they play indefinitely as follows

```
$w := \lambda$
REPEAT forever
        player I plays setting $w := g(w)$
        player II plays setting $w := h(w)$.
END REPEAT
```

As they play eternally they build an element of $\{0,1\}^{\infty}$. We denote as $\mathrm{R}(g,h)$ the language built in this Banach-Mazur game. Notice that following Definitions 1.14 and 1.15, the composition of g and h, $h \circ g$, is a constructor and $\mathrm{R}(g,h) = \mathrm{R}(h \circ g)$.

Definition 3.12. A *winning strategy* for player II in the game $G[X;\Gamma_1,\Gamma_2]$ is a constructor $h \in \Gamma_2$ such that for every constructor $g \in \Gamma_1$, $\mathrm{R}(g,h) \notin X$.

Intuitively, player II has a winning strategy when he has the ability to, starting with any finite prefix $w \in \{0,1\}^*$, construct a language L with $w \sqsubseteq L$ that is not in X.

Now we can define Γ-meager classes, which are the 'smallest' ones in category. (In classical Baire Category, a meager class is sometimes referred to as a class of first category.)

Definition 3.13. Let X be a class of languages. X is Γ-*meager* iff player II has a winning strategy for $G[X;\mathbf{all},\Gamma]$.

We define co-meager classes as 'large' classes.

Definition 3.14. Let X be a class of languages. X is Γ-*co-meager* iff X^c is Γ-meager.

We can now compare the definitions of measure and category (for instance Definition 1.12 and Definition 3.13), to find a hint of why category and measure are incomparable. A class X is Γ-meager when there exists a function in Γ that can, starting with any finite prefix $w \in \{0,1\}^*$, construct a language L with $w \sqsubseteq L$ that is not in X. This intuitively means that X is Γ-meager when Γ has enough computing power to find "holes" in X in every cylinder. In the case of measure, X has Γ-measure 0 when there is a function in Γ that, for each $w \in \{0,1\}^*$, predicts reasonably well all languages in $X \cap \mathbf{C}_w$. Roughly X is meager when it is easy to get out of it, and it is measure 0 when it is easy to predict.

Next we need to translate the last definitions into a concept of "category within a class".

Definition 3.15. Let X be a class of languages. X is *meager in* $\mathrm{R}(\Gamma)$ iff $X \cap \mathrm{R}(\Gamma)$ is Γ-meager.

Definition 3.16. Let X be a class of languages. X is *co-meager in* $\mathrm{R}(\Gamma)$ iff X^c is meager in $\mathrm{R}(\Gamma)$.

These definitions are nontrivial because Theorem 3.12 in [Lutz90] implies that $\mathrm{R}(\Gamma)$ is not Γ-meager. That theorem is a resource-bounded version of the classical Baire Category Theorem; in fact when $\Gamma = \mathbf{all}$ in Definitions 3.15 and 3.16 we get classical Baire category. In that context, typical languages are called generic. We define here Γ-generic or pseudo-generic languages.

Definition 3.17. Let L be a language. L is Γ-*generic* iff L belongs to every Γ-co-meager class.

(There exist different notions of genericity, for instance the one studied by Ambos-Spies, Fleischhack and Huwik in [AmboFlH], that has been recently connected with resource-bounded measure in [AmboNeT].)

The following lemma states some basic properties of meager sets, and is proved by Lutz in [Lutz90].

Lemma 3.18. A subset of a Γ-meager set is Γ-meager. A finite union of Γ-meager sets is Γ-meager. Every Γ-meager set is meager in $\mathrm{R}(\Gamma)$.

Let us show that the class of P-bi-immune languages is neither meager nor co-meager in E. Even a larger class, the P-immune languages, is not co-meager in E.

Theorem 3.19. The class of P-bi-immune languages is not meager in E.

Proof . We denote with X the class of P-bi-immune languages. By Definition 3.15 we have to see that there is no winning strategy for player II in the game $G[X \cap \mathrm{E}; \mathbf{all}, \mathrm{p}]$, that is to say, for every constructor $h \in \mathrm{p}$, there exists a constructor $g \in \mathbf{all}$, such that $\mathrm{R}(g, h) \in X \cap \mathrm{E}$.

Let us start by introducing some notation for P-bi-immunity, that we use next in the definition of g.

Let $A \in \mathrm{E}$ be a universal language for the class P, as in Theorem 3.3, that is, for each $i \in \mathbb{N}$, $A_i = \{x \mid \langle x, i\rangle \in A\}$, then $\mathrm{P} = \{A_i \mid i \in \mathbb{N}\}$.

Given two languages B and L, there exist $u, v \in B$ such that $L(u) \neq L(v)$ if and only if $B \not\subseteq L$ and $B \not\subseteq L^c$. Thus a language L is P-bi-immune if and only if for each $i \in \mathbb{N}$ with $|A_i| = \infty$, there exist $u, v \in A_i$ such that $u \in L$ and $v \notin L$. We can express this last condition in terms of finite prefixes of L as follows. A language L is P-bi-immune if and only if for each $i \in \mathbb{N}$ with $|A_i| = \infty$, there exist $\gamma \in \{0,1\}^*$, $\gamma \sqsubseteq L$ such that

$$\exists s_n, s_m \in A_i \text{ with } 0 \leq n, m < |\gamma| \text{ and } \gamma[n] \neq \gamma[m]. \tag{3.1}$$

We say that index i has been diagonalized in γ, and denote it Diagonalized(i, γ), when condition (3.1) holds for this i and γ, that is,

$$\text{Diagonalized}(i, \gamma) \equiv [\![\; \exists s_n, s_m \in A_i \text{ such that } 0 \leq n, m < |\gamma| \; \text{ and } \; \gamma[n] \neq \gamma[m] \;]\!].$$

L is P-bi-immune if and only if for each $i \in \mathbb{N}$ with $|A_i| = \infty$, there exist $\gamma \in \{0,1\}^*$, $\gamma \sqsubseteq L$, such that Diagonalized(i, γ) =True.

For $\gamma \in \{0,1\}^*$ and $q \geq |\gamma|$, the set Diagonalizable(γ, q) contains those indexes that have not been diagonalized in γ and can be diagonalized using a string s_m in $\{s_{|\gamma|}, \ldots, s_q\}$, that is

$$\begin{aligned}\text{Diagonalizable}(\gamma, q) = \{i \mid \text{Diagonalized}(i, \gamma) = \text{False and}& \\ \exists s_n, s_m \in A_i \text{ such that } n < m,\ |\gamma| \leq m \leq q\}&.\end{aligned}$$

Fix $h \in \mathrm{p}$. Next we define g such that $\mathrm{R}(g, h)$ is a P-bi-immune language in E. On input α, g tries to get Diagonalized$\big(i, g(\alpha)\big)$ =True for i in $\{1, \ldots, |\alpha|\}$. In order to do this, for each s_k with $k \geq |\alpha|$, g checks whether some index in $\{1, \ldots, |\alpha|\}$ can be diagonalized using s_k, and if so the diagonalization is performed. This process goes on until no diagonalization of an index in $\{1, \ldots, |\alpha|\}$ can be performed using a string in $\{s_k, \ldots, s_{2^k}\}$. Then g gives an output of length k. Since Player II next turn uses only polynomial time, it can only set values of $\mathrm{R}(g, h)$ for strings in $\{s_k, \ldots, s_{2^k}\}$ and no opportunity of diagonalization for indexes in $\{1, \ldots, |\alpha|\}$ is jeopardized by Player II.

Formally, g is the function computed by the algorithm in Figure 1.

Let us show that $\mathrm{R}(g, h)$ is P-bi-immune, that is, for each $i \in \mathbb{N}$ if $|A_i| = \infty$ then there exists $\gamma \in \{0,1\}^*$ such that $\gamma \sqsubseteq \mathrm{R}(g, h)$ and Diagonalized(i, γ)=True.

```
BEGIN
  INPUT $\alpha$
  $\gamma := \alpha$
  IF Diagonalizable$(\alpha, 2^{|\alpha|}) \cap \{1, \ldots, |\alpha|\} = \emptyset$ THEN $\gamma := \alpha 0$  {This is to ensure $\alpha \subsetneq g(\alpha)$}
  $k := |\gamma|$
  WHILE Diagonalizable$(\gamma, 2^k) \cap \{1, \ldots, |\alpha|\} \neq \emptyset$ DO
      IF Diagonalizable$(\gamma, k) \cap \{1, \ldots, |\alpha|\} \neq \emptyset$
      THEN
          $i := \min\{j \mid j \in$ Diagonalizable$(\gamma, k)\}$
          $n := \min\{r \mid s_r \in A_i\}$
          IF $\gamma[n] = 0$ THEN $\gamma := \gamma 1$
          IF $\gamma[n] = 1$ THEN $\gamma := \gamma 0$
          {At this point we know that Diagonalized$(\gamma, i)$ =True, since $s_n, s_k \in A_i$ and
          $\gamma[n] \neq \gamma[k]$}
      ELSE $\gamma := \gamma 0$
      $k := |\gamma|$
  END WHILE
  OUTPUT $\gamma$
END .
```

Figure 1: Algorithm that computes g.

Remark that by the termination condition of the while loop, for each $\alpha \in \{0,1\}^*$

$$\text{Diagonalizable}(g(\alpha), 2^{|g(\alpha)|}) \cap \{1, \ldots, |\alpha|\} = \emptyset. \tag{3.2}$$

For each $l \in \mathbb{N}$, let $\alpha_l = (h \circ g)^l(\lambda)$. That is, $\alpha_0, \alpha_1, \ldots$, are the successive inputs to g in the game against h, and for every l, $\alpha_l \sqsubseteq \mathrm{R}(g,h)$. Since $h \in \mathrm{p}$, there is an $l_0 \leq 1$ such that $|h(x)| < 2^{|x|}$ for each x such that $|x| \geq |\alpha_{l_0}|$.

Next we show by induction on i that if $|A_i| = \infty$ then there exists $\gamma \in \{0,1\}^*$ such that $\gamma \sqsubseteq \mathrm{R}(g,h)$ and Diagonalized(i, γ)=True.

For $i = 1$, if $|A_1| < \infty$ then we are done. If $|A_1| = \infty$, let s_n be the first string in A_1, let s_m be the smallest string in A_1 such that $n < m$ and $|\alpha_{l_0}| \leq m$. Let $l \in \mathbb{N}$ be such that $|\alpha_l| \leq m < |\alpha_{l+1}|$. We show that Diagonalized$\big(1, g(\alpha_l)\big)$ =True. From equation (3.2) Diagonalizable$(g(\alpha_l), 2^{|g(\alpha_l)|}) \cap \{1, \ldots, |\alpha_l|\} = \emptyset$, thus $1 \notin$ Diagonalizable$(g(\alpha_l), 2^{|g(\alpha_l)|})$. But by the choice of l, $2^{|g(\alpha_l)|} \geq |f\big(g(\alpha_l)\big)| = |\alpha_{l+1}| > m$. Thus s_m is an opportunity of diagonalizing $i = 1$ in the computation of $g(\alpha_l)$, this means that either Diagonalized$\big(1, \alpha_l\big)$ was already True or g uses s_m to get Diagonalized$\big(1, g(\alpha_l)\big)$ =True. This finishes the case $i = 1$.

For the induction step, if $|A_i| < \infty$ then we are done. If $|A_i| = \infty$ then by induction hypothesis for each $j < i$ with $|A_j| = \infty$ there exists $\gamma_j \sqsubseteq \mathrm{R}(g,h)$ such that Diagonalized(j, γ_j)=True. Take γ the longest of these γ_j. Let F be the union of all finite languages in $A_1, \ldots, A_{i-1}$, let s_t be the last string in F. Let r be the maximum of t, $|\gamma|$, i and $|\alpha_{l_0}|$. Let s_n be the first string in A_i. Let s_m be the smallest string in A_i such that $n, r < m$. Let $l \in \mathbb{N}$ be such that $|\alpha_l| \leq m < |\alpha_{l+1}|$. By equation (3.2)

$i \notin \text{Diagonalizable}(g(\alpha_l), 2^{|g(\alpha_l)|})$. But by the choice of m and l,

$$2^{|g(\alpha_l)|} \geq |f\big(g(\alpha_l)\big)| = |\alpha_{l+1}| > m,$$

and for each $\gamma \sqsubseteq \mathrm{R}(g, h)$ with $|\gamma| \geq |\alpha_l|$,

$$\min\{j \mid j \in \text{Diagonalizable}(g(\alpha_l), 2^{|g(\alpha_l)|})\} \geq i.$$

Thus s_m is an opportunity of diagonalizing i in the computation of $g(\alpha_l)$, this means that either Diagonalized$\big(i, \alpha_l\big)$ was already True or g uses s_m to get Diagonalized$\big(i, g(\alpha_l)\big)$ =True. In both cases Diagonalized$\big(i, g(\alpha_l)\big)$ =True, and the induction proof is finished. We have shown that $\mathrm{R}(g, h) \in X$.

The language built in this game, $\mathrm{R}(g, h)$, is in E because to see if $z \in \mathrm{R}(g, h)$ it is enough to play the game up to obtaining a string of length $2^{|z|+1} - 1$. In the worst case we have to recognize languages $A_1, \dots, A_{2^{|z|+1}-2}$ on inputs $s_0, \dots, s_{2^{|z|+1}-2}$, which have length at most $|z|$, and to compute h for $2^{|z|+1} - 2$ inputs of length $\leq 2^{|z|+1} - 2$. So the total time is bounded by $2^{O(|z|)}$. This is why, even though $g \notin \mathrm{p}$, $\mathrm{R}(g, h) \in \mathrm{E}$. ■

Note that using Lemma 3.18 we have that the class of P-bi-immune languages is not p-meager.

Theorem 3.20. The class of P-immune languages is not co-meager in E.

Proof . We will denote with Y the class of non-P-immune languages.

By Definition 3.16 we have to see that there is no winning strategy for player II in the game $G[Y \cap \mathrm{E}; \textbf{all}, \mathrm{p}]$, that is to say: for every constructor $h \in \mathrm{p}$, there exists a constructor $g \in \textbf{all}$, such that $\mathrm{R}(g, h) \in Y \cap \mathrm{E}$.

So given h, we have to build g that puts a set in P inside $\mathrm{R}(g, h)$.

For $h \in \mathrm{p}$, there exists c with $|h(w)| < 2^{|w|}$ for all $w \in \{0,1\}^*$ such that $|w| \geq 2^c$.

We define the sequence $\{a_n\}$:

$$\begin{aligned} a_0 &= c \\ a_n &= 2^{a_{n-1}},\ n \geq 1. \end{aligned}$$

The set in P that we are going to include in $\mathrm{R}(g, h)$ is $L = \{0^{a_n} \mid n \geq 0\}$. Note that $0^{a_n} = s_{a_{n+1}-1}$.

Algorithm for g:

```
BEGIN
  INPUT $\alpha$
  IF $\alpha = \lambda$ THEN $\gamma = 0^{a_1-1}1$
  ELSE compute $n$ such that $a_{n-1} \leq |\alpha| < a_n$
       $\gamma := \alpha 0^{a_n-|\alpha|-1}1$
  OUTPUT $\gamma$
END .
```

To see that $L \subseteq \mathrm{R}(g,h)$ just notice that if $|\alpha| = a_n$ then $|h(\alpha)| < 2^{a_n} = a_{n+1}$, so the bits corresponding to the strings in L are never affected by h.

As the exponential function is time constructible, g is in p and since $\mathrm{R}(g,h) = \mathrm{R}(h \circ g)$, $\mathrm{R}(g,h) \in \mathrm{E}$. ■

Thus the smaller class of P-bi-immune sets is not co-meager in E either, therefore the P-bi-immune languages form a class that is neither meager not co-meager in E.

Using essentially the same techniques we have the following results.

Theorem 3.21. The class of E-bi-immune languages is neither meager not co-meager in E_2. The class of PSPACE-bi-immune languages is neither meager not co-meager in ESPACE.

For the class REC we obtain:

Theorem 3.22. For any recursively presentable class $\mathcal{C}$ with $\mathrm{P} \subseteq \mathcal{C}$, the class of $\mathcal{C}$-bi-immune languages is neither meager nor co-meager in REC.

Lutz (personal communication) has pointed out that these results imply that the class of P-bi-immune languages lacks the property of Baire in E (and classes up to REC). For the sake of completeness, we now introduce the resource-bounded property of Baire and the zero-one law for Baire category that supports this inference.

Classically, an open set in $\{0,1\}^\infty$ is a union of cylinders and a closed set is the complement of an open set. Also in the classical sense, a set X has the property of Baire if and only if there is an open set G such that $X \Delta G$ is meager. (This is the Baire category analogue of the fact that a set X is Lebesgue measurable if and only if there is an F_σ set —equivalently, a G_δ set— H such that $X \Delta H$ has measure 0.) The extension of this notion to complexity classes is natural. We restrict the open sets to those that are Γ-unions of cylinders, and define the property of Baire in $\mathrm{R}(\Gamma)$ as follows

Definition 3.23. A class X is *open in* $\mathrm{R}(\Gamma)$ iff $\exists h \in \Gamma$ such that $X \cap \mathrm{R}(\Gamma) = (\bigcup_k \mathbf{C}_{h(0^k)}) \cap \mathrm{R}(\Gamma)$. A class X is *closed in* $\mathrm{R}(\Gamma)$ iff it is the complement of an open class in $\mathrm{R}(\Gamma)$.

Definition 3.24. A class X has the *property of Baire in* $\mathrm{R}(\Gamma)$ iff $X = G \Delta Q$, where G is open in $\mathrm{R}(\Gamma)$ and Q is meager in $\mathrm{R}(\Gamma)$.

Definition 3.25. A class X of languages is *closed under finite variations* if for all languages L and L', if $L \in X$ and $L \Delta L'$ is finite, then $L' \in X$.

The following lemma is a straightforward generalization of Theorem 21.4 in [Oxto], which is the Baire category analogue of the Kolmogorov zero-one law for measure. To prove the lemma we use the next auxiliary proposition.

Proposition 3.26. If X is a class of languages that is closed under finite variations then X is meager in $\mathrm{R}(\Gamma)$ if and only if there exists $w \in \{0,1\}^*$ such that $X \cap \mathbf{C}_w$ is meager in $\mathrm{R}(\Gamma)$.

Proof . From left to right, just take $w = \lambda$.

From right to left, let $w \in \{0,1\}^*$ be such that $X \cap \mathbf{C}_w$ is meager in $\mathrm{R}(\Gamma)$, then there is a winning strategy h for player II in the game $G[(X \cap \mathbf{C}_w) \cap \mathrm{R}(\Gamma); \mathbf{all}, \Gamma]$.

Take $y \in \{0,1\}^*$ such that $|w| = |y|$. Let us show that $X \cap \mathbf{C}_y$ is meager in $\mathrm{R}(\Gamma)$.

Define $\widehat{h}\colon \{0,1\}^* \to \{0,1\}^*$ as follows. If $y \sqsubseteq x$, then let $z = w \cdot x[|w|..|x|-1]$, that is, z is the result of substituting y by w as prefix of x and let $\widehat{h}(x) = x \cdot h(z)[|x|..|h(z)|-1]$, that is, $\widehat{h}(x)$ is the result of substituting w by y as a prefix of $h(z)$. If $y \not\sqsubseteq x$, then $\widehat{h}(x)$ is the first string $z \in \{0,1\}^*$ such that $x \sqsubsetneq z$ and $\mathbf{C}_y \cap \mathbf{C}_z = \emptyset$.

We claim that $\widehat{h}$ is a winning strategy for player II in the game

$$G[X \cap \mathbf{C}_y \cap \mathrm{R}(\Gamma); \mathbf{all}, \Gamma].$$

It is clear that $\widehat{h} \in \Gamma$. To see that $\widehat{h}$ wins, let g be an arbitrary strategy for player I. We have two cases:

(i) Case $y \sqsubseteq g(\lambda)$. We define g' a constructor in **all** such that $\mathrm{R}(g',h)$ is a finite variation of $\mathrm{R}(g,\widehat{h})$. $g'(\lambda) = w \cdot g(\lambda)[|w|..|g(\lambda)|-1]$. If $w \sqsubseteq x$, then let $z = y \cdot x[|y|..|x|-1]$, and let $g'(x) = x \cdot g(z)[|x|..|g(z)|-1]$. If $w \not\sqsubseteq x$ then $g'(x) = g(x)$. Since h is a winning strategy for player II in the game $G[(X \cap \mathbf{C}_w) \cap \mathrm{R}(\Gamma); \mathbf{all}, \Gamma]$, $\mathrm{R}(g',h) \notin X \cap \mathbf{C}_w$, but $w \sqsubseteq \mathrm{R}(g',h)$ and then $\mathrm{R}(g',h) \notin X$. Since $y \sqsubseteq g(\lambda)$ we always use the first part in the definition of $\widehat{h}$ to compute $\mathrm{R}(g,\widehat{h})$, and thus $\mathrm{R}(g,\widehat{h})$ is the result of substituting w by y as a prefix of $\mathrm{R}(g',h)$. But X is closed under finite variations and since $\mathrm{R}(g,\widehat{h})$ is a finite variation of $\mathrm{R}(g',h)$, then $\mathrm{R}(g,\widehat{h}) \notin X$.

(ii) If $y \not\sqsubseteq g(\lambda)$, then $\mathrm{R}(g,\widehat{h}) \notin \mathbf{C}_y$ by the definition of $\widehat{h}$.

Each of (i) and (ii) implies that $\mathrm{R}(g,\widehat{h}) \notin X \cap \mathbf{C}_y \cap \mathrm{R}(\Gamma)$, so $\widehat{h}$ is indeed a winning strategy for player II. Thus $X \cap \mathbf{C}_y$ is meager in $\mathrm{R}(\Gamma)$ for each y with $|y| = |w|$. But since

$$X = \bigcup_{y \in \{0,1\}^{|w|}} (X \cap \mathbf{C}_y),$$

X is a finite union of sets that are meager in $\mathrm{R}(\Gamma)$, which by Lemma 3.18 implies that X is meager in $\mathrm{R}(\Gamma)$. This completes the proof. ∎

Lemma 3.27. If X is a class of languages that is closed under finite variations and has the property of Baire in $\mathrm{R}(\Gamma)$, then X is either meager in $\mathrm{R}(\Gamma)$ or co-meager in $\mathrm{R}(\Gamma)$.

Proof . Assume that X is closed under finite variations, has the property of Baire in $\mathrm{R}(\Gamma)$, and is not meager in $\mathrm{R}(\Gamma)$. It suffices to prove that X is co-meager in $\mathrm{R}(\Gamma)$.

Since X has the property of Baire in $\mathrm{R}(\Gamma)$, there is a class G that is open in $\mathrm{R}(\Gamma)$ such that $X \Delta G$ is meager in $\mathrm{R}(\Gamma)$. Since X is not meager in $\mathrm{R}(\Gamma)$, $G \neq \emptyset$. Thus there exists $w \in \{0,1\}^*$ such that $\mathbf{C}_w \cap \mathrm{R}(\Gamma) \subseteq G \cap \mathrm{R}(\Gamma)$.

$X^c \cap \mathbf{C}_w$ is meager in $\mathrm{R}(\Gamma)$ because $X^c \cap \mathbf{C}_w \subseteq X^c \cap G \subseteq X \Delta G$. By the last proposition this is equivalent to saying that X^c is meager in $\mathrm{R}(\Gamma)$. This completes the proof. ∎

The following theorem thus summarizes the results of this section.

Theorem 3.28. The class of P-bi-immune languages does not have the property of Baire in E, E_2, ESPACE, or REC.

Proof . This follows from Theorems 3.19 and 3.20 (extended to the classes E, E_2 and ESPACE), Theorem 3.22 and Lemma 3.27. ∎

In contrast with Theorem 3.28, it is easy to see that the class of RE-bi-immune languages is **all**-co-meager, so P-bi-immunity is co-meager in the classical Baire category sense.

From Theorem 3.28 and the remark following Lemma 3.9 in [Fenn] we note that we cannot assume anything about the immunity of a pseudo-generic language.

- There exists a p-generic language in E_2 that is E-bi-immune.
- There exists a p-generic language in E_2 that is not P-immune.

We can define a category in PSPACE from the notion of plogon-category, simply by using Definitions 3.13 to 3.16. For this category we can study the class of DLOG-bi-immune sets in PSPACE, and prove the following results that are analogous to Theorems 3.19, 3.20 and 3.28.

Theorem 3.29. The class of DLOG-bi-immune languages is neither meager nor co-meager in PSPACE. Thus it does not have the property of Baire in PSPACE.

The proof is a translation of those of Theorems 3.19, 3.20 and 3.28 to plogon bounds. This, together with Theorem 3.9, witnesses the differences of measure and category in PSPACE.

Chapter 4: Measure of nonuniform complexity classes

4.1 Introduction

The models of computation can be classified into uniform and nonuniform. In a uniform model, programs are valid for arbitrarily long inputs, while in a nonuniform one each program is valid only for inputs of a fixed length. Examples of uniform models are the Turing Machine and the RAM, and the main example of nonuniform model is the Boolean Circuit.

Nonuniform complexity classes are defined in connection with nonuniform complexity models and the associated complexity measures. In this context appears the concept of advice class, introduced by Karp and Lipton in [KarpLi]. An advice class is defined by adding nonuniform advice to a uniform complexity class. In section 4.2 we review the concept of advice class and study P/log, the class of languages recognized in polynomial time with the help of a logarithmic advice. The characterization of P/log that we use in this section is a part of the material in [HermMa] and [BalcHeM].

P/poly is the most thoroughly studied nonuniform class, defined as those languages that can be recognized in P with the help of a polynomially long advice. P/poly can also be characterized as the class of languages decidable by boolean circuits whose size is polynomial on the length of the input (see [BalcDíG] and the references there for a full study of this class).

Regarding the comparison of uniform and nonuniform complexity classes, we know that P is a subclass of P/poly [Sava], and that ESPACE is not included in P/poly [Kann]. What is more, Lutz has shown in [Lutz92] that P/poly has measure 0 in ESPACE, that is to say, most languages in ESPACE are out of P/poly.

The main open problem in this context is the relationship of P/poly with E. There exists an oracle A for which E^A is not included in P^A/poly (for instance, any oracle for which E equals ESPACE), and there exists an oracle B for which E^B is included in P^B/poly [Wils]. This means that if we want to settle whether E is a subclass of P/poly we must use nonrelativizing techniques. Thus our work deals with a subclass of P/poly and with a superclass of E.

There is a characterization of P/poly as the class of languages that are Turing reducible in polynomial time to a sparse set, where A is *sparse*, and we write $A \in$ SPARSE, if there is a polynomial p such that $|A_{\leq n}| \leq p(n)$ for all $n \in \mathbb{N}$. It is also known that the weaker truth-table reducibility can be substituted for the Turing reducibility in this characterization. Such result suggests the reformulation of the question of whether E is included in P/poly to 'how dense must a language $A \subseteq \{0,1\}^*$ be in order to be hard for E?' (see [HemaOgW] for a thorough survey). As a consequence of the cited result for P/poly, we cannot expect to solve this question with relativizable techniques for polynomial-time Turing reducibility. The first result on the density of hard languages was the following.

Let us say a language A is *dense*, and we write $A \in \mathrm{DENSE}$, if there is a real number $\epsilon > 0$ such that $|A_{\leq n}| \geq 2^{n^\epsilon}$ for all sufficiently large $n \in \mathbb{N}$. It is clear that $\mathrm{SPARSE} \subset \mathrm{DENSE}^c$.

Theorem 4.1. [Meye]. Every $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard language for E(or any larger class) is dense. That is, $\mathrm{E} \not\subseteq \mathrm{P_m}(\mathrm{DENSE}^c)$.

Theorem 4.1 was subsequently improved to truth-table reducibility with $O(\log n)$ queries

Theorem 4.2. [Wata87a], [Wata87c]. Every $\leq_{O(\log n)-\mathrm{tt}}^{\mathrm{p}}$-hard language for E is dense. That is, $\mathrm{E} \not\subseteq \mathrm{P}_{O(\log n)-\mathrm{tt}}(\mathrm{DENSE}^c)$.

The Main Theorem in section 4.4, Theorem 4.17, extends Theorems 4.1 and 4.2 above by showing that, for every real $\alpha < 1$ (e.g., $\alpha = 0.99$), only a measure 0 subset of the languages in E are $\leq_{n^\alpha-\mathrm{tt}}^{\mathrm{p}}$-reducible to non-dense languages, that is

$$\mu(\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c) \mid \mathrm{E}) = 0. \tag{4.1}$$

This means that $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c) \cap \mathrm{E}$ is a *negligibly small* subset of E.

In particular, this implies that

$$\mathrm{E} \not\subseteq \mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c), \tag{4.2}$$

i.e., that every $\leq_{n^\alpha-\mathrm{tt}}^{\mathrm{p}}$-hard language for E is dense. This strengthens Theorem 4.2 above by extending the truth table reducibility from $O(\log n)$ queries to n^α queries ($\alpha < 1$). Very recently, and independently, Fu [Fu] has used resource bounded Kolmogorov complexity to prove that for every $\alpha < \frac{1}{4}$, $\mathrm{E} \not\subseteq \mathrm{P}_{n^\alpha-\mathrm{T}}(\mathrm{DENSE}^c)$, which generalizes (4.2) to Turing reducibilities instead of truth-table reducibilities, although with a slightly worse query bound.

It is worth noting that the combinatorial technique used to prove (4.1) and (4.2)—the *sequentially most frequent query selection*—is simpler than Watanabe's direct proof of Theorem 4.2. This is not surprising, once one considers that our proof of (4.2) via (4.1) is a resource-bounded instance of the probabilistic method reviewed in Chapter 1, which exploits the fact that it is often easier to prove the *abundance* of objects of a given type than to construct a *specific* object of that type.

The proof of Theorem 4.17, is based on a very general result, the Weak Stochasticity Theorem proven in section 4.3. In very brief terms, this result says that almost every language in E is "weakly stochastic", in the sense that it is statistically unpredictable by feasible deterministic algorithms, even with linear nonuniform advice. (See section 4.3 for precise definitions.) This result enables us to prove Theorem 4.17, that

$$\mu(\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c) \mid \mathrm{E}) = 0$$

for all $\alpha < 1$, by a simple combinatorial technique, without reference to measure-theoretic notions. Specifically, in section 4.4 below, this combinatorial technique–the *sequentially most frequent query selection*–is introduced and used to prove that no language in

$\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$ is weakly stochastic. Theorem 4.17 follows immediately from this and the Weak Stochasticity Theorem.

This use of weak stochasticity in E is analogous to earlier uses of space-bounded Kolmogorov complexity in ESPACE. It is known ([Lutz92], [JuedLu92]) that almost every language in ESPACE has very high space-bounded Kolmogorov complexity. Using this fact, a variety of sets X have been shown to have measure 0 in ESPACE, simply by proving that every element of X has low space-bounded Kolmogorov complexity ([Lutz91b], [Lutz92], [LutzSc], [JuedLu92]). Informally, we say that high space-bounded Kolmogorov complexity is a "general-purpose randomness property of languages in ESPACE". This expression, which is heuristic, means the following two things.

(a) Almost every language in ESPACE has the property (high space-bounded Kolmogorov complexity).

(b) It is often the case that, when one wants to prove a result of the form $\mu(X|\mathrm{ESPACE}) = 0$, it is convenient to prove that no language in X has the property, and then appeal to (a).

It is natural to hope that high time-bounded Kolmogorov complexity would be, in the analogous sense, a general-purpose property of languages in E. Unfortunately, however, the strongest known lower bound on time-bounded Kolmogorov complexity in this class [Lutz 92] is far too weak to provide a useful time-bounded analogue of condition (a) above. Moreover, improving these bounds appears to require a major breakthrough in complexity theory.

Our results suggest that, even without such a breakthrough, weak stochasticity may be a "general-purpose randomness property of languages in E". This would entail the following two heuristic conditions.

(a') Almost every language in E is weakly stochastic.

(b') It is often the case that, when one wants to prove a result of the form $\mu(X|\mathrm{E}) = 0$, it is convenient to prove that no language in X is weakly stochastic, and then appeal to (a').

The Weak Stochasticity Theorem gives us condition (a') immediately. The proof of Theorem 4.17 gives us the *instance* $X = \mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$ of condition (b'). It appears likely that more such instances will arise, i.e., that weak stochasticity is a general-purpose randomness property of languages in E that will be useful in future investigations. Sections 4.3 and 3.4 are contained in [LutzMa94a].

In section 4.5, we study the size of P/poly inside the exponential time hierarchy. Kannan showed in [Kann] that there exists a language in the second level of the exponential hierarchy that is not in P/poly. We define a measure in each level of the exponential hierarchy and show that P/poly has measure 0 in the third level. This last result is unpublished.

4.2 Advice complexity classes

In this section we review the definition of advice complexity classes and show that P/log has measure 0 in E, that is, almost every language in E is not in P/log. To prove this measure result we use a circuit-based characterization of P/log given by Hermo and Mayordomo in [HermMa].

The notion of advice function was introduced in [KarpLi] to provide connections between uniform and nonuniform computation models.

Definition 4.3. Given a class of sets C and a class of bounding functions F, the class C/F is formed by the sets A such that

$$\forall n\ \exists w\ (|w| \leq h(n))\ \forall x\ (|x| = n)\ x \in A \iff \langle x, w\rangle \in B$$

where $B \in C$ and $h \in F$.

The words w mentioned in the definition are frequently called "advice words". The corresponding Skolem function mapping each n into an appropriate advice w_n for length n is called "advice function". Thus we define

Definition 4.4. Given a language $A \subseteq \{0,1\}^*$ and a function $f\colon \mathbb{N} \to \{0,1\}^*$, we define the language A/f ("A with advice f") by

$$A/f = \{x \in \{0,1\}^* \ \big|\ \langle x, f(|x|)\rangle \in A\}.$$

In Definition 4.3, C is usually a uniform complexity class, most frequently P, whereas the class poly $= \{n^{O(1)}\}$ of polynomials and the class log $= O(\log n)$ of logarithms are the most frequent bounding functions. In particular, [KarpLi] focused on the study of the classes P/poly and P/log, and proved that for certain problems, the hypothesis of being in nonuniform classes has implications on the structure of uniform classes.

The class P/poly can be characterized in various manners, one of them as the class of languages decidable by boolean circuits whose size grows polynomially on the length of the input. This important class has been studied in depth (see for instance [BalcDíG] and the references there). On the contrary, the class P/log, corresponding to polynomial time with the help of a logarithmically long advice string, has received up to now much less attention. We address here the question of finding a characterization of the class P/log in terms of circuits.

We first settle our notions of Boolean circuit and size of a boolean circuit. We fix the following family of boolean functions: the 0-ary functions "true" and "false"; the unary function of negation, denoted $\neg$; and the binary functions $\wedge$ and $\vee$. With these functions as basis, we can compute any n-ary boolean function.

Definition 4.5. Given a set $\delta = \{x_1, x_2, \ldots, x_n\}$ of n boolean variables, *a computation chain over* δ is a sequence $g_1, g_2, \ldots, g_k$, in which each g_j is

- either a source element: an element of δ or a boolean constant,

- or a gate: a pair $(\neg, g_l)$ for $1 \leq l \leq j-1$, or a triple (b, g_l, g_m) for $1 \leq l, m \leq j-1$ and $b \in \{\vee, \wedge\}$. The inputs to a gate g_j are the (one or two) smaller elements of the computation chain which appear in g_j.

With each element g_j of a computation chain over $\delta = \{x_1, x_2, \ldots, x_n\}$, we can associate an n-ary function $result(g_j)$ which represents the boolean value computed at element g_j. It is defined as follows.

Definition 4.6.

$$result(g_j) = \begin{cases} g_j & \text{if } g_j \text{ is a source} \\ \neg result(g_l) & \text{if } g_j = (\neg, g_l) \\ result(g_l) \vee result(g_m) & \text{if } g_j = (\vee, g_l, g_m) \\ result(g_l) \wedge result(g_m) & \text{if } g_j = (\wedge, g_l, g_m). \end{cases}$$

Representing computation chains by acyclic graphs, circuits are obtained

Definition 4.7. *A boolean circuit* is an acyclic graph representation of a computation chain, which is constructed by associating to each step g_j of the chain a node labeled with the variable, constant, or function present in g_j , and joining node g_j to node g_i by a directed edge if g_j is an input to g_i.

Definition 4.8. Given a boolean function $f\colon \Sigma^n \to \Sigma^m$, which can be expressed as an m-tuple $f = (f_1, \ldots, f_m)$ of boolean functions from Σ^n to Σ, we say that *a circuit C computes* f if for every $r, 1 \leq r \leq m$, there exists an s, such that $f_r = result(g_s)$.

The *cost or size* of a circuit is the number of gates it has. Given a boolean function f, its boolean cost is the size of the smallest circuit computing it. A set A has polynomial size circuits if and only if there is a polynomial p such that for each length n the boolean cost of the characteristic function of $A^{=n}$ is bounded by $p(n)$.

In our characterization of P/log we use the next theorem from Savage that relates size complexity and the time of a Turing machine computation.

Theorem 4.9. *[Sava].* If a function f is computed by a Deterministic Turing Machine (DTM) in time $T(n)$, then the restriction of f to $\{0,1\}^n$ can be computed by a circuit of size $O(T(n)^2)$.

We will also use the following result stating that the evaluation of a circuit on a given input can be done in time bounded by a polynomial in the size of the circuit.

Definition 4.10. The *circuit value problem*, CVP for short, is the set of all pairs $\langle x, y\rangle$ where $x \in \Sigma^*$ and y encodes a circuit with $|x|$ source elements which outputs 1 on input x.

It is known that CVP$\in$ P and, as a consequence of Theorem 4.9, it was shown that CVP is complete for P under very weak reductions [Ladn]. In [HermMa] we obtain the following characterization of P/log in terms of resource-bounded Kolmogorov complexity.

Theorem 4.11. The following are equivalent:

(i) $A \in$ P/ log.

(ii) A is accepted by a family $\{C_n\}$ of polynomial size circuits such that $\{C_n\} \in K[\log, \text{poly}]$.

Proof .

i/ $\Rightarrow$ ii/:

Let $A \in$ P/ log. This is to say, we have $B \in$P such that $\exists c\ \forall n\ \exists w_n$ such that $\forall x(|x| = n), x \in A$ iff $\langle x, w_n\rangle \in B$, with $|w_n| \leq c \log n$.

Using Theorem 4.9 it is possible to construct the polynomial size circuit C_n that recognizes $A^{=n}$ as follows. For each x, y the pairing function $\langle x, y\rangle$ is polynomial time computable. We take the circuit that computes it according to Theorem 4.9. The inputs corresponding to y are then fixed to the bits of $w_{|x|}$.

Next, since B ∈P, another circuit generated in the same way simulates the DTM that recognizes B for inputs of size $|\langle x, w_{|x|}\rangle|$.

Composing both circuits we obtain C_n.

An interesting characteristic of the circuits constructed in the proof of Theorem 4.9 is that they are generated having a very regular interconnection pattern, in fact from the number of a gate we know its exact position in the circuit —a full construction is given in Balcázar, Díaz, and Gabarró [BalcDíG]—.

The gates of C_n are of two types, namely the constant gates corresponding to the advice and the gates produced by Savage's simulation of a DTM, with a regular interconnection pattern. These gates can be codified placing the constant gates first, so that the two types can be distinguished easily. This is why C_n can be generated by a DTM in $O(p(n)^2)$ time for a polynomial p, from seeds n and w_n. Thus $\{C_n\} \in$ K[log, poly].

ii/ ⇒ i/:

Let $\{C_n\}$ be the polynomial size family that accepts A. Since $\{C_n\} \in$ K[log, poly], then for all n $\exists w_n$, $|w_n| \leq c \log n$, such that $U(w_n) = C_n$ in time bounded by n^k. We define $B = \{\langle x, y\rangle / \langle x, z\rangle \in$ CVP, where $z = U(y)$ in time $\leq |z|^k\}$.

As CVP is in P it is clear that B is in P. Finally, $x \in A$ iff $\langle x, w_{|x|}\rangle \in B$. ∎

Karp and Lipton made an attempt of characterizing P/log in terms of Boolean circuits in [KarpLi]. With this purpose they introduce the concept of “small circuits with easy descriptions” where an easy description of a circuit C is again a circuit of size logarithmic in the size of C that describes the interconnection pattern of C. As proven in [HermMa], the family of languages that have “small circuits with easy descriptions” corresponds exactly to the class P/$O(\log n * \log(\log n))$, which can be proven to be different from P/log. Thus Theorem 4.11 is the first known characterization of P/log in terms of circuits.

This characterization is useful to give an elegant proof of the fact that P/log has measure 0 in E.

Theorem 4.12. P/log has measure 0 in E.

Proof . The idea is that we can only produce $2^{\log^2 n}$ circuits from seeds of length $\log^2 n$, and thus a p-martingale with input of length 2^n has enough time to compute all the different n-input $n^{\log n}$-size circuits in $K[\log^2 n, n^{\log n}]$, which includes all polynomial size circuits in $K[O(\log n), poly]$, for n large enough.

Let $k \in \mathbb{N}$ be a power of 2, $k = 2^i$. We define

$$X_k = \{A \mid \text{for each } n > i, \text{ there exists a } n^{\log n}\text{-size boolean circuit} \\ C_n \in K[\log^2 n, n^{\log n}] \text{ that recognizes } A^{=n}\}.$$

By Theorem 4.11, we know that $\mathrm{P}/\log \subseteq \bigcup\limits_{k=1}^{\infty} X_k$. We use the Γ-additivity Lemma (Lemma 1.35) to show that $\mu_{\mathrm{p}}(\bigcup_k X_k) = 0$.

The function $f\colon \mathbb{N} \times \{0,1\}^* \to \mathbb{N}$ counts the number of $n^{\log n}$ circuits in $K[\log^2 n, n^{\log n}]$ and is defined as follows, for $m = 2^n$ and $w \in \{0,1\}^*$:

$$f_m(w) = \#\{C \mid C \text{ is an } n\text{-input boolean circuit of size } n^{\log n} \text{ in } K[\log^2 n, n^{\log n}] \\ \text{such that for each } s_j \in \{0,1\}^n,\ j < |w|,\ C(s_j) = w[j]\}.$$

Clearly f can be computed in polynomial time.

Let us define $d\colon \mathbb{N} \times \{0,1\}^* \to [0,\infty)$ as follows. Let $w \in \{0,1\}^*$, $b \in \{0,1\}$

$$d_k(wb) = \begin{cases} 2^{-k}, & \text{if } |w| < 2k-1; \\ d_k(w)\,\dfrac{2 \cdot f_m(wb)}{f_m(w)}, & \text{otherwise, for } m = 2^n \text{ such that } m-1 \leq |w| < 2m-1. \end{cases}$$

Since we can compute f in polynomial time and $f_m(w0) + f_m(w1) = f_m(w)$ for each m and w, d is clearly a 1-MS in p.

By Lemma 1.35, if we see that $X_k \subseteq \mathrm{S}^{\infty}[d_k]$, we have the theorem. Let $k = 2^i$, $A \in X_k$. From the definition of X_k we know that for any $m > k$, $m = 2^n$, $f_m(\chi_A[0..2^{n+1}-2]) \geq 1$. The fact that $\#K[\log^2 n, n^{\log n}] \leq 2^{\log^2 n}$ implies that for any $w \in \{0,1\}^*$, $f_m(w) \leq 2^{\log^2 n}$. We have then the following inequalities

$$d_k(\chi_A[0..2^{n+1}-2]) \geq d_k(\chi_A[0..2^n-2]) \cdot 2^{(2^n)} \frac{1}{2^{\log^2 n}} \geq d_k(\chi_A[0..2^n-2]) 2^{(2^{n-1})},$$

thus by induction

$$d_k(\chi_A[0..2^{n+1}-2]) \geq d_k(\lambda) \cdot 2^{2^n - k} = 2^{2^n - 2k},$$

which implies that $\limsup_s d_k(\chi_A[0..s]) = \infty$, and $A \in \mathrm{S}[d_k]$. ■

In fact, if $f \in o(n)$ then from [HermMa] we have a circuit based characterization of $\mathrm{P}/O(f)$, and it can be proven that $\mathrm{P}/O(f)$ has measure 0 in E with the same technique as in Theorem 4.12.

4.3 Weak stochasticity

In this section we prove the Weak Stochasticity Theorem. This theorem will be useful in the proof of our main result in section 4.4. We also expect it to be useful in future investigations of the measure structure of E and E_2.

Let us formulate our notion of weak stochasticity.

Definition 4.13. Let $t, q, \nu \colon \mathbb{N} \to \mathbb{N}$ and let $A \subseteq \{0,1\}^*$. Then A is *weakly* (t, q, ν)-*stochastic* if, for all $B \in \mathrm{DTIME}(t)/\{q\}$ and all $C \in \mathrm{DTIME}(t)$ such that $|C_{=n}| \geq \nu(n)$ for all sufficiently large n,

$$\lim_{n\to\infty} \frac{|(A \bigtriangleup B) \cap C_{=n}|}{|C_{=n}|} = \frac{1}{2}.$$

Intuitively, B and C together form a "prediction scheme" in which B tries to guess the behavior of A on the set C. A is weakly (t, q, ν)-stochastic if no such scheme is better in the limit than guessing by random tosses of a fair coin.

Our use of the term "stochastic" follows Kolmogorov's terminology ([KolmUs], [UspeSeS]) for properties defined in terms of limiting frequencies of failure of prediction schemes. The adverb "weakly" distinguishes our notion from a stronger stochasticity property considered in [Lutz94c], but weak stochasticity is a powerful and convenient tool.

The following lemma captures the main technical content of the Weak Stochasticity Theorem.

Lemma 4.14. Fix $c \in \mathbb{N}$ and $0 < \gamma \in \mathbf{R}$ and let

$$WS_{c,\gamma} = \{A \subseteq \{0,1\}^* | A \text{ is weakly } (2^{cn}, cn, 2^{\gamma n}) - \text{ stochastic}\}.$$

Then $\mu_{\mathrm{p}}(WS_{c,\gamma}) = 1$.

Proof. Assume the hypothesis. Let $U \in \mathrm{DTIME}(2^{(c+1)n})$ be a language that is universal for $\mathrm{DTIME}(2^{cn}) \times \mathrm{DTIME}(2^{cn})$ in the following sense. For each $i \in \mathbf{N}$, let

$$\begin{aligned} C_i =& \{x \in \{0,1\}^* | \langle 0^i, 0x\rangle \in U\}, \\ D_i =& \{x \in \{0,1\}^* | \langle 0^i, 1x\rangle \in U\}. \end{aligned}$$

Then $\mathrm{DTIME}(2^{cn}) \times \mathrm{DTIME}(2^{cn}) = \{(C_i, D_i) | i \in \mathbf{N}\}$.

Our objective is to use Lemma 1.47 to prove that $WS^c_{c,\gamma}$, the complement of $WS_{c,\gamma}$, has p-measure 0. In order to do this, for all $i, j, k \in \mathbb{N}$, define the set $Y_{i,j,k}$ of languages as follows. If k is not a power of 2, then $Y_{i,j,k} = \emptyset$. Otherwise, if $k = 2^n$, where $n \in \mathbb{N}$, then

$$Y_{i,j,k} = \bigcup_{z \in \{0,1\}^{\leq cn}} Y_{i,j,k,z},$$

where each

$$Y_{i,j,k,z} = \left\{ A \subseteq \{0,1\}^* \;\middle|\; |(C_i)_{=n}| \geq 2^{\gamma n} \text{ and } \left| \frac{|(A \bigtriangleup (D_i/z)) \cap (C_i)_{=n}|}{|(C_i)_{=n}|} - \frac{1}{2} \right| \geq \frac{1}{j+1} \right\}.$$

(The notation D_i/z here denotes D_i/h, where $h \colon \mathbb{N} \to \{0,1\}^*$ is the constant function $h(n) = z$.) The point of this definition is that, if a language $A \subseteq \{0,1\}^*$ is *not* an element of $WS_{c,\gamma}$, then the definition of weak stochasticity says that there exists $i, j \in \mathbb{N}$ such that $A \in Y_{i,j,k}$ for infinitely many k. That is,

$$WS^c_{c,\gamma} \subseteq \bigcup_{i=0}^{\infty} \bigcup_{j=0}^{\infty} \bigcap_{m=0}^{\infty} \bigcup_{k=m}^{\infty} Y_{i,j,k}.$$

It follows by Lemma 1.47 that it suffices to exhibit a p-computable 2-MS d with the following two properties.

(I) The series $\sum_{k=0}^{\infty} d_{m,k}(\lambda)$, for $m \in \mathbb{N}$, are uniformly p-convergent.

(II) For all $i, j, k \in \mathbb{N}$, $Y_{i,j,k}(w) \subseteq \mathrm{S}^{\frac{1}{d_{\langle i,j\rangle,k}}(\lambda)}[d_{\langle i,j\rangle,k}]$.

Define the function $d\colon \mathbb{N}^2 \times \{0,1\}^* \to [0,\infty)$ as follows. If k is not a power of 2, then $d_{m,k}(w) = 0$. Otherwise, if $k = 2^n$, where $n \in \mathbb{N}$, and $m = \langle i,j\rangle$ then

$$d_{m,k}(w) = \sum_{z \in \{0,1\}^{\leq cn}} \Pr_C[C \in Y_{i,j,k,z} | C \in \mathbf{C}_w].$$

It follows immediately from the definition of conditional probability that d is a 2-MS. Since $U \in \mathrm{DTIME}(2^{(c+1)n})$ and c is fixed, we can use binomial coefficients to (exactly) compute $d_{m,k}(w)$ in time polynomial in $m + k + |w|$. Thus d is p-computable.

To see that d has property (I), note first that the Chernoff bound, Lemma 1.1, tells us that, for all $i, j, k \in \mathbb{N}$ and $z \in \{0,1\}^{\leq cn}$ (writing $k = 2^n$ and $N = k^\gamma = 2^{\gamma n}$),

$$\Pr_C[C \in Y_{i,j,k,z}] \leq 2e^{-\frac{N}{2(j+1)^2}},$$

whence

$$\begin{aligned} d_{\langle i,j\rangle,k}(\lambda) &= \sum_{z \in \{0,1\}^{\leq cn}} \Pr_C[C \in Y_{i,j,k,z}] \\ &\leq 2^{cn+1} \cdot 2e^{-\frac{N}{2(j+1)^2}} \\ &< e^{cn+2-\frac{N}{2(j+1)^2}}. \end{aligned}$$

Let $a = \left\lceil \frac{1}{\gamma} \right\rceil$, let $\delta = \frac{\gamma}{4}$, and fix $k_0 \in \mathbb{N}$ such that

$$k^{2\delta} \geq k^\delta + c \log k + 2$$

for all $k \geq k_0$. Define $g\colon \mathbb{N} \to \mathbb{N}$ by

$$g(j) = 4^a (j+1)^{4a} + k_0.$$

Then g is a polynomial and, for all $i, j, n \in \mathbb{N}$ (writing $k = 2^n$ and $N = k^\gamma = k^{4\delta}$),

$$k \geq g(j) \implies \begin{cases} N = k^{2\delta} k^{2\delta} \\ \quad \geq [4^a (j+1)^{4a}]^{2\delta} (k^\delta + c \log k + 2) \\ \quad \geq 2(j+1)^2 (k^\delta + cn + 2) \end{cases} \implies d_{\langle i,j\rangle,k}(\lambda) < e^{-k^\delta}.$$

Thus $d_{\langle i,j\rangle,k}(\lambda) < e^{-k^\delta}$ for all $i, j, k \in \mathbb{N}$ such that $k \geq g(j)$. Since $\delta > 0$, it follows by Lemma 1.48 that (I) holds.

Finally, to see that (II) holds, fix $i, j, k \in \mathbb{N}$. If k is not a power of 2, then (II) is trivially affirmed, so assume that $k = 2^n$, where $n \in \mathbb{N}$. Let $A \in Y_{i,j,k}$. Fix $z \in \{0,1\}^{\leq cn}$ such that $A \in Y_{i,j,k,z}$ and let w be the $(2^{n+1} - 1)$-bit characteristic string of $A^{\leq n}$. Then

$$d_{\langle i,j\rangle,k}(w) \geq \Pr_C[C \in Y_{i,j,k,z} | C \in \mathbf{C}_w] = 1,$$

so $A \in \mathbf{C}_w \subseteq \mathrm{S}^{\frac{1}{d_{\langle i,j\rangle,k}(\lambda)}}[d_{\langle i,j\rangle,k}]$. This completes the proof of Lemma 4.14. ∎

We now have the main theorem of this section.

Theorem 4.15. Weak Stochasticity Theorem.

(1) For each $c \in \mathbb{N}$ and $\gamma > 0$, $WS_{c,\gamma}$ has measure 1 in E; that is, for each $c \in \mathbb{N}$ and $\gamma > 0$, almost every language $A \in \mathrm{E}$ is weakly $(2^{cn}, cn, 2^{\gamma n})$-stochastic.

(2) $\bigcap_{c,\gamma} WS_{c,\gamma}$ has measure 1 in E_2, that is, almost every language $A \in \mathrm{E}_2$ is, for every $c \in \mathbb{N}$ and $\gamma > 0$, weakly $(2^{cn}, cn, 2^{\gamma n})$-stochastic.

Proof. Part (1) follows immediately from Lemma 4.14 via Lemma 1.21. Part (2) follows from Lemma 4.14 via Lemmas 1.40 and 1.21. ∎

4.4 Measure of $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$

In this section we show that for every real $\alpha < 1$, the set $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$ has measure 0 in E and in E_2.

Our proof is based on the Weak Stochasticity Theorem from the last section, stating that almost every language in E and almost every language in E_2 is weakly stochastic. We give a simple combinatorial proof that *no* language in $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$ is weakly $(2^{3n}, 3n, 2^{\frac{1}{2}n})$-stochastic, thereby proving that $\mu(\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c) \mid \mathrm{E}) = 0$.

The sequentially most frequent query selection is the main construction in the combinatorial proof of Lemma 4.16. Let f be an n^α-query function and let $n \in \mathbb{N}$. Our objective is to obtain a set $S \subseteq \{0,1\}^n$ of a reasonable size such that there exist as many as possible queries that are made by f on every input in S. For this we construct $S_0, \ldots, S_{n^\alpha} \subseteq \{0,1\}^n$ and $y_0, \ldots, y_{n^\alpha - 1} \in \{0,1\}^*$ such that for every $k \leq n^\alpha$, $y_0, \ldots, y_{k-1}$ are queries of f on every string in S_k.

Let us formalize. Given an n^α-query function f and $n \in \mathbb{N}$, the *sequentially most frequent query selection (smfq selection)* for f on inputs of length n is the sequence

$$(S_0, Q_0, y_0), (S_1, Q_1, y_1), \ldots, (S_{n^\alpha}, Q_{n^\alpha}, y_{n^\alpha})$$

defined as follows. Each $S_k \subseteq \{0,1\}^n$. Each Q_k is an $|S_k| \times n^\alpha$ matrix of strings, with each string in Q_k colored either green or red. The rows of Q_k are indexed lexicographically by the elements of S_k. For $x \in S_k$, row x of Q_k is the sequence $f_1(x), \ldots, f_{n^\alpha}(x)$ of queries of f on input x. If Q_k contains at least one green string, then y_k is the green string occurring in the greatest number of rows of Q_k. (Ties are broken lexicographically.) If Q_k is entirely red, then $y_k = \perp$ ("bottom," i.e., undefined). The sets S_k and the coloring are specified recursively. We set $S_0 = \{0,1\}^n$ and color all strings in Q_0 green. Assume that S_k, Q_k, and y_k have been defined, where $0 \le k < n^\alpha$. If $y_k = \perp$, then $(S_{k+1}, Q_{k+1}, y_{k+1}) = (S_k, Q_k, y_k)$. If $y_k \ne \perp$, then S_{k+1} is the set of all $x \in S_k$ such that y_k appears in row x of Q_k. The strings in Q_{k+1} are then colored exactly as they were in Q_k, except that all y_k's are now colored red. This completes the definition of the smfq selection.

For $0 \le k \le n^\alpha$, it is clear that every row of Q_k contains at least k red strings. In particular, the matrix Q_{n^α} is entirely red.

Our main results follow from the following lemma.

Lemma 4.16. For every real $\alpha < 1$, $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c) \cap WS_{3,\frac{1}{2}} = \emptyset$.

Proof. Let $\alpha < 1$ and assume that $A \le^{\mathrm{p}}_{n^\alpha-\mathrm{tt}} L$ via (f,g), where $L \notin \mathrm{DENSE}$. It suffices to show that $A \notin WS_{3,\frac{1}{2}}$. Intuitively, in order to do this we consider a language C such that for each $n \in \mathbb{N}$ there exist several queries that are made by f for all inputs in $C^{=n}$. We will construct C using the smfq selection. We then want to predict $A \cap C$, for which we use a language $B \in \mathrm{DTIME}(2^{3n})/\{3n\}$, that is obtained from (f,g) by answering according to the advice to those queries that appear more often and answering no to the rest of the queries. This prediction scheme of A will work well because of the low density of the oracle, as proven below.

Fix a polynomial p such that $|f_i(x)| \le p(|x|)$ for all $x \in \{0,1\}^*$ and $1 \le i \le |x|^\alpha$. Let $\epsilon = \frac{1-\alpha}{4}$ and fix $n_0 \in \mathbb{N}$ such that the following conditions hold for all $n \ge n_0$.

$$\text{(i) } n \ge 2 \cdot n^{1-2\epsilon}.$$
$$\text{(ii) } n^{2\epsilon} - n^\epsilon \ge 2.$$

Let

$$K = \{n \in \mathbb{N} \big| n \ge n_0 \text{ and } |L_{\le p(n)}| < 2^{n^\epsilon}\}.$$

Note that K is infinite because L is not dense.

Define languages B, C, D and an advice function $h\colon \mathbb{N} \to \{0,1\}^*$ as follows. $C = \bigcup_n C_n$, $D = \bigcup_n D_n$. For all $n < n_0$, $C_n = D_n = \{0,1\}^n$ and $h(n) = \lambda$. For all $n \ge n_0$, C_n, D_n, and $h(n)$ are defined from the smfq selection for f on inputs of length n as follows: Let $k = k(n)$ be the greatest integer such that $0 \le k \le n^\alpha$ and $|S_k| \ge 2^{n-kn^{2\epsilon}}$. (Note that k exists because $|S_0| = 2^n$.) We then define

$$C_n = S_k,$$
$$h(n) = [\![y_0 \in L]\!] \ldots [\![y_{k-1} \in L]\!],$$

and we let D_n be the set of all coded pairs $\langle x, z\rangle$ such that $x \in S_k$, $z \in \{0,1\}^k$, and $g(x)(b_1 \ldots b_{n^\alpha}) = 1$, where each

$$b_i = \begin{cases} z[j] & \text{if } f_i(x) = y_j,\ 0 \le j < k, \\ 0 & \text{if } f_i(x) \notin \{y_0, \ldots, y_{k-1}\}. \end{cases}$$

Finally, we let $B = D/h$. For each $n \geq n_0$ and each $x \in C_n = S_k$, the bit $[\![x \in B]\!]$ is a "guessed value" of the bit $[\![x \in A]\!]$. The actual value, given by the reduction (f, g) to L, is

$$[\![x \in A]\!] = g(x)([\![w_i \in L]\!] \ldots [\![w_{n^\alpha} \in L]\!]),$$

where $w_1, \ldots, w_{n^\alpha}$ are the entries in row x of the matrix Q_k. The guessed value $[\![x \in B]\!] = g(x)(b_1 \ldots b_{n^\alpha})$ uses the advice function h to get the *correct* bit $b_i = [\![w_i \in L]\!]$ when the string w_i is red in Q_k, and *guesses* that $w_i \notin L$ when the string w_i is green in Q_k.

To construct C we just have to perform the smfq selection, by listing all strings of length n and their corresponding queries, and keeping a list of the repetitions according to the definition of (S_j, Q_j, y_j) in the smfq selection. This can be done in time $2^{2n}p(n)$, which implies that $C \in \mathrm{DTIME}(2^{3n})$. An algorithm for D on input $\langle x, z\rangle$ checks whether $x \in C$ in time $2^{2n}p(n)$ and then computes $y_0, \ldots, y_{k-1}$ in time $2^{2n}p(n)$. The algorithm finishes with the simulation of $g(x)$, with total time less than 2^{3n}. Thus $C \in \mathrm{DTIME}(2^{3n})$ and $B \in \mathrm{DTIME}(2^{3n})/\{n^\alpha\} \subseteq \mathrm{DTIME}(2^{3n})/\{3n\}$.

Also, by condition (i) in our choice of n_0,

$$|C_n| \geq 2^{n - n^\alpha n^{2\epsilon}} \geq 2^{\frac{n}{2}}$$

for all $n \geq n_0$, whence $|C_n| \geq 2^{\frac{n}{2}}$ for all $n \in \mathbb{N}$.

We now show that B does a good job of predicting A on C_n, for all $n \in K$. Let $n \in K$. We have two cases.

(I) If $k = k(n) = n^\alpha$, then all strings in Q_k are red, so *all* the guesses made by B are correct, so

$$|(A \bigtriangleup B) \cap C_n| = 0.$$

(II) If $k = k(n) < n^\alpha$, let r be the number of rows in Q_k, i.e., $r = |S_k| = |C_n|$. By our choice of k, we have

$$|S_{k+1}| \leq 2^{n-(k+1)n^{2\epsilon}} \leq 2^{-n^{2\epsilon}} r.$$

That is, no green string appears in more than $2^{-n^{2\epsilon}} r$ of the rows of Q_k. Moreover, since $|L_{\leq p(n)}| \leq 2^{n^\epsilon}$, there are at most 2^{n^ϵ} different strings w in Q_k such that $w \in L$. Thus there are at most $2^{n^\epsilon} \cdot 2^{-n^{2\epsilon}} r = 2^{n^\epsilon - n^{2\epsilon}} r$ rows of Q_k in which B makes an incorrect guess that a green string is not in L; the guesses made by B are correct in all other rows! By condition (ii) in our choice of n_0, then, B is incorrect in at most $\frac{1}{4}r$ rows of Q_k. That is,

$$|(A \bigtriangleup B) \cap C_n| \leq \frac{1}{4} r.$$

In either case, (I) or (II), we have

$$|(A \bigtriangleup B) \cap C_n| \leq \frac{1}{4}|C_n|.$$

Since this holds for all $n \in K$, and since K is infinite,

$$\frac{|(A \bigtriangleup B) \cap C_n|}{|C_n|} \not\to \frac{1}{2}.$$

Thus B and C testify that A is not weakly $(2^{3n}, 3n, 2^{\frac{n}{2}})$-stochastic, i.e., that $A \notin WS_{3,\frac{1}{2}}$. ■

Our main results of this chapter are now easily derived. We start with the fact that most languages decidable in exponential time are not $\leq^{\mathrm{P}}_{n^\alpha-\mathrm{tt}}$-reducible to non-dense languages.

Theorem 4.17. For every real number $\alpha < 1$,

$$\mu_{\mathrm{p}}(\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)) = \mu(\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c) \mid \mathrm{E}) = \mu(\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c) \mid \mathrm{E}_2) = 0.$$

Proof . This follows immediately from Theorem 4.15 and Lemma 4.16. ■

The Main Theorem yields the following separation result.

Theorem 4.18. For every real $\alpha < 1$,

$$\mathrm{E} \not\subseteq \mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c).$$

That is, every $\leq^{\mathrm{P}}_{n^\alpha-\mathrm{tt}}$-hard language for E is dense.

Proof . By the Measure Conservation Theorem (Theorem 1.18), $\mu(\mathrm{E} \mid \mathrm{E}) \neq 0$, so this follows immediately from Theorem 4.17. ■

Note that Theorem 4.18 strengthens Theorem 4.2 by extending the number of queries from $O(\log n)$ to n^α, where $\alpha < 1$ (e.g., $\alpha = 0.99$).

We can generalize the concept of dense language and obtain the following generalizations of Lemma 4.16, Theorem 4.17 and Theorem 4.18, where the number of queries we allow in the truth-table reduction is related to the density of the oracles.

Definition 4.19. Let $f\colon \mathbb{N} \to \mathbb{N}$ with $f \in o(n)$. The class DENSE_f contains those languages A such that there exists $\epsilon > 0$ such that for almost every n, $|A^{\leq n}| \geq 2^{f(n)^\epsilon}$.

Notice that for each $c > 0$, $\mathrm{DENSE}_{n^c} = \mathrm{DENSE}$.

Theorem 4.20. For every $f \in o(n)$, $\mathrm{P}_{n/f(n)-\mathrm{tt}}(\mathrm{DENSE}^c_f) \cap WS_{3,\frac{1}{2}} = \emptyset$. Thus

$$\mu(\mathrm{P}_{n/f(n)-\mathrm{tt}}(\mathrm{DENSE}^c_f) \mid \mathrm{E}) = \mu(\mathrm{P}_{n/f(n)-\mathrm{tt}}(\mathrm{DENSE}^c_f) \mid \mathrm{E}_2) = 0,$$

and

$$\mathrm{E} \not\subseteq \mathrm{P}_{n/f(n)-\mathrm{tt}}(\mathrm{DENSE}^c_f).$$

That is, every $n/f(n)$-tt-hard language for E is in DENSE_f.

The proof uses the smfq selection technique in the same way as Lemma 4.16.

In particular, for the class of sparse sets we have

Corollary 4.21. For every $g \in o(n/\log n)$ the following holds

$$\mu(\mathrm{P}_{g(n)-\mathrm{tt}}(\mathrm{SPARSE}) \mid \mathrm{E}) = \mu(\mathrm{P}_{g(n)-\mathrm{tt}}(\mathrm{SPARSE}) \mid \mathrm{E}_2) = 0,$$

and

$$\mathrm{E} \not\subseteq \mathrm{P}_{g(n)-\mathrm{tt}}(\mathrm{SPARSE}).$$

Therefore, no sparse set is $g(n)$-tt-hard for E.

It is worthwhile to examine the roles played by various methods. Theorem 4.17, a measure-theoretic result concerning the *quantitative* structure of E and E_2, yields the *qualitative* separation result Theorem 4.18. From a technical standpoint, this proof of Theorem 4.18 has the following three components.

(i) The sequentially most frequent query selection (Lemma 4.16). This is used to prove that every language in $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$ is predictable, i.e., fails to be weakly stochastic (with suitable parameters).

(ii) The Weak Stochasticity Theorem (Theorem 4.15). This shows that only a measure 0 subset of the languages in E are predictable.

(iii) The Measure Conservation Theorem (Theorem 1.18). This shows that E is not a measure 0 subset of itself.

Of these three components, (ii) and (iii) are general theorems concerning measure in E. Only component (i) is specific to the issue of the densities of $\leq^{\mathrm{p}}_{n^\alpha-\mathrm{tt}}$-hard languages. That is, *given the general principles* (ii) and (iii), the proof of Theorem 4.18 is just the sequentially most frequent query selection, i.e., the proof of Lemma 4.16. The latter proof is combinatorially much simpler than Watanabe's direct proof of Theorem 4.2. This is not surprising, once it is noted that our proof of Theorem 4.18 is an application of (a resource-bounded generalization of) the *probabilistic method*, ([Erdö], [Shan48], [Shan49], [ErdöSp], [Spen], [AlonSp]) which exploits the fact that it is often easier to establish the *abundance* of objects of a given type than to construct a *specific* object of that type. Much of our proof of Theorem 4.18 is "hidden" in the power of this method (i.e., in the proofs of the Measure Conservation and Weak Stochasticity Theorems), freeing us to apply the sequentially most frequent query selection to the problem at hand.

An important feature of this general method is that it is *uniformly constructive* in the following sense. Taken together, the proofs of the Measure Conservation and Weak Stochasticity Theorems give a straightforward, "automatic" construction of a language $A \in \mathrm{E} \cap \mathrm{WS}_{3,\frac{1}{2}}$. By Lemma 4.16, it follows immediately that $A \in \mathrm{E} \backslash \mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$. Thus one can apply this complexity-theoretic version of the probabilistic method with complete assurance that the resulting existence proof will automatically translate into a construction.

Remember though that the primary objective of resource-bounded measure theory is to give a detailed account of the *quantitative structure* of E, E_2 and other complexity classes. The derivation of *qualitative* separation results, such as Theorems 4.18 and 4.2, is only a by-product of this quantitative objective. (By analogy, the value of classical Lebesgue measure and probability far surpasses their role as tools for existence proofs.) In the case of E, for example, the quantitative content of Theorem 4.17 is that the set $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c) \cap \mathrm{E}$ is a *negligibly small* subset of E.

The density criterion in Theorem 4.17 cannot be improved, since using padding it can be shown that for every $\epsilon > 0$ there is a language $A \in \mathrm{E}$ that is $\leq^{\mathrm{p}}_{\mathrm{m}}$-hard for E_2 and satisfies $|A_{\leq n}| < 2^{n^\epsilon}$ for all n. It is an open question whether the query bound n^α can be significantly relaxed. A construction of Wilson [Wils] shows that there is an oracle B such that $\mathrm{E}^B \subseteq \mathrm{P}^B_{O(n)-\mathrm{tt}}(\mathrm{SPARSE})$, so progress in this direction will require nonrelativizable techniques. (The proof of Theorem 4.17 relativizes in a straightforward manner.)

4.5 P/poly inside the Exponential Hierarchy

We finish the study of nonuniform complexity classes from the measure point of view by looking at the measure of P/poly inside the exponential hierarchy. Kannan proved in [Kann] that there is a set in the second level of the exponential hierarchy ($\Sigma_2^{\mathrm{E}} \cap \Pi_2^{\mathrm{E}}$) that is not in P/poly. In this section we prove that P/poly has measure 0 in the third level (Δ_3^{E}), that is, almost every language in Δ_3^{E} is out of P/poly.

We start by reviewing the definition of the exponential hierarchy (weak exponential hierarchy in [Hema], defined in [HartImS]) and defining a resource bounded measure in each level of the hierarchy.

Definition 4.22. Let *the exponential hierarchy* be the class EH defined as follows

$$\mathrm{EH} = \mathrm{NE} \cup \mathrm{NE}(\mathrm{NP}) \cup \mathrm{NE}(\mathrm{NP}(\mathrm{NP})) \cup \ldots \mathrm{NE}(\Sigma_k^{\mathrm{P}}) \cup \ldots$$

By a standard argument, it can be shown that for each $k \in \mathbb{N}$, $\mathrm{NE}(\Sigma_k^{\mathrm{P}}) \subseteq \mathrm{E}(\Sigma_{k+1}^{\mathrm{P}})$ which implies that

$$\mathrm{EH} = \mathrm{E} \cup \mathrm{E}(\mathrm{NP}) \cup \mathrm{E}(\mathrm{NP}(\mathrm{NP})) \cup \ldots \mathrm{E}(\Sigma_k^{\mathrm{P}}) \cup \ldots$$

Definition 4.23. For each $k \in \mathbb{N}$, we define

$$\begin{aligned} \Delta_k^{\mathrm{E}} &= \mathrm{E}(\Sigma_{k-1}^{\mathrm{P}}), \\ \Sigma_k^{\mathrm{E}} &= \mathrm{NE}(\Sigma_{k-1}^{\mathrm{P}}), \text{ and} \\ \Pi_k^{\mathrm{E}} &= \mathrm{coNE}(\Sigma_{k-1}^{\mathrm{P}}). \end{aligned}$$

For each k the following holds

$$\Delta_k^{\mathrm{E}} \subseteq \Sigma_k^{\mathrm{E}} \cap \Pi_k^{\mathrm{E}} \subseteq \Delta_{k+1}^{\mathrm{E}}.$$

For each $k \in \mathbb{N}$, let Δ_k^{P} be the class of functions that can be computed in polynomial time when having access to an oracle in $\Sigma_{k-1}^{\mathrm{P}}$. Notice that Δ_k^{P} is a measure resource-bound as defined in Chapter 1, thus we can define a measure in Δ_k^{E} using Δ_k^{P}-measure because of next Lemma.

Lemma 4.24. $\mathrm{R}(\Delta_k^{\mathrm{P}}) = \Delta_k^{\mathrm{E}}$.

Proof . Remark that the proof of R(p)=E (Lemma 1.17) relativizes. ■

We now show that P/poly has measure 0 in Δ_3^{E}, by using an approximation of the number of polynomial size circuits that agree with a prefix of a language. This approximation is in Δ_3, and is given for any counting function by Stockmeyer in [Stoc85].

We give the formal definition of the class of counting functions, #P.

Definition 4.25. We say that a function $f\colon \{0,1\}^* \to \mathbb{N}$ is in #P when there exists a nondeterministic polynomial time Turing machine M such that, for any $x \in \{0,1\}^*$, $f(x)$ is the number of accepting paths of M on input x.

It is an open problem whether #P is contained in p(PH), that is, if we can count with the help of an oracle in PH. Indeed, Toda's results in [Toda] show that PH $\subseteq$ BPP(#P), which means that counting is at least as hard as the polynomial hierarchy. But even if we do not know how to count in PH, Stockmeyer shows that we can approximate counting in Δ_3^{P}. The next theorem is a particular case of Theorem 3.1 in [Stoc85].

Theorem 4.26. Let $f \in$ #P and let $\epsilon > 0$. There is a function $g \in \Delta_3^{\mathrm{P}}$ such that, for any $x \in \{0,1\}^*$,

$$\left|\frac{g(x)}{f(x)} - 1\right| \leq \epsilon.$$

We finish this chapter showing that P/poly has measure 0 in Δ_3^{E}.

Theorem 4.27. P/poly has measure 0 in Δ_3^{E}.

Proof . Let $k \in \mathbb{N}$ be a power of 2. We define

$$\begin{aligned} X_k = \{A \mid & \text{ for each } n > \log k, \text{ there exists a} \\ & n^{\log n}\text{-size boolean circuit } C_n \text{ that recognizes } A^{=n}\}. \end{aligned}$$

We know that P/poly $\subseteq \bigcup_{k=1}^{\infty} X_k$. We use the Γ-additivity Lemma (Lemma 1.35) to show that $\mu_{\mathrm{p}}(\cup X_k) = 0$.

Let $f\colon \mathbb{N} \times \{0,1\}^* \to \mathbb{N}$ be the following function. For $m = 2^n$, $w \in \{0,1\}^*$

$$\begin{aligned} f_m(w) = \#\{C \mid & C \text{ is an } n\text{-input circuit of size bounded by } n^{\log n} \text{ such that} \\ & \text{for each } s_j \in \{0,1\}^n,\ j < |w|,\ C(s_j) = w[j]\}. \end{aligned}$$

Clearly f is in #P, because each n-input circuit of size bounded by $n^{\log n}$ can be viewed as a path for a nondeterministic polynomial time Turing machine that on input $\langle 2^n, w\rangle$ checks whether there is a circuit with the mentioned size-bound that agrees with w (remember that natural numbers are codified in unary). By the definition of f, for any $m = 2^n$, $w \in \{0,1\}^*$, $f_m(w0) + f_m(w1) = f_m(w)$.

By Theorem 4.26 and fixing $\epsilon = \frac{1}{6}$, we know that there exists $g \in \Delta_3^{\mathrm{P}}$ such that for any $m = 2^n$, $w \in \{0,1\}^*$,

$$\left|\frac{g(m,w)}{f(m,w)} - 1\right| \leq \epsilon.$$

Now we define a 1-MS in a similar way to the proof of Theorem 4.12. In that theorem, we used a function defined as f here, but looking only at circuits with a certain property. That function was in p, thus we could use it in the definition of a martingale in p. Here f is not in Δ_3^{P}, and in order to define a martingale in Δ_3^{P} we use the approximation g.

Since $g_m(w0) + g_m(w1)$ can be bigger than $g_m(w)$, we use

$$\frac{1-\epsilon}{1+\epsilon} \cdot \frac{g_m(wb)}{g_m(w)}$$

instead of $\frac{g_m(wb)}{g_m(w)}$ in the definition of the next 1-MS. Notice that

$$\frac{1-\epsilon}{1+\epsilon} \cdot \frac{g_m(wb)}{g_m(w)} \leq \frac{f_m(wb)}{f_m(w)}$$

and then

$$\frac{1-\epsilon}{1+\epsilon} \cdot \frac{g_m(w0)}{g_m(w)} + \frac{1-\epsilon}{1+\epsilon} \cdot \frac{g_m(w1)}{g_m(w)} \leq 1.$$

We define $d\colon \mathbb{N} \times \{0,1\}^* \to [0,\infty)$ as follows. Let $w \in \{0,1\}^*$, $b \in \{0,1\}$. We have two cases:

If $|w| < k-1$ then $d_k(wb) = 2^{-k}$.

If $|w| \geq k-1$, let m be a power of 2 such that $m-1 \leq |w| < 2m-1$, then

$$d_k(wb) = d_k(w) \cdot \Big(2\frac{1-\epsilon}{1+\epsilon} \cdot \frac{g_m(wb)}{g_m(w)} + 1 - \frac{1-\epsilon}{1+\epsilon} \cdot \frac{g_m(w0)+g(w1)}{g_m(w)}\Big).$$

The second term in the second case of the definition,

$$1 - \frac{1-\epsilon}{1+\epsilon} \cdot \frac{g_m(w0)+g(w1)}{g_m(w)}$$

is nonnegative by the above notice and makes d_k fulfill $d_k(w0) + d_k(w1) = 2\,d_k(w)$, thus d is a 1-MS. Since we can compute g in Δ_3^{P}, d is a 1-MS in Δ_3^{P}.

Now we check that for each $k = 2^i$, $X_k \subseteq \mathrm{S}^{\infty}[d_k]$. For this we use that g approximates f and thus

$$2 \cdot \frac{1-\epsilon}{1+\epsilon} \cdot \frac{g_m(wb)}{g_m(w)} \geq 2 \cdot \frac{1-\epsilon}{1+\epsilon} \cdot \frac{(1-\epsilon) f_m(wb)}{(1+\epsilon) f_m(w)} = 2 \cdot \frac{25}{49} \frac{f_m(wb)}{f_m(w)},$$

since we had fixed $\epsilon = \frac{1}{6}$.

Let $k = 2^i$, $A \in X_k$. From the definition of X_k we know that for any $n > i$, $m = 2^n$, $f_m(\chi_A[0..2^{n+1}-2]) \leq 1$, and from the definition of f, for any $w \in \{0,1\}^*$, $f_m(w) \leq 2^{(2^{\log^2 n})}$. We have then the following inequalities

$$d_k(\chi_A[0..2^{n+1}-2]) \geq d_k(\chi_A[0..2^n-2]) \cdot 2^{(2^n)} \left(\frac{25}{49}\right)^{2^n} \frac{1}{2^{(2^{\log^2 n})}} \geq d_k(\chi_A[0..2^n-2]) 2^{c2^{n-1}},$$

for some constant $c > 0$, independent of k and n. By induction,

$$d_k(\chi_A[0..2^{n+1}-2]) \geq d_k(\lambda) \cdot 2^{c(2^n-k)} = 2^{c(2^n-k)-k}$$

for any $n > i$. This implies that $\limsup_s d_k(\chi_A[0..s]) = \infty$, and $A \in \mathrm{S}[d_k]$. ■

Remark that Kannan showed that there exists a language out of P/poly in the class $\Sigma_2^{\mathrm{E}} \cap \Pi_2^{\mathrm{E}}$, while our techniques can only get the measure result for Δ_3^{E}. A measure in $\Sigma_2^{\mathrm{E}} \cap \Pi_2^{\mathrm{E}}$ can be defined (using the class of single-valued functions that are computable in Σ_2^{P}) but in order to get that P/poly has measure 0 in $\Sigma_2^{\mathrm{E}} \cap \Pi_2^{\mathrm{E}}$ with our techniques, we need to approximate #P using single-valued functions in Σ_2^{P}, which has not been obtained so far.

Chapter 5
If NP is not small

5.1 Introduction

Many of the main open problems in Structural Complexity, such as whether the class NP coincides with one of the classes P or E_2, are instances of a more general problem: the relationship between deterministic and nondeterministic time. There is a strong belief in the area that NP is different from both P and E_2, and that nondeterministic time defines classes whose structure is essentially different from that of deterministic time classes; for instance it is widely believed that NP is not closed under complement.

In this chapter we study NP inside the classes E and E_2 from the measure point of view. A reason to choose E and E_2 as measure environments for NP is that P has measure 0 in both E and E_2, and we want to give some light on whether the same holds for NP.

We study the hypothesis "NP does not have p-measure 0", denoted $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$, and meaning that either NP is not p-measurable or NP has p-measure 1. We are unable to prove or disprove it at this time, because $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ implies $\mathrm{P} \neq \mathrm{NP}$, and $\mu_{\mathrm{p}}(\mathrm{NP}) = 0$ implies $\mathrm{NP} \neq \mathrm{E}_2$. Until such a mathematical resolution is available, the condition $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ is best investigated as a *scientific hypothesis,* to be evaluated in terms of the extent and credibility of its consequences.

In section 5.2 below it is argued that "NP does not have p-measure 0" is a reasonable hypothesis for two reasons: First, its negation would imply the existence of a surprisingly efficient algorithm for betting on all NP languages (the corresponding martingale witnessing that NP has p-measure 0). Second, the hypothesis has a rapidly growing body of credible consequences. We first summarize those that are consequence of the results in previous chapters, dealing with the existence of P-bi-immune languages in NP and with the density of hard languages for NP. We then mention another consequence by Juedes and Lutz [JuedLu94a] that deals with the density of complexity cores of NP-complete languages. Finally we prove two new consequences, namely the class separation $\mathrm{E} \neq \mathrm{NE}$ and (building on recent work of Bellare and Goldwasser [BellGo]) the existence of NP search problems that are not reducible to the corresponding decision problems.

In section 5.3 we use the hypothesis "NP does not have p-measure 0" to separate different types of NP-completeness. The NP-completeness of decision problems has two principal, well-known formulations. These are the polynomial-time Turing completeness ($\leq_{\mathrm{T}}^{\mathrm{p}}$-completeness) introduced by Cook in [Cook] and the polynomial-time many-one completeness ($\leq_{\mathrm{m}}^{\mathrm{p}}$-completeness) introduced by Karp in [Karp] and by Levin in [Levi]. These two completeness notions, sometimes called "Cook completeness" and "Karp-Levin completeness," have been widely conjectured, but not proven (even under the hypothesis that $\mathrm{P} \neq \mathrm{NP}$) to be distinct.

It is clear that $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ implies $A \leq_{\mathrm{T}}^{\mathrm{p}} B$, and hence that every $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete language for NP is $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete for NP. Conversely, all known, natural $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete languages for NP are also $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete. Nevertheless, it is widely conjectured (e.g., [LadnLyS], [LongYo], [Home], [Youn]) that Cook completeness is more general than Karp-Levin completeness:

CvKL Conjecture ("Cook versus Karp-Levin"). There exists a language that is $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete, but not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete, for NP.

The CvKL conjecture immediately implies that $\mathrm{P} \neq \mathrm{NP}$, so it may be very difficult to prove. We mention five items of evidence that the conjecture is reasonable.

1. Selman [Selm79] proved that the widely-believed hypothesis $\mathrm{E} \neq \mathrm{NE}$ implies that the reducibilities $\leq_{\mathrm{T}}^{\mathrm{p}}$ and $\leq_{\mathrm{m}}^{\mathrm{p}}$are distinct in $\mathrm{NP} \cup \text{co-NP}$. That is, if $\mathrm{E} \neq \mathrm{NE}$, then there exist A, $B \in \mathrm{NP} \cup \text{co-NP}$ such that $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ but $A \not\leq_{\mathrm{m}}^{\mathrm{p}} B$. Under the stronger hypothesis $\mathrm{E} \neq \mathrm{NE} \cap \text{co-NE}$, Selman proved that the reducibilities $\leq_{\mathrm{T}}^{\mathrm{p}}$ and $\leq_{\mathrm{m}}^{\mathrm{p}}$are distinct in NP.

2. Watanabe and Tang [WataTa] exhibited reasonable complexity-theoretic hypotheses implying the existence of languages that are $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete, but not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete, for PSPACE.

3. Ko and Moore [KoMo] constructed a language that is $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete, but not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete, for E. Watanabe refined this in [Wata87a], [Wata87b] by separating a spectrum of completeness notions in E.

4. Buhrman, Homer, and Torenvliet [BuhrHoT] constructed languages that are $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete, but not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete, for NE.

5. Longpré and Young [LongYo] showed that, for every polynomial time bound t, there exist languages A and B, both $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete for NP, such that A is $\leq_{\mathrm{T}}^{\mathrm{p}}$-reducible to B in linear time, but A is not $\leq_{\mathrm{m}}^{\mathrm{p}}$-reducible to B in $t(n)$ time.

Item 1 above indicates that the reducibilities $\leq_{\mathrm{T}}^{\mathrm{p}}$ and $\leq_{\mathrm{m}}^{\mathrm{p}}$are expected to differ in NP. Item 2 indicates that the CvKL conjecture is expected to hold with NP replaced by PSPACE. Items 3 and 4 indicate that the CvKL Conjecture definitely holds with NP replaced by E or by NE. Item 5 would imply the CvKL Conjecture, were it not for the dependence of A and B upon the polynomial t. Taken together, these five items suggest that the CvKL Conjecture is reasonable. (See [BuhrTo] for an updated survey of the work on completeness notions.)

The CvKL Conjecture is very ambitious, since it implies that $\mathrm{P} \neq \mathrm{NP}$. The question has thus been raised ([LadnLyS], [Selm79], [Home], [BuhrHoT]) whether the CvKL Conjecture can be derived from some reasonable complexity-theoretic hypothesis, such as $\mathrm{P} \neq \mathrm{NP}$ or the separation of the polynomial-time hierarchy into infinitely many levels. To date, even this more modest objective has not been achieved.

The Main Theorem of this chapter, Theorem 5.10 below, says that the CvKL Conjecture follows from the hypothesis that "NP does not have p-measure 0". We even achieve a stronger result, namely that $\leq_{2-\mathrm{tt}}^{\mathrm{p}}$ and $\leq_{2-\mathrm{T}}^{\mathrm{p}}$-completeness are different for NP, that is, there is a set that is Turing complete using two adaptive queries but, using only two nonadaptive ones, it is not.

In section 5.4, we prove that, if NP is not small, then most truth-table reducibilities are distinct in NP, and in section 5.5 we mention the hypothesis of NP not being small in PSPACE and give some consequences that are corollaries of the results in previous chapters. Section 5.5 also has some open problems. Most of the results in this chapter are contained in [LutzMa94b].

Observe that, for each of the treated questions, the hypothesis "NP does not have p-measure 0" gives the answer that seems most likely, relative to our current knowledge. Taken together, our results suggest that it is a *reasonable scientific hypothesis,* which may have the explanatory power to resolve many questions that have not been resolved by traditional complexity-theoretic hypotheses.

5.2 If NP does not have p-measure 0

We study here the consequences and reasonableness of the hypothesis that NP does not have p-measure 0.

Let us summarize the known implications among various conditions asserting the smallness of NP.

$$\mathrm{P} = \mathrm{NP} \Longrightarrow (\exists c)\mathrm{NP} \subseteq \mathrm{DTIME}(2^{cn}) \Longrightarrow \mu_{\mathrm{p}}(\mathrm{NP}) = 0,$$

$$\mu_{\mathrm{p}}(\mathrm{NP}) = 0 \Longrightarrow \mu_{\mathrm{p}_2}(\mathrm{NP}) = 0 \Longleftrightarrow \mu(\mathrm{NP} \mid \mathrm{E}_2) = 0 \Longrightarrow \mu(\mathrm{NP} \mid \mathrm{E}) = 0.$$

The last implication is a consequence of Corollary 1.26. The second one was proven in Proposition 1.42, the third and fourth follow from Corollary 1.24.

Lutz has conjectured that NP does not have measure 0 in E (denoted $\mu(\mathrm{NP} \mid \mathrm{E}) \neq 0$) and that NP does not have measure 0 in E_2 (denoted $\mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0$). From the previous implications we have

$$\mu(\mathrm{NP} \mid \mathrm{E}) \neq 0 \Longrightarrow \mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0 \Longleftrightarrow \mu_{\mathrm{p}_2}(\mathrm{NP}) \neq 0 \Longrightarrow \mu_{\mathrm{p}}(\mathrm{NP}) \neq 0.$$

This means that $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ is the weakest measure-theoretic hypothesis asserting that NP is not small in exponential time.

By the definition of p-measure, we know that NP has p-measure 0 if and only if there is a single martingale $d \in \mathrm{p}$ that succeeds on every language $A \in \mathrm{NP}$. Since $d \in \mathrm{p}$, when betting on the condition "$x \in A$" d requires only $2^{c|x|}$ time for some fixed constant c. On the other hand, for all $k \in \mathbb{N}$, there exist languages $A \in \mathrm{NP}$ with the property that the apparent search space (space of witnesses) for each input x has $2^{|x|^k}$ elements. Since c is fixed, we have $2^{cn} \ll 2^{n^k}$ for large values of k. Such a martingale d would thus be a very remarkable algorithm! It would bet successfully on all NP languages, using far less than enough time to examine the search spaces of most such languages. It is reasonable to conjecture that no such martingale exists, i.e., that NP does not have p-measure 0.

Kautz and Miltersen have recently proven in [KautMi] that for a randomly chosen A, with probability 1 $\mathrm{NP}(A)$ does not have $\mathrm{p}(A)$-measure 0.

Next we describe some consequences of the hypothesis that NP does not have p-measure 0. The first one concerns P-bi-immunity and is a corollary of the results in Chapter 3. Note that the existence of P-bi-immune sets inside NP has been proven in certain relativizations:

see for instance the oracle constructed by Gasarch and Homer in [GasaHo]. Recall also that E_2 is the smallest deterministic time complexity class known to contain NP.

Theorem 5.1. If NP does not have p-measure 0 then NP contains a P-bi-immune set. If NP does not have measure 0 in E_2 then NP contains an E-bi-immune set.

Proof . From Theorem 3.3 we know that the class of P-bi-immune sets has p-measure 1, so if NP does not have p-measure 0 then NP is not included in the class of non-P-bi-immune languages, and we have the first part. For the second part the argument is the same this time using Theorem 3.6. ■

The next known consequence of $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ is proven by Juedes and Lutz in [JuedLu94a], and involves exponential complexity cores of NP-complete languages, defined as follows:

Definition 5.2. An infinite set $K \subseteq \{0,1\}^*$ is an *exponential complexity core* for a language A if there is a real number $\epsilon > 0$ such that for every machine M that accepts A there are at most finitely many $x \in K$ such that the time of machine M on input x is smaller than $2^{|x|^\epsilon}$.

(Intuitively, an exponential complexity core for a language L is a set of 'very infeasible' inputs for every algorithm that correctly recognizes L.)

Theorem 5.3. [JuedLu94a]. If NP does not have p-measure 0, then every $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete language A for NP has a dense exponential complexity core.

Thus, for example, if NP is not small, then there is a dense set K of Boolean formulas in conjunctive normal form such that every machine that is consistent with SAT performs exponentially badly (either by running for more than $2^{|x|^\epsilon}$ steps or by giving no output) on all but finitely many inputs $x \in K$. (The weaker hypothesis $\mathrm{P} \neq \mathrm{NP}$ was already known [OrpoSc] to imply the weaker conclusion that every $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete language for NP has a nonsparse polynomial complexity core.)

The third consequence of $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ to be mentioned here concerns the density of hard languages for NP. Let us consider the usual polynomial-time reducibilities ranging from $\leq^{\mathrm{p}}_{\mathrm{m}}$ to $\leq^{\mathrm{p}}_{\mathrm{T}}$. If $\leq^{\mathrm{p}}_{r}$ is any of these reducibilities, *all known $\leq^{\mathrm{p}}_{r}$-hard languages for NP are dense.* Efforts to explain this observation (and similar observations for other classes and reducibilities) have yielded many results. (See [HemaOgW] for a thorough survey.) Berman and Hartmanis [BermHa] conjectured that no sparse language is $\leq^{\mathrm{p}}_{\mathrm{m}}$-hard for NP, unless $\mathrm{P} = \mathrm{NP}$. This conjecture was subsequently proven correct:

Theorem 5.4. [Maha]. If $\mathrm{P} \neq \mathrm{NP}$, then no sparse language is $\leq^{\mathrm{p}}_{\mathrm{m}}$-hard for NP. That is,

$$\mathrm{P} \neq \mathrm{NP} \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}_{\mathrm{m}}(\mathrm{SPARSE}).$$

Theorem 5.4 was extended much later to truth-table reducibility with a bounded number of queries:

Theorem 5.5. (Ogihara and Watanabe [OgihWa]). If $\mathrm{P} \neq \mathrm{NP}$, then no sparse language is $\leq^{\mathrm{p}}_{\mathrm{btt}}$-hard for NP. That is,

$$\mathrm{P} \neq \mathrm{NP} \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}_{\mathrm{btt}}(\mathrm{SPARSE}).$$

One is thus led to ask whether there is a reasonable hypothesis θ such that we can prove results of the form

$$\theta \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}_r(\mathrm{DENSE}^c), \tag{5.1}$$

for various choices of the reducibility $\leq_r^{\mathrm{p}}$. (Such a result is much stronger than the corresponding result

$$\theta \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}_r(\mathrm{SPARSE}),$$

because there is an enormous gap between polynomial and 2^{n^ϵ} growth rates.)

Ogihara and Watanabe's proof of Theorem 5.5 does not appear to allow significant relaxation of either the query bound or the sparseness criterion. In fact, it appears to be beyond current understanding to prove results of the form (5.1) if θ is "P $\neq$ NP." Karp and Lipton [KarpLi] have proven that

$$\Sigma_2^{\mathrm{P}} \neq \Pi_2^{\mathrm{P}} \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}(\mathrm{SPARSE}).$$

That is, the stronger hypothesis $\Sigma_2^{\mathrm{P}} \neq \Pi_2^{\mathrm{P}}$ gives a stronger conclusion than those of Theorems 5.4 and 5.5. However, Karp and Lipton's proof does not appear to allow relaxation of the sparseness criterion, and results of the form (5.1) do not appear to be achievable at this time if θ is taken to be "$\Sigma_2^{\mathrm{P}} \neq \Pi_2^{\mathrm{P}}$."

As a corollary of the main result in Chapter 4 the following holds

Theorem 5.6. If NP does not have p-measure 0 then, for every real number $\alpha < 1$, every $\leq_{n^\alpha-\mathrm{tt}}^{\mathrm{P}}$-hard language for NP is dense.

Proof . The result follows trivially from Theorem 4.17, stating that $\mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$ has p-measure 0. ■

This last conclusion of the hypothesis NP does not have p-measure 0, which is credible and consistent with all observations to date, is not known to follow from P $\neq$ NP or other traditional complexity-theoretic hypotheses.

Note that the hypothesis and conclusion of Theorem 5.6 are both stronger than their counterparts in Ogihara and Watanabe's result that

$$\mathrm{P} \neq \mathrm{NP} \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}_{\mathrm{btt}}(\mathrm{SPARSE}).$$

Note also that our proof of Theorem 5.6 (based on Theorem 4.17) actually shows that

$$\mathrm{NP} \cap \mathrm{WS}_{3,\frac{1}{2}} \neq \emptyset \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c).$$

We conclude this section by noting some new consequences of the hypothesis that $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$. The following lemma does not depend on this hypothesis; it involves the exponential complexity classes E and NE, and also the doubly exponential complexity classes, $\mathrm{EE} = \bigcup_{c=0}^{\infty} \mathrm{DTIME}(2^{2^{n+c}})$ and $\mathrm{NEE} = \bigcup_{c=0}^{\infty} \mathrm{NTIME}(2^{2^{n+c}})$.

Lemma 5.7.

1. If NP contains a P-bi-immune language, then $\mathrm{E} \neq \mathrm{NE}$ and $\mathrm{EE} \neq \mathrm{NEE}$.
2. If $\mathrm{NP} \cap \text{co-NP}$ contains a P-bi-immune language, then $\mathrm{E} \neq \mathrm{NE} \cap \text{co-NE}$ and $\mathrm{EE} \neq \mathrm{NEE} \cap \text{co-NEE}$.

Proof . Let $T = \{0^{2^n} | n \in \mathbb{N}\}$. For each $A \subseteq \{0,1\}^*$, let

$$\sigma(A) = \{s_n | 0^{2^n} \in A\},$$

It is routine to show that, for all $A \subseteq \{0,1\}^*$,

$$\begin{aligned} &\sigma(A) \in \mathrm{EE} \text{ iff } A \cap T \in \mathrm{P}, \\ &\sigma(A) \in \mathrm{NEE} \text{ iff } A \cap T \in \mathrm{NP}, \text{ and} \\ &\sigma(A) \in \text{co-NEE} \text{ iff } A \cap T \in \text{co-NP}. \end{aligned}$$

1. Let $A \in \mathrm{NP}$ be P-bi-immune. Then $A \cap T \in \mathrm{NP}$, so $\sigma(A) \in \mathrm{NEE}$. Since A^c is P-immune, $A \cap T$ is infinite. Since A is P-immune, it follows that $A \cap T \notin \mathrm{P}$, whence $\sigma(A) \notin \mathrm{EE}$. Thus $\sigma(A) \in \mathrm{NEE} - \mathrm{EE}$, so $\mathrm{EE} \neq \mathrm{NEE}$. Note also that $A \cap T$ is a tally language in $\mathrm{NP} - \mathrm{P}$. The existence of such a language is known [Book74] to be equivalent to $\mathrm{E} \neq \mathrm{NE}$.

The proof of 2 is similar. ∎

Theorem 5.8.

1. If NP does not have p-measure 0, then $\mathrm{E} \neq \mathrm{NE}$ and $\mathrm{EE} \neq \mathrm{NEE}$.
2. If $\mathrm{NP} \cap \text{co-NP}$ does not have p-measure 0, then $\mathrm{E} \neq \mathrm{NE} \cap \text{co-NE}$ and $\mathrm{EE} \neq \mathrm{NEE} \cap \text{co-NEE}$.

Proof . This follows immediately from Theorem 5.1 and Lemma 5.7. ∎

Corollary 5.9. If NP does not have p-measure 0, then there is an NP search problem that does not reduce to the corresponding decision problem.

Proof . Bellare and Goldwasser [BellGo] have shown that, if $\mathrm{EE} \neq \mathrm{NEE}$, then there is an NP search problem that does not reduce to the corresponding decision problem. The present corollary follows immediately from this and Theorem 5.8. ∎

5.3 Separating completeness notions in NP

In this section we prove our main consequence of $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$, that is:

Theorem 5.10. If NP does not have p-measure 0, then there is a language C that is $\leq^{\mathrm{P}}_{2-\mathrm{T}}$-complete, but not $\leq^{\mathrm{P}}_{2-\mathrm{tt}}$-complete, for NP.

This theorem implies that if $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ then the CvKL conjecture holds, that is

Corollary 5.11. If $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ then there is a language that is $\leq^{\mathrm{P}}_{\mathrm{T}}$-complete, but not $\leq^{\mathrm{P}}_{\mathrm{m}}$-complete, for NP.

Our proof of Theorem 5.10 uses the following definitions and lemma.

Definition 5.12. The *tagged union* of languages $A_0, \cdots, A_{k-1} \subseteq \{0,1\}^*$ is the language

$$A_0 \oplus \cdots \oplus A_{k-1} = \left\{ x10^i \,|0 \leq i < k \text{ and } x \in A_i \right\}.$$

Definition 5.13. For each language $A \subseteq \{0,1\}^*$ let $A_{(0)}$ and $A_{(1)}$ be the following languages

$$A_{(0)} = \left\{ x \mid x \in A \text{ and } x = y0 \right\};$$

$$A_{(1)} = \left\{ x \mid x \in A \text{ and } x = y1 \right\}.$$

Lemma 5.14. For any language $S \in \mathrm{E}$, the set

$$X = \left\{ A \subseteq \{0,1\}^* \,|A_{(0)} \leq^{\mathrm{p}}_{2-\mathrm{tt}} A_{(1)} \oplus (A_{(1)} \cap S) \oplus (A_{(1)} \cup S) \right\}$$

has p-measure 0.

Before proving Lemma 5.14, we use it to prove the Main Theorem.

Proof of Theorem 5.10. Assume that NP does not have p-measure 0. Let

$$X = \left\{ A \,|A_{(0)} \leq^{\mathrm{p}}_{2-\mathrm{tt}} A_{(1)} \oplus (A_{(1)} \cap \mathrm{SAT}) \oplus (A_{(1)} \cup \mathrm{SAT}) \right\}.$$

By Lemma 5.14, X has p-measure 0, so there exists a language $A \in \mathrm{NP} - X$. Fix such a language A and let

$$C = A_{(1)} \oplus (A_{(1)} \cap \mathrm{SAT}) \oplus (A_{(1)} \cup \mathrm{SAT}).$$

Since $A \in \mathrm{NP}$, we have $A_{(0)}, A_{(1)} \in \mathrm{NP}$. Since $A_{(1)}, \mathrm{SAT} \in \mathrm{NP}$ and NP is closed under $\cap$, $\cup$, and $\oplus$, we have $C \in \mathrm{NP}$. Also, the algorithm

```
BEGIN
        INPUT  $x$;
        IF  $x1 \in C$
                THEN IF  $x10 \in C$
                        THEN  accept
                        ELSE  reject
                ELSE IF  $x100 \in C$
                        THEN  accept
                        ELSE  reject
END
```

clearly decides SAT using just two (adaptive) queries to C, so $\mathrm{SAT} \leq^{\mathrm{p}}_{2-\mathrm{T}} C$. Thus C is $\leq^{\mathrm{p}}_{2-\mathrm{T}}$-complete for NP. On the other hand, $A \notin X$, so $A_{(0)} \not\leq^{\mathrm{p}}_{2-\mathrm{tt}} C$. Since $A_{(0)} \in \mathrm{NP}$, it follows that C is not $\leq^{\mathrm{p}}_{2-\mathrm{tt}}$-complete for NP. ■

Next we prove Lemma 5.14.

Proof of Lemma 5.14.

Let S and X be as in the hypothesis. Our objective is to prove that $\mu_{\rm p}(X) = 0$. Let $c > 0$ be such that $S \in \mathrm{DTIME}(2^{cn})$.

For each language A, we denote as A_S the language

$$A_S = A_{(1)} \oplus (A_{(1)} \cap S) \oplus (A_{(1)} \cup S).$$

Using this notation,

$$X = \left\{A \subseteq \{0,1\}^* \;\middle|\; A_{(0)} \leq^{\rm p}_{2-\rm tt} A_S\right\}.$$

Let $\{M_i \mid i \in \mathbb{N}\}$ be a feasible enumeration of all oracle Turing machines performing $\leq^{\mathrm{DTIME}(2^{(c+1)n})}_{2-\rm tt}$-reductions. (The reason to allow time $2^{(c+1)n}$ to the reductions is to be able to simulate the computation of $S(y)$ for strings y shorter than the input, as we will see below.) For each $i \in \mathbb{N}$, let (f^i, g^i) be the 2-tt reduction performed by M_i (we use the notation for truth-table reducibilities introduced in Chapter 1).

For each $i \in \mathbb{N}$, for each $x \in \{0,1\}^*$, $l \in \{1,2\}$, if $f^i_l(x) = w10^a$ with $a \in \{0,1,2\}$, we write $q^i_l(x)$ for w, and $a^i_l(x)$ for a. Notice that if M_i performs a reduction to A_S, then $q^i_l(x)$ is the actual query to either $A_{(1)}$, $A_{(1)} \cap S$ or $A_{(1)} \cup S$ and $a^i_l(x)$ tells us which part of A_S is being queried.

Let $A \in X$, and M be a polynomial time machine witnessing that $A_{(0)} \leq^{\rm p}_{2-\rm tt} A_S$. Then there exists an $i \in \mathbb{N}$ such that $A_{(0)} = L(M_i, A_S)$ and the following conditions hold

(i) for every $x \in \{0,1\}^*$ and $l \in \{1,2\}$, $f^i_l(x)$ has the form $w10^a$, for $a \in \{0,1,2\}$;

(ii) for every $x \in \{0,1\}^*$, $q^i_1(x) \leq q^i_2(x)$ in lexicographical order, and if $q^i_1(x) = q^i_2(x)$ then $a^i_1(x) < a^i_2(x)$;

(iii) for every $x \in \{0,1\}^*$ and $l \in \{1,2\}$, $q^i_l(x) \neq x$, $|q^i_l(x)| < 2^{|x|}$;

(iv) for every $x \in \{0,1\}^*$ and $l \in \{1,2\}$, either $|q^i_l(x)| > |x|$ or $a^i_l(x) = 0$.

Conditions (i) to (iii) can be induced because we are dealing with a polynomial-time reduction from $A_{(0)}$ to A_S. Condition (iv) holds because we can simulate $M(x)$ computing $S(q)$ for all queries q with $|q| \leq |x|$ in time $2^{c|x|}$, and transform these queries into queries to $A_{(1)}$.

For each $i \in \mathbb{N}$ we define the set X_i as follows:

If i fulfills conditions (i) to (iv) then

$$X_i = \{A \subseteq \{0,1\}^* \mid A_{(0)} = L(M_i, A_S)\}.$$

Otherwise, $X_i = \emptyset$.

By the above observation, $X \subseteq \bigcup\limits_{i=0}^{\infty} X_i$.

In order to prove that X has p-measure 0, by Lemma 1.35, it is enough to define a p-computable 1-MS d, such that $X_i \subseteq \mathrm{S}^\infty[d_i]$, for all $i \in \mathbb{N}$.

The clue in the definition of d is the next claim.

Claim 1. Let $i \in \mathbb{N}$, let $A \in X_i$. For every $x \in \{0,1\}^*$ one of the following holds

(a) There exists a language B such that $A^{<x} = B^{<x}$ and for every R, $L(M_i, B_R)(x) = L(M_i, B_S)(x)$.

(b) $q_1^i(x) > x$ and $A_{(0)}(x) = L(M_i, \emptyset)(x) \Leftrightarrow A(q_1^i(x)) = S(q_1^i(x))$.

Assuming Claim 1, we can define $d\colon \mathbb{N} \times \{0,1\}^* \to [0,\infty)$ as follows. We use the sequence of strings $\{x_n \mid n \in \mathbb{N}\}$, where $x_1 = \lambda$, $x_n = 0^{2^{|x_{n-1}|}}$ for $n > 0$.

Let $i \in \mathbb{N}$, $w \in \{0,1\}^*$. Let n be such that $x_n \leq s_{|w|} < x_{n+1}$.

(1) If there exists a language B such that $B^{<x_n} \sqsubseteq w$ and for every R, $L(M_i, B_R)(x_n) = L(M_i, B_S)(x_n)$, then fix the first such B and let Z_n be the class

$$Z_n = \Big\{ C \;\Big|\; \text{If } \big(C(q_1^i(x_n)) = B(q_1^i(x_n)) \text{ and } C(q_2^i(x_n)) = B(q_2^i(x_n))\big) \\ \text{then } C(x_n) = L(M_i, B_\emptyset)(x_n) \Big\}.$$

If $\Pr_C[C \in Z_n | C \in \mathbf{C}_w] \neq 0$ then

$$d_i(wb) = d_i(w) \cdot \frac{\Pr_C[C \in Z_n | C \in \mathbf{C}_{wb}]}{\Pr_C[C \in Z_n | C \in \mathbf{C}_w]},$$

If $\Pr_C[C \in Z_n | C \in \mathbf{C}_w] = 0$ then $d_i(wb) = d_i(w)$.

(2) Otherwise, if $s_{|w|} = q_1^i(x_n)$ then

$$d_i(wb) = \begin{cases} 2 \cdot d_i(w) & \text{if } L(M_i, \emptyset)(x_n) = w[2^{|x_n|} - 1] \Leftrightarrow S(q_1^i(x)) = b, \\ 0 & \text{if } L(M_i, \emptyset)(x_n) = w[2^{|x_n|} - 1] \Leftrightarrow S(q_1^i(x)) \neq b, \end{cases}$$

If $s_{|w|} \neq q_1^i(x_n)$ then $d_i(wb) = d_i(w)$.

For each $i \in \mathbb{N}$, d_i is a martingale by the definition of conditional probability in case (1) and by definition in case (2). In order to compute $d(i,w)$ we need to compute $\{f_1^i(x), f_2^i(x)\}$ for several $x < s_{|w|}$ for part (1), and $S(s_{|w|})$ for part (2), all of which can be done in time polynomial in $|w| + i$. Thus d is a p-computable 1-MS.

We now show that $X_i \subseteq \mathrm{S}^\infty[d_i]$ for all $i \in \mathbb{N}$. Fix $i \in \mathbb{N}$ and $A \in X_i$. Let $w_1 = \lambda$, $r_1 = 1$. For each $n > 1$, let $w_n = \chi_A[0..2^{|x_n|} - 2]$ and $r_n = d_i(w_n)$. (That is, w_n is the initial segment of the characteristic sequence χ_A of A up to but not including the bit that decides whether $x_n \in A$.) Since the choice of (1) or (2) in the definition of d_i depends only on n, either for every $w \sqsubseteq A$ with $x_n \leq s_{|w|} < x_{n+1}$ we are in case (1) or for every $w \sqsubseteq A$ with $x_n \leq s_{|w|} < x_{n+1}$ we are in case (2).

If we are in case (1) then, if $A(q_1^i(x_n)) = B(q_1^i(x_n))$ and $A(q_2^i(x_n)) = B(q_2^i(x_n))$ we know that $L(M_i, A_S)(x_n) = L(M_i, B_S)(x_n) = L(M_i, B_\emptyset)(x_n)$. Since $A \in X_i$ we have $A_{(0)}(x_n) = L(M_i, A_S)(x_n)$, thus $A \in Z_n$ and for each $w \sqsubseteq A$, $\Pr_C[C \in Z_n | C \in \mathbf{C}_{w_n}] \neq 0$. Therefore,

$$r_{n+1} = r_n \cdot \frac{\Pr_C[C \in Z_n | C \in \mathbf{C}_{w_{n+1}}]}{\Pr_C[C \in Z_n | C \in \mathbf{C}_{w_n}]}.$$

Moreover, x_n, $q_1^i(x_n)$ and $q_2^i(x_n)$ are decided by w_{n+1}, so $\Pr_C[C \in Z_n | C \in \mathbf{C}_{w_{n+1}}] = 1$, that is,

$$r_{n+1} = \frac{r_n}{\Pr_C[C \in Z_n | C \in \mathbf{C}_{w_n}]}.$$

Finally,

$$\Pr_C[C \in Z_n | C \in \mathbf{C}_{w_n}] < 1$$

because a set C such that $C(x_n) \neq B(x_n)$, $C(q_1^i(x_n)) = B(q_1^i(x_n))$ and $C(q_2^i(x_n)) = B(q_2^i(x_n))$ is in $\mathbf{C}_{w_n} \setminus Z_n$. Since there are at most 3 bits that influence whether $C \in Z_n$, then $\Pr_C[C \in Z_n | C \in \mathbf{C}_{w_n}] < 1$ implies that $\Pr_C[C \in Z_n | C \in \mathbf{C}_{w_n}] \leq 1 - 2^{-3}$. We thus have

$$r_{n+1} \geq \frac{8}{7} \cdot r_n.$$

If we are in case (2) then by Claim 1, $q_1^i(x_n) > x_n$ and $A(x_n) = L(M_i, \emptyset)(x_n)$ if and only if $A(q_1^i(x_n)) = S(q_1^i(x_n))$. Thus $r_{n+1} = 2 \cdot r_n$.

This implies that $\lim_{n\to\infty} d_i(\chi_A[0..|w_n| - 1]) = \infty$, thus $\limsup_m d_i(\chi_A[0..m]) = \infty$ and $A \in \mathrm{S}^\infty[d_i]$. This completes the proof that $X_i \subseteq \mathrm{S}^\infty[d_i]$ for all $i \in \mathbb{N}$. By Lemma 1.35, we have finished the proof of Lemma 5.14 via the proof of Claim 1.

Proof of Claim 1. Let $i \in \mathbb{N}$ be such that $X_i \neq \emptyset$. Let $x \in \{0,1\}^*$. One of the following seven cases must happen. Cases I) to VI) correspond to (a), case VII) corresponds to (b).

I) $q_1^i(x) = q_2^i(x)$ and $|q_1^i(x)| \leq |x|$.

For every B and every R, $L(M_i, B_R)(x) = L(M_i, B_S)(x)$, because all queries of machine M_i on input x involving R are longer than $|x|$ (condition (iv)).

II) $q_1^i(x) \neq q_2^i(x)$.

For $l \in \{1,2\}$, if $q_l^i(x) > x$ let

$$B(q_l^i(x)) = \begin{cases} 0 & \text{if } a_l^i(x) = 1 \\ 1 & \text{otherwise.} \end{cases}$$

Then for every R, $B_R(f_l^i(x)) = B_S(f_l^i(x))$, and $L(M_i, B_R)(x) = L(M_i, B_S)(x)$ (remember that by condition (iv) all queries of machine M_i on input x involving R are longer than $|x|$).

III) $q_1^i(x) = q_2^i(x)$, $|q_1^i(x)| > |x|$ and $a_1^i(x) = 0$ or $a_2^i(x) \neq 2$.

Let

$$B(q_1^i(x)) = \begin{cases} 0 & \text{if } a_2^i(x) = 1 \\ 1 & \text{otherwise.} \end{cases}$$

Then for every R, for each $l \in \{1,2\}$, $B_R(f_l^i(x)) = B_S(f_l^i(x))$, and $L(M_i, B_R)(x) = L(M_i, B_S)(x)$.

The rest of the cases happen when $q_1^i(x) = q_2^i(x)$, $a_1^i(x) = 1$, and $a_2^i(x) = 2$. Notice that the answers $(1,0)$ are not possible.

IV) $q_1^i(x) = q_2^i(x)$, $a_1^i(x) = 1$, $a_2^i(x) = 2$ and $g^i(x)(0,0) = g^i(x)(1,1) = g^i(x)(0,1)$.

In this case, for every B and for every R, $L(M_i, B_R)(x) = L(M_i, B_S)(x)$.

V) $q_1^i(x) = q_2^i(x)$, $|q_1^i(x)| > |x|$, $a_1^i(x) = 1$, $a_2^i(x) = 2$ and $g^i(x)(0,0) \neq g^i(x)(1,1) = g^i(x)(0,1)$.

Let $B(q_1^i(x)) = 1$. Then for every R, $B_R(q_2^i(x)) = 1$, and $L(M_i, B_R)(x) = L(M_i, B_S)(x)$.

VI) $q_1^i(x) = q_2^i(x)$, $|q_1^i(x)| > |x|$, $a_1^i(x) = 1$, $a_2^i(x) = 2$ and $g^i(x)(1,1) \neq g^i(x)(0,0) = g^i(x)(0,1)$.

Let $B(q_1^i(x)) = 0$. Then for every R, $B_R(q_1^i(x)) = 0$, and $L(M_i, B_R)(x) = L(M_i, B_S)(x)$.

VII) $q_1^i(x) = q_2^i(x)$, $|q_1^i(x)| > |x|$, $a_1^i(x) = 1$, $a_2^i(x) = 2$ and $g^i(x)(0,0) = g^i(x)(1,1) \neq g^i(x)(0,1)$.

Here we are in case (b). If $A_{(0)}(x) = g^i(x)(0,0)$ then $S(q_1^i(x)) = A(q_1^i(x))$, because it is the only way of having answers $(0,0)$ or $(1,1)$. In the same way if $A_{(0)}(x) = g^i(x)(0,1)$ then $S(q_1^i(x)) \neq A(q_1^i(x))$. ■

5.4 Separating reducibilities in NP

In this section, assuming that NP is not small, we establish the distinctness of many polynomial-time reducibilities in NP.

Our first such result involves known consequences of E $\neq$ NE.

Theorem 5.15. Assume that NP does not have p-measure 0.

There exist $A, B \in \mathrm{NP} \cup \text{co-NP}$ such that $A \leq^{\mathrm{p}}_{\mathrm{T}} B$, but $A \not\leq^{\mathrm{p}}_{\mathrm{pos-T}} B$.

There exist $A, B \in \mathrm{NP} \cup \text{co-NP}$ such that $A \leq^{\mathrm{p}}_{\mathrm{tt}} B$, but $A \not\leq^{\mathrm{p}}_{\mathrm{pos-tt}} B$.

Proof. Selman [Selm82] has shown that these conclusions follow from E $\neq$ NE, so the present theorem follows immediately from Theorem 5.8. ■

Similarly, we have the following.

Theorem 5.16. Assume that $\mathrm{NP} \cap \text{co-NP}$ does not have p-measure 0.

There exist $A, B \in \mathrm{NP}$ such that $A \leq^{\mathrm{p}}_{\mathrm{T}} B$ but $A \not\leq^{\mathrm{p}}_{\mathrm{pos-T}} B$.

There exist $A, B \in \mathrm{NP}$ such that $A \leq^{\mathrm{p}}_{\mathrm{tt}} B$ but $A \not\leq^{\mathrm{p}}_{\mathrm{pos-tt}} B$.

Proof. Selman [Selm82] has shown that these conclusions follow from $\mathrm{E} \neq \mathrm{NE} \cap \text{co-NE}$, so the present theorem follows immediately from Theorem 5.8. ■

The rest of our results concern the separation of various polynomial-time truth-table reducibilities in NP, according to the number of queries. Theorem 5.17 separates $\leq^{\mathrm{p}}_{(k+1)-\mathrm{tt}}$ reducibility from $\leq^{\mathrm{p}}_{k-\mathrm{tt}}$, for any constant k, while Theorem 5.19 separates $\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}}$ reducibility from $\leq^{\mathrm{p}}_{r(n)-\mathrm{tt}}$, for $q(n) \in o(\sqrt{r(n)})$ and $r(n) \in O(n)$.

Theorem 5.17. If NP does not have p-measure 0, then for all $k \in \mathbb{N}$ there exist $A, B \in \mathrm{NP}$ such that $A \leq^{\mathrm{p}}_{(k+1)-\mathrm{tt}} B$ but $A \not\leq^{\mathrm{p}}_{k-\mathrm{tt}} B$.

The proof of Theorem 5.17 uses the following notation and lemma For $x \in \{0,1\}^*$ and $k \in \mathbb{N}$, let

$$Q_k(x) = \{x10^i | 0 \leq i < k\}.$$

For all $B \subseteq \{0,1\}^*$ and $k \in \mathbb{N}$, then, define the *k-fold disjunction of* B to be the language

$$\vee^{(k)} B = \{x \in \{0,1\}^* | Q_k(x) \cap B \neq \emptyset\}.$$

Lemma 5.18. For all $k \in \mathbb{N}$, the set

$$X_k = \{B \subseteq \{0,1\}^* | \vee^{(k+1)} B \leq^{\mathrm{p}}_{k-\mathrm{tt}} B\}$$

has p-measure 0.

Proof of Theorem 5.17. Assume that NP does not have p-measure 0 and let $k \in \mathbb{N}$. Then Lemma 5.18 tells us that there exists $B \in$ NP such that $\vee^{(k+1)} B \not\leq^{\mathrm{p}}_{k-\mathrm{tt}} B$. Fix such a language B and let $A = \vee^{(k+1)} B$. Then $A \in$ NP (because $A \leq^{\mathrm{p}}_{\mathrm{pos-T}} B$ and NP is closed under $\leq^{\mathrm{p}}_{\mathrm{pos-T}}$-reducibility [Selm82]), $A \leq^{\mathrm{p}}_{(k+1)-\mathrm{tt}} B$ (trivially), and $A \not\leq^{\mathrm{p}}_{k-\mathrm{tt}} B$ (by our choice of B). ■

Proof of Lemma 5.18. Fix $k \in \mathbb{N}$ and let X_k be as in the statement of the lemma. Let $\{M_i \mid i \in \mathbb{N}\}$ be a feasible enumeration of all Turing Machines performing $\leq^{\mathrm{p}}_{k-\mathrm{tt}}$-reductions. We can assume that for every $i \in \mathbb{N}$ and every $x \in \{0,1\}^*$, all queries of machine M_i on input x are shorter than $2^{|x|}$. For each $i \in \mathbb{N}$, let (f^i, g^i) be the k-tt reduction performed by M_i

For $i \in \mathbb{N}$, we define the set

$$Y_i = \{B \subseteq \{0,1\}^* \mid \vee^{(k+1)} B = L(M_i, B)\}.$$

Let

$$Y = \bigcup_{i=0}^{\infty} Y_i.$$

It is clear that $X_k \subseteq Y$, so it suffices to prove that $\mu_{\mathrm{p}}(Y) = 0$.

In order to prove that Y has p-measure 0, by Lemma 1.35, it is enough to define a p-computable 1-MS d, such that $Y_i \subseteq \mathrm{S}^\infty[d_i]$, for all $i \in \mathbb{N}$.

The function d: $\mathbb{N} \times \{0,1\}^* \to [0,\infty)$ is defined as follows:

We use again $\{x_n \mid n \in \mathbb{N}\}$ a sequence of very separated strings defined as $x_1 = \lambda$, $x_n = 0^{2^{|x_{n-1}|}}$ for $n > 1$.

Let $i \in \mathbb{N}$, $d_i(\lambda) = 1$. Let $w \in \{0,1\}^*$, and $b \in \{0,1\}$. Let n be such that $x_n \leq s_{|w|} < x_{n+1}$.

(i) If $\Pr_C\left[L(M_i, C)(x_n) = \vee^{(k+1)} C(x_n) | C \in \mathbf{C}_w\right] \neq 0$ then

$$d_i(wb) = d_i(w) \cdot \frac{\Pr_C\left[L(M_i, C)(x_n) = \vee^{(k+1)} C(x_n) | C \in \mathbf{C}_{wb}\right]}{\Pr_C\left[L(M_i, C)(x_n) = \vee^{(k+1)} C(x_n) | C \in \mathbf{C}_w\right]}.$$

(ii) Otherwise, $d_i(wb) = d_i(w)$.

For each $i \in \mathbb{N}$, d_i is a martingale; as in previous proofs this is by the definition of conditional probability. In order to compute $d(i, w)$ we need to compute $Q_{k+1}(x) \cup \{f_1^i(x), \ldots, f_k^i(x)\}$ for several $x < s_{|w|}$, which can be done in time polynomial in $|w| + i$. Thus d is a p-computable 1-MS.

We now show that $Y_i \subseteq \mathrm{S}^\infty[d_i]$ for all $i \in \mathbb{N}$. Fix $i \in \mathbb{N}$ and $A \in Y_i$. Let $w_1 = \lambda$, $r_1 = 1$. For each $n > 1$, let $w_n = \chi_A[0..2^{|x_n|} - 2]$ and $r_n = d_i(w_n)$. (That is, w_n is a prefix of the characteristic sequence χ_A of A up to but not including the bit corresponding to x_n.) Since $A \in Y_i$, for every $w \sqsubseteq A$, $\Pr_C\big[L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n) \bigm| C \in \mathbf{C}_w\big] \neq 0$, and for each $w \sqsubseteq A$, we use case (i) in the definition of $d_i(w)$. Thus for $n > 1$

$$r_{n+1} = r_n \cdot \frac{\Pr_C\big[L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n) | C \in \mathbf{C}_{w_{n+1}}\big]}{\Pr_C\big[L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n) | C \in \mathbf{C}_{w_n}\big]}.$$

For all $n > 1$ all the queries of M_i on input x_n and all the strings in $Q_{k+1}(x_n)$ are decided by w_{n+1}, so

$$\Pr_C\big[L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n) | C \in \mathbf{C}_{w_{n+1}}\big] = 1,$$

that is,

$$r_{n+1} = \frac{r_n}{\Pr_C\big[L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n) | C \in \mathbf{C}_{w_n}\big]}.$$

Finally, for all $n > 1$, the fact that

$$\Pr_C\big[L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n) | C \in \mathbf{C}_{w_n}\big] < 1$$

follows from the next claim

Claim 2. Let $A \in X_i$. For every $x \in \{0,1\}^*$ there exists a language B such that $A^{<x} = B^{<x}$ and $L(M_i, B)(x) \neq \vee^{(k+1)}B(x)$.

The claim holds because no string in $Q_{k+1}(x)$ is fixed by $A^{<x}$ and M_i can make at most k queries on input x. ■

Since there are at most $2k + 1$ bits that influence whether $L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n)$ for each C, then

$$\Pr_C\big[L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n) | C \in \mathbf{C}_{w_n}\big] < 1$$

implies that

$$\Pr_C\big[L(M_i, C)(x_n) = \vee^{(k+1)}C(x_n) | C \in \mathbf{C}_{w_n}\big] \leq 1 - 2^{-2k-1}. \tag{5.2}$$

We thus have

$$r_{n+1} \geq \frac{r_n}{1 - 2^{-2k-1}}$$

for all $n > 1$. This implies that $\lim_{n\to\infty} d_i(\chi_A[0..|w_n| - 1]) = \infty$, thus

$$\limsup_m d_i(\chi_A[0..m]) = \infty$$

and $A \in \mathrm{S}^\infty[d_i]$. This completes the proof that $Y_i \subseteq \mathrm{S}^\infty[d_i]$ for all $i \in \mathbb{N}$, and by Lemma 1.35, the proof of Lemma 5.18. ■

For non constant query-bounds, we have the following result.

Theorem 5.19. If NP does not have p-measure 0 and $q, r\colon \mathbb{N} \to \mathbb{N}$ are polynomial-time computable query-counting functions satisfying the conditions $q(n) \in o(\sqrt{r(n)})$ and $r(n) \in O(n)$, then there exist $A, B \in \mathrm{NP}$ such that $A \leq^{\mathrm{p}}_{r(n)-\mathrm{tt}} B$ but $A \not\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}} B$.

To prove this theorem, we use a very similar technique to that of Theorem 5.17, this time substituting the disjunctive operator by a majority operator. The following notation and lemma are used

For all $B \subseteq \{0,1\}^*$ and $r\colon \mathbb{N} \to \mathbb{N}$, we define the *r-fold majority of B* to be the language

$$\mathrm{maj}^{(r)}B = \left\{ x \in \{0,1\}^* \;\middle|\; \left|Q_{r(|x|)}(x) \cap B\right| \geq \left\lceil \frac{r(|x|)}{2} \right\rceil \right\}.$$

Lemma 5.20. If $q, r\colon \mathbb{N} \to \mathbb{N}$ are polynomial-time computable functions satisfying the conditions $q(n) \in o(\sqrt{r(n)})$ and $r(n) \in O(n)$, then the set

$$X = \{B \subseteq \{0,1\}^* | \mathrm{maj}^{(r)}B \leq^{\mathrm{p}}_{q(n)-\mathrm{tt}} B\}$$

has p-measure 0.

Proof of Theorem 5.19. This is similar to the proof of Theorem 5.17, using Lemma 5.20 and $\mathrm{maj}^{(r)}B$ in place of Lemma 5.18 and $\vee^{(k+1)}B$. ■

Proof of Lemma 5.20. The proof of this lemma is similar to that of Lemma 5.18, but we now have unbounded query-counting functions where we previously had constants.

Let $\{M_i \mid i \in \mathbb{N}\}$ be a feasible enumeration of all Turing Machines performing $\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}}$-reductions.

Following the steps and notation in the proof Lemma 5.18, we need a constant upper bound for

$$\Pr_C\big[L(M_i, C)(x_n) = \mathrm{maj}^{(r)}C(x_n) | C \in \mathbf{C}_{w_n}\big],$$

as in (5.2). In this case the existence of such a bound is a consequence of the fact that

$$\Pr_C\big[L(M_i, C)(x_n) = \mathrm{maj}^{(r)}C(x_n) | C \in \mathbf{C}_{w_n}\big]$$

has a limit $\leq \frac{1}{2}$ as n goes to infinity. So there exists a n_0 such that

$$\Pr_C\big[L(M_i, C)(x_n) = \mathrm{maj}^{(r)}C(x_n) | C \in \mathbf{C}_{w_n}\big] \leq \frac{3}{4}$$

for every $n \geq n_0$.

The rest of the proof follows the same arguments as in Lemma 5.18. ■

The query bounds of Theorems 5.17 and 5.19 can be relaxed if we make the stronger assumption that $\mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0$. Theorem 5.17 can be extended to logarithmically bounded query-counting functions, while in Theorem 5.19 we can remove the requirement that $r(n) \in O(n)$.

Theorem 5.21. If $\mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0$ and q is a polynomial-time computable query-counting function such that $q(n) \in O(\log n)$, then there exist $A, B \in \mathrm{NP}$ such that $A {\leq}^{\mathrm{p}}_{q(n)+1-\mathrm{tt}} B$ but $A \not\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}} B$.

Theorem 5.22. If $\mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0$ and $q, r \colon \mathbb{N} \to \mathbb{N}$ are polynomial-time computable query-counting functions satisfying $q(n) \in o(\sqrt{r(n)})$, then there exist $A, B \in \mathrm{NP}$ such that $A {\leq}^{\mathrm{p}}_{r(n)-\mathrm{tt}} B$ but $A \not\leq^{\mathrm{p}}_{q(n)-\mathrm{tt}} B$.

The proofs of Theorems 5.21 and 5.22 are similar to those of Theorems 5.17 and 5.19, respectively. Details are omitted.

5.5 Further results and open problems

We look here at the consequences of the hypothesis 'NP does not have measure 0 in PSPACE' we know so far. Notice that if $\mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0$ then $\mu(\mathrm{NP} \mid \mathrm{PSPACE}) \neq 0$

Theorem 5.23. If NP does not have measure 0 in PSPACE then

(i) NP contains a DLOG-bi-immune language.

(ii) NP contains a language that is not LINSPACE-oq-self reducible.

Proof . The proof follows from Theorems 3.9 and 2.11. ∎

We have seen that for each of the treated questions, the hypothesis "NP does not have p-measure 0" gives the answer that seems most likely, relative to our current knowledge. Further investigation of this hypothesis and its power to resolve other questions is clearly indicated. Additionally, it allows us to combine all relativizable proofs of measure 1 in E results for specific properties with the result of Kautz and Miltersen [KautMi] that the set of all A such that $\mu(\mathrm{NP}(A) \mid \mathrm{E}(A)) \neq 0$ has Lebesgue measure 1. For instance, it follows that with probability 1, a random oracle has a $\mathrm{P}(A)$-bi-immune set in $\mathrm{NP}(A)$, which generalizes the construction of an oracle for which NP contains a P-bi-immune set in [GasaHo] (in fact, for that oracle $\mathrm{NP} = \mathrm{E}_2$).

Regarding the density of hard languages for NP, there are several open questions involving special reducibilities. We mention just one example. Very recently, Arvind, Köbler, and Mundhenk [ArviKöM] have proven that

$$\mathrm{P} \neq \mathrm{NP} \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}_{\mathrm{btt}}(\mathrm{P}_{\mathrm{ctt}}(\mathrm{SPARSE})),$$

where $\mathrm{P}_{\mathrm{ctt}}$ refers to polynomial-time *conjunctive* reducibility. (This strengthens Theorem 5.5.) Does the class $\mathrm{P}_{\mathrm{btt}}(\mathrm{P}_{\mathrm{ctt}}(\mathrm{DENSE}^c))$ have measure 0 in E?

As noted in section 5.2, all known $\leq_{\mathrm{T}}^{\mathrm{P}}$-hard languages for NP are dense, i.e., our experience suggests that NP $\not\subseteq$ P(DENSE^c). This suggests two open questions. Karp and Lipton [KarpLi] have shown that

$$\Sigma_2^{\mathrm{P}} \neq \Pi_2^{\mathrm{P}} \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P(SPARSE)}.$$

Theorem 5.6 shows that

$$\mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0 \Longrightarrow \mathrm{NP} \not\subseteq \mathrm{P}_{n^\alpha-\mathrm{tt}}(\mathrm{DENSE}^c)$$

for $\alpha < 1$. The first question, posed by Selman, is whether the strong hypothesis $\mu(\Sigma_2^{\mathrm{P}} \backslash \Pi_2^{\mathrm{P}} \mid \mathrm{E}_2) \neq 0$ can be used to combine these ideas to get a conclusion that NP $\not\subseteq$ P(DENSE^c). The second, more fundamental, question is suggested by the first. A well-known downward separation principle [Stoc77] says that, if the polynomial time hierarchy separates at some level, then it separates at all lower levels. Thus, for example, $\Sigma_2^{\mathrm{P}} \neq \Pi_2^{\mathrm{P}}$ implies that P $\neq$ NP. Is there a "downward measure separation principle," stating that $\mu(\Sigma_{k+1}^{\mathrm{P}} \backslash \Pi_{k+1}^{\mathrm{P}} \mid \mathrm{E}_2) \neq 0 \Longrightarrow \mu(\Sigma_k^{\mathrm{P}} \backslash \Pi_k^{\mathrm{P}} \mid \mathrm{E}_2) \neq 0$? In particular, does $\mu(\Sigma_2^{\mathrm{P}} \backslash \Pi_2^{\mathrm{P}} \mid \mathrm{E}_2) \neq 0$ imply that $\mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0$?

The next immediate open problem involves the further separation of completeness notions in NP. We have shown that the hypothesis $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ separates $\leq_{2-\mathrm{T}}^{\mathrm{P}}$-completeness from $\leq_{2-\mathrm{tt}}^{\mathrm{P}}$-completeness in NP. However, there is a large spectrum of completeness notions between $\leq_{\mathrm{T}}^{\mathrm{P}}$ and $\leq_{\mathrm{m}}^{\mathrm{P}}$. Watanabe ([Wata87a], [Wata87b]) and Buhrman, Homer, and Torenvliet [BuhrHoT] have shown that nearly all these completeness notions are distinct in E and in NE, respectively. In light of the results of sections 5.3 and 5.4 above, it is reasonable to conjecture that the hypothesis "NP does not have p-measure 0" yields a similarly detailed separation of completeness notions in NP. Investigation of this conjecture may shed new light on NP-completeness phenomena.

We finish by looking at the Berman-Hartmanis isomorphism conjecture formulated in 1977, namely that all NP $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete sets are polynomial time isomorphic [BermHa]. A lot of work has been done around this conjecture (for a survey, see [KurtMaR]). Most researchers now believe that the isomorphism conjecture as stated by Berman and Hartmanis is false. It would be very interesting to obtain results of the form "If NP does not have p-measure 0 then the isomorphism conjecture is false for $\leq_r^{\mathrm{P}}$-complete sets", for different reducibilities $\leq_r^{\mathrm{P}}$.

Chapter 6

Cones

6.1 Introduction

Given a reducibility R, we can picture the lattice defined by the preorder relation R on the class of all languages. Fix a language A and look at the two classes formed respectively by languages that are R-reducible to A and languages to which A is R-reducible to. These two classes can be intuitively viewed as the two parts of the cone starting in vertex A. The R-upper cone of A, denoted $R^{-1}(A)$, is the class of all languages to which A is R-reducible, and the R-lower cone of A, denoted $R(A)$, is the class of languages that are R-reducible to A. In this chapter we want to study the size of the upper and lower cones of a language A as a way of having information on the usefulness of A as oracle and on the amount of oracles A reduces to. In this line, Juedes and Lutz have studied in [JuedLu94a] the measure of $\leq^{\mathrm{p}}_{\mathrm{m}}$-cones in E.

The size of the lower cone of a language A gives us information on the usefulness of A as oracle. Using this concept, a language A is R-hard for a class $\mathcal{C}$ iff the R-lower cone of A contains $\mathcal{C}$. Lutz proposed in [Lutz90] to weaken this condition and consider languages A such that $R(A)$ contains a non negligible part of $\mathcal{C}$, languages which were later called weakly-hard by Juedes and Lutz in [JuedLu94a]. Thus a language A is R-weakly-hard for a class $\mathcal{C}$ when $R(A)$ does not have measure 0 in $\mathcal{C}$. Intuitively, A is weakly-hard when a non-negligible subclass of $\mathcal{C}$ is reducible to A. Clearly every hard set A is weakly-hard, since its lower cone contains the whole $\mathcal{C}$. Lutz posed in [Lutz90] the question of whether the opposite holds, that is, whether there exist R-weakly-hard languages that are not R-hard for a class $\mathcal{C}$. The interest of this question comes from the use of hardness as a proof of intractability, for instance Stockmeyer and Chandra show in [StocCh] that certain two-person combinatorial games are intractable by proving that they are polynomial time many-one hard for E. The existence of languages that are R-weakly-hard and not R-hard for a certain reducibility R would imply the existence of another level in the classification of languages by criteria of R-intractability, below the level of R-hardness.

In [Lutz94a] Lutz solves affirmatively this question for the class E and the reducibility $\leq^{\mathrm{p}}_{\mathrm{m}}$, showing that there exists a language in E that is $\leq^{\mathrm{p}}_{\mathrm{m}}$-weakly-hard and not $\leq^{\mathrm{p}}_{\mathrm{m}}$-hard for E. The proof of this result is based on Lutz's new diagonalization technique called martingale diagonalization, that allows us to diagonalize against all martingales in Γ, for a particular Γ, while at the same time we pursue another agenda, using for this second objective classical diagonalization techniques. In section 6.2 below we see how martingale diagonalization can be used to show that there exists a language H such that H is weakly-useful (as defined in [JuedLaL]), which means that for some time bound t, H is $\leq^{t}_{\mathrm{T}}$-weakly-hard for the class REC, but such that H is not strongly useful, that is, for every time bound f H is not $\leq_T^f$-hard for REC. As a consequence of an auxiliary result, we show that the class of $\leq_{tt}^p$-complete languages for ESPACE has measure 0 in ESPACE.

Regarding the size of the upper cone of a language A, it gives us an idea of the amount of oracles A reduces to. Thus if the upper cone is large, A reduces to almost every oracle; this means intuitively that access to the oracle is merely used as a source of random bits. In this way, for each reducibility R the class ALMOST-R is defined as the class of languages A such that the R-upper cone of A has Lebesgue probability 1 (studied for instance in [Book94] and [BookLuW]). The "ALMOST-R" formalism provides characterizations of certain complexity classes that are well-studied in structural complexity theory. For example, P=ALMOST-$\leq_m^p$[Ambo], P=ALMOST-$\leq_{btt}^p$ [TangBo], BPP=ALMOST-$\leq_T^p$ ([Ambo], [BennGi]), BPP=ALMOST-$\leq_{tt}^p$ [TangBo], AM=ALMOST-$\leq_T^{NP}$ ([Cai], [NisaWi]), PH=ALMOST-$\leq_T^{PH}$ ([Cai], [NisaWi]) and IP=ALMOST-IP [Breu]. Book, Lutz, and Wagner studied these classes in [BookLuW] and [Book94], characterizing them in terms of algorithmically random languages in the sense of Martin-Löf [Mart]. The notion of Martin-Löf algorithmically random language is the strongest definition of random language that is considered to represent randomness of infinite sequences. Book, Lutz and Wagner ([Book94], [BookLuW]) have characterized each class of the form ALMOST-R as the class of recursive languages that can be R-reduced to Martin-Löf algorithmically random languages. This characterizations lead to observations about the relationships between complexity classes such as: P = NP if and only if some algorithmically random language is $\leq_{btt}^p$-hard for NP, and PH = PSPACE if and only if some algorithmically random language is $\leq_T^{PH}$-hard for PSPACE.

In section 6.3 we give new characterizations of the classes ALMOST-R. For each natural n, we consider a subclass of Martin-Löf random languages, denoted n-random languages, and show that a language A in Δ_n^0 (the nth level of the Kleene arithmetical hierarchy) is in ALMOST-R if and only if A is R-reducible to an n-random language. This gives us an idea of, for instance, how difficult can $\leq_T^p$-oracles for BPP be. We also see that n-random oracles are useless for the class $\Delta_n^0 - \mathrm{REC}$. Considering the Kleene arithmetical hierarchy as a whole, we show that a language A in it is in ALMOST-R if and only if A is R-reducible to an ω-random language. The concept of "ω-randomness" is, in a sense, the "limit" of the n-random sets, and has been introduced and studied in [Kaut].

In section 6.4 we discuss the interest of defining a bidimensional resource-bounded measure to study problems that are better formulated in terms of pair of languages. We see that for any well-behaved bidimensional measure, the class of pairs of languages that are $\leq_m^p$-incomparable, that is, (A, B) such that $A \not\leq_m^p B$ and $B \not\leq_m^p A$ is non measurable in E × E.

The results in section 6.2 are joint work with S. Fenner and J.H. Lutz, included in [FennLuM]. The results in section 6.3 appear in [BookMa]. The study of bidimensional measure is still incomplete and therefore unpublished. This chapter contains ongoing research in the area of measure of cones: we include at the end of each section some open problems as guidelines of possible future investigation.

6.2 Weakly-useful languages

We start by giving the definition of lower cone.

Definition 6.1. Given a reducibility R and a language A, we define the *R-lower cone of A*, denoted $R(A)$, as the following class

$$R(A) = \{B \mid B \leq_R A\}.$$

Notice that for each class $\mathcal{C}$, a language A is R-hard for $\mathcal{C}$ iff $\mathcal{C}$ is contained in the R-lower cone of A.

Based on the widely believed assumptions that P is different from NP, researchers have viewed the fact that a problem is $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard for NP as a proof that the problem is intractable. An absolute proof of intractability is obtained when we show that a language is $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard for E. This intractability comes from some properties of $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard problems for E, for instance they are not in P and they have a dense complexity core. In general proving that a language A is R-hard for E, for a polynomial-time reduction R, gives us an idea of intractability of the language, because if A is R-hard for E then it cannot be in P.

Lutz proposes in [Lutz90] to weaken hardness as the notion of intractability and consider intractable those languages A such that $R(A)$ does not have measure 0. Formally,

Definition 6.2. Given a reducibility R and a class $\mathcal{C}$ (with a non trivial measure), we say that a language A is *R-weakly-hard for $\mathcal{C}$* if and only if $R(A)$ does not have measure 0 in $\mathcal{C}$. We say that a language A is *R-weakly-complete for $\mathcal{C}$* if and only if A is R-weakly-hard for $\mathcal{C}$ and $A \in \mathcal{C}$.

Notice that every hard language is weakly-hard because $\mathcal{C}$ does not have measure 0 in $\mathcal{C}$. Remark that, since P has measure 0 in E, for each polynomial-time reducibility R, R-weakly-hard problems for E are not in P, thus are intractable in a sense.

The next step would be to show that indeed this concept is more general than that of hardness, by showing that for each class and reducibility, there is a weakly-hard language that is not hard. In [Lutz94a], Lutz gets this for the class E with reducibility $\leq_{\mathrm{m}}^{\mathrm{p}}$, showing that there exists a language that is $\leq_{\mathrm{m}}^{\mathrm{p}}$-weakly-complete but not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete for E. Let us briefly review his proof.

Lutz constructs a language $H \in \mathrm{E}_2$ with two properties

1) H is not $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard for E.
2) $\mathrm{P}_{\mathrm{m}}(H)$ does not have measure 0 in E.

By a padding argument, this shows that there is a $\leq_{\mathrm{m}}^{\mathrm{p}}$-weakly-complete language for E that is not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete.

For property 1), the concept of incompressibility is used. A language A is $\leq_{\mathrm{m}}^{\mathrm{p}}$-incompressible when for every $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction f from A to $f(A)$, the following holds

$$\left|\{(x,y) \mid x,y \in \{0,1\}^*,\ x \neq y,\ \text{and } f(x) = f(y)\}\right| < \infty,$$

that is, f is almost everywhere one-one. (This concept was introduced with the name of strong P-bi-immunity in [BalcSc].)

Juedes and Lutz show in [JuedLu94a] that every $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard language for E is not $\leq_{\mathrm{m}}^{\mathrm{p}}$-incompressible (From [Berm] and [BalcSc] this was already known for $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete languages.) In order to obtain a language H that is not $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard, the language H is constructed with the property of being $\leq_{\mathrm{m}}^{\mathrm{p}}$-incompressible. This can be done by classical diagonalization.

For property 2), we define for each $i \in \mathbb{N}$ and each language A let the ith strand of A be the following language

$$A_i = \{x \mid \langle i, x\rangle \in A\}.$$

Notice that for every A, $\{A_i \mid i \in \mathbb{N}\} \subseteq \mathrm{P}_{\mathrm{m}}(A)$.

Lutz constructs H such that for every $i \in \mathbb{N}$, $H_i \in \mathrm{E}$ and such that $\{H_i \mid i \in \mathbb{N}\}$ does not have measure 0 in E, thus obtaining property 2).

Here we look at the class REC of all recursive languages and study hardness and weakly-hardness for this class. In REC we only restrict Turing reductions to those that always stop, that is, $\leq_{\mathrm{T}}^{t}$-reductions for any recursive function t. In this context, the concepts of strongly-useful and weakly-useful appear in the place of hardness and weakly hardness. A language is strongly useful if it is $\leq_{\mathrm{T}}^{t}$-hard for some t, and it is weakly useful if it is $\leq_{\mathrm{T}}^{t}$-weakly-hard for some t. Let us start by defining these concepts, introduced in [JuedLaL].

Definition 6.3. A language A is *strongly-useful* when there exists a recursive function t such that A is $\leq_{\mathrm{T}}^{t}$-hard for REC.

Definition 6.4. A language A is *weakly-useful* when there exists a recursive function t such that A is $\leq_{\mathrm{T}}^{t}$-weakly-hard for REC.

Notice that every strongly-useful language is weakly-useful, and that no weakly-useful language is recursive.

With a technique similar to the martingale diagonalization we just described, we construct next a weakly-useful language that is not strongly-useful.

The results in this section relate to time-bounded Turing reductions, but we will lose no generality if we restrict our attention to tt-reductions. This is because if t is a recursive function and M is an oracle Turing machine running in time $t(n)$ for all oracles, then there is a well defined truth-table oracle Turing machine M' running in time exponential in $t(n)$ such that $L(M, C) = L(M', C)$ for any oracle C. Moreover, M' can be found effectively from M. (This is a Theorem by Nerode that is included in [Roge], page 143.)

The main result in this section is the following

Theorem 6.5. There exists a language H which is weakly-useful but not strongly-useful.

To prove this theorem, we will define H one strand at a time to satisfy the following conditions:

1. For every recursive time bound t, there is a recursive set A such that $A \notin \mathrm{DTIME}^{H}(t)$.
2. Each strand H_k is recursive.
3. If d is any martingale in rec, then there is some k such that d fails on H_k.

These three conditions suffice for our purposes. Condition 1 ensures that H is not strongly-useful. By Condition 2, the set $J = \{H_0, H_1, H_2, \ldots\} \subseteq \mathrm{REC}$, and by Condition 3, no recursive martingale can succeed on all its elements. Thus $\mu(J \mid REC) \neq 0$, which makes H weakly-useful, since $J \subseteq \mathrm{DTIME}^H(\text{linear})$.

Highly incompressible languages

In order to have Condition 1, we start by constructing, for each recursive time bound t, a recursive set A that is 'highly incompressible' for $\leq_{\mathrm{tt}}^t$-reductions, that is, for almost every B, $A \notin \mathrm{DTIME}^B(t)$. Then it will be very easy to construct a language H such that $A \notin \mathrm{DTIME}^H(t)$.

Proposition 6.6. If $\{f_i \bigm| i \in \mathbb{N}\}$ is a uniform family of recursive tt-reductions, then there is a recursive set A such that

$$\mu_{\mathrm{rec}}(\{B \bigm| \exists i \text{ with } A = f_i(B)\}) = 0.$$

We obtain the following theorem as an immediate corollary.

Theorem 6.7. For any recursive time bound t, there is a recursive set A such that $\mu_{\mathrm{rec}}(\{B \bigm| A \in \mathrm{DTIME}^B(t)\}) = 0$.

Proof of Proposition 6.6.

We first consider a single tt-reduction f and define a recursive set A_f with $\mu_{\mathrm{rec}}(\{B \mid A_f = f(B)\}) = 0$ as follows:

$$A_f(\lambda) = \begin{cases} 0 & \text{if } \Pr_C[f(C,\lambda) = 0] \leq \frac{1}{2}, \\ 1 & \text{otherwise,} \end{cases}$$

and for each $n \in \mathbb{N}, n > 0$,

$$A_f(s_n) = \begin{cases} 0 & \text{if } \Pr_C\big[f(C,s_n) = 0 \mid f(C)[0..n-1] = A_f[0..n-1]\big] \leq \frac{1}{2}, \\ 1 & \text{otherwise.} \end{cases}$$

Fact 6.8. For all $n \in \mathbb{N}$, $\Pr_C\big[f(C)[0..n-1] = A_f[0..n-1]\big] \leq 2^{-n}$.

We now describe a rec-computable martingale d^f that succeeds on any set B such that $A_f = f(B)$. We split d^f up into infinitely many martingales

$$d^f = \sum_{\ell=1}^{\infty} d_\ell,$$

where each martingale d_ℓ bets a finite number of times. Fixing ℓ, let

$$d_\ell(w) = 2^{|w|-\ell} \cdot \Pr_C\big[w \sqsubseteq C \mid f(C)[0..\ell-1] = A_f[0..\ell-1]\big]$$

for all $w \in \{0,1\}^*$, if $\Pr_C\big[f(C)[0..\ell-1] = A_f[0..\ell-1]\big] > 0$. Otherwise, let $d_\ell(w) = 2^{-\ell}$ for all $w \in \{0,1\}^*$. It can be easily checked that d_ℓ is a martingale. Consider any $w_0 \sqsubseteq B$ that is long enough to be defined on all queries made by f for all inputs less than s_ℓ, and let $\mathbf{E}_\ell$ be the event that $f(C)[0..\ell-1] = A_f[0..\ell-1]$. We have $\Pr_C\big[\mathbf{E}_\ell \mid w_0 \sqsubseteq C\big] = 1$, and thus

$$\begin{aligned} d_\ell(w_0) &= 2^{|w_0|-\ell} \cdot \Pr_C[w_0 \sqsubseteq C \mid \mathbf{E}_\ell] \\ &= 2^{|w_0|-\ell} \cdot \frac{\Pr_C[\mathbf{E}_\ell \mid w_0 \sqsubseteq C] \cdot \Pr_C[w_0 \sqsubseteq C]}{\Pr_C[\mathbf{E}_\ell]} \\ &= \frac{2^{-\ell}}{\Pr_C[\mathbf{E}_\ell]} \\ &\geq 1, \end{aligned}$$

by Fact 6.8. Applying the above inequality to the martingale d^f, we get that for any ℓ, there is a $w \sqsubseteq B$ such that $d(w) \geq \ell$. Thus d^f succeeds on B.

Returning to the $\{f_i \bigm| i \in \mathbb{N}\}$ in the hypothesis of Proposition 6.6, we let

$$A = \{\langle i, x\rangle \mid x \in A_{\tilde{f}_i}\},$$

where $\tilde{f}_i(C, x)$ simulates $f_i(C, \langle i, x\rangle)$ for all i, x and C. (Therefore, $A_{\tilde{f}_i}$ is the ith strand of A.) We call A the *highly incompressible set with respect to* $\{f_i \bigm| i \in \mathbb{N}\}$. A is clearly recursive. To prove the proposition, we define a recursive martingale d that succeeds on all B such that $A = f_i(B)$ for some i. Let

$$d = \sum_{i=0}^{\infty} d^{\tilde{f}_i} \cdot 2^{-i}.$$

The martingale d is recursive, since each $d^{\tilde{f}_i}$ can be computed uniformly. For some B, if $A = f_i(B)$ for some i, then $A_{\tilde{f}_i} = \tilde{f}_i(B)$, and so $d^{\tilde{f}_i}$ succeeds on B by the previous discussion. Hence d succeeds on B.

Remark that in Proposition 6.6 if we take $\{f_i \bigm| i \in \mathbb{N}\}$ to be the family of all $\leq^t_{\mathrm{tt}}$-reductions, for t a recursive time bound, then in the proof of the proposition, A is in DSPACE($2^n \cdot t(n)$) and d is DSPACEF$\big(m \cdot t(\log m)\big)$-approximable. Thus we have the following corollary.

Corollary 6.9. There exists a language $A \in$ ESPACE such that

$$\mu(\{B \bigm| A \leq^{\mathrm{p}}_{\mathrm{tt}} B\} \mid \mathrm{ESPACE}) = 0.$$

This implies the next result for the class of $\leq^{\mathrm{p}}_{\mathrm{tt}}$-complete languages for ESPACE.

Corollary 6.10. The class of $\leq^{\mathrm{p}}_{\mathrm{tt}}$-complete languages for ESPACE has measure 0 in ESPACE.

Proof of Theorem 6.5

We start by giving some notation on partial characteristic functions.

A *partial characteristic function* is a function with domain a subset of $\{0,1\}^*$ and with range $\{0,1\}$. We will identify a binary string w with the characteristic function whose domain is $\{s_0, \ldots, s_{|w|-1}\}$. If σ and τ are partial characteristic functions, we let $\mathrm{dom}(\sigma)$ denote the domain of σ, and say that σ and τ are *compatible* if they agree on all elements in $\mathrm{dom}(\sigma) \cap \mathrm{dom}(\tau)$. We say that σ *is extended by* τ ($\sigma \sqsubseteq \tau$) if σ and τ are compatible and $\mathrm{dom}(\sigma) \subseteq \mathrm{dom}(\tau)$ (if in addition $\sigma \neq \tau$, we write $\sigma \sqsubsetneq \tau$). If σ and τ are compatible, we let $\sigma \cup \tau$ be their smallest common extension.

If σ is a partial characteristic function, $i \in \mathbb{N}$ and $x \in \{0,1\}^*$, then $\sigma[i, < x]$ denotes the unique partial characteristic function τ such that for all y,

$$\tau(y) = \begin{cases} \sigma(\langle i, y \rangle) & \text{if } y < x \text{ and } \sigma(\langle i, y \rangle) \text{ is defined,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

That is, $\sigma[i, < s_n]$ results from "excising" the first n bits of σ from the ith column. Inversely, if w is a binary string, then $\{i\} \times w$ denotes the unique partial characteristic function τ such that $\tau(\langle i, s_n \rangle) = w[n]$ for all $n < |w|$, and is undefined on all other arguments. That is, $\{i\} \times w$ is w "transported" over to the ith column. Of particular importance will be the finite characteristic function defined for an arbitrary language C, $k \in \mathbb{N}$ and $y \in \{0,1\}^*$ as

$$\xi^C(k, y) = \bigcup_{k' < k} \{k'\} \times C[k', < y].$$

Fix an arbitrary enumeration $\{t_k \mid k \in \mathbb{N}\}$ of all recursive time bounds, and an enumeration $\{\tilde{d}_k \mid k \in \mathbb{N}\}$ of all recursive martingales. These enumerations need not be uniform in any sense, since at present we are not trying to control the complexity of H. We will define (in order) a number of different objects for each k:

- a uniform family $\{f_j^k \mid j \in \mathbb{N}\}$ of tt-reductions corresponding to t_k.
- a recursive A^k such that $A^k \notin \mathrm{DTIME}^H(t_k)$ (A^k will be the highly incompressible set with respect to $\{f_j^k \mid j \in \mathbb{N}\}$),
- a partial characteristic function α_k of finite domain, compatible with all the previous strands of H (ultimately, $\alpha_k \sqsubseteq H$ for all k),
- martingales $d_{k;q}^{i,j}$ (uniformly recursive on j and q) for all $i, j, q \in \mathbb{N}$ with $i \leq k$, which, taken together, witness that each A^i is highly incompressible, and
- the strand H_k itself, which is designed to make the martingale

$$d'_k = \tilde{d}_k + \sum_{i=0}^{k} \sum_{j=0}^{\infty} \sum_{q=0}^{\infty} d_{k;q}^{i,j} \cdot 2^{-q-j}$$

  fail on H_k, thus $\tilde{d}_k$ fails on H_k and Condition 3 above is satisfied. H_k will also participate in a fixed finite number of diagonalizations against tt-reductions from the A^i to H for $i \leq k$.

Fix $k \in \mathbb{N}$, and assume that all the above objects have been defined for all $k' < k$ (define $\alpha_{-1} = \lambda$). Also assume that for each $k' < k$ we have at our disposal programs to compute (uniformly over j and q) $f_j^{k'}$, $A^{k'}$, $H_{k'}$, and $d^{i,j}_{k';q}$ for all $i \leq k'$. Let $\{M_{j,k} \mid j \in \mathbb{N}\}$ be a recursive enumeration of all oracle Turing machines running in time t_k, and for all j let $M'_{j,k}$ be the same as $M_{j,k}$ except that when $M_{j,k}$ makes a query of the form $\langle m, y\rangle$ for $m < k$, $M'_{j,k}$ instead simulates the answer by computing $H_m(y)$ directly. We let f_j^k be the tt-reduction corresponding to $M'_{j,k}$. Note that on any input, f_j^k only makes queries of the form $\langle m, y\rangle$ for $m \geq k$.

We define A^k to be the highly incompressible set constructed in the proof of Proposition 6.6 using $\{f_j^k \mid j \in \mathbb{N}\}$ as the family of tt-reductions. The analogue of Fact 6.8 above says that

Fact 6.11. For all $j, k, n \in \mathbb{N}$, $\Pr_C\left[f_j^k(C)[j, < s_n] = A^k[j, < s_n]\right] \leq 2^{-n}$.

Let $H_{<k}$ denote the partial characteristic function that agrees with H on all $\langle m, y\rangle$ with $m < k$, and is undefined otherwise. Given α_{k-1}, which is compatible with $H_{<k}$, we define α_k as follows: let $\langle i, j\rangle = k$. If there is a set C with $H_{<k} \cup \alpha_{k-1} \sqsubseteq C$ such that $A^i \neq f_j^i(C)$, then we *diagonalize* against f_j^i by letting α_k be the least finite characteristic function extending α_{k-1} that preserves such a miscomputation, i.e., for some C and x such that $A^i(x) \neq f_j^i(C, x)$, α_k will agree with C on all queries made by f_j^i on input x. If no such C exists, let $\alpha_k = \alpha_{k-1}$.

Now fix any i and j with $i \leq k$. We would like to define a martingale to perform the task that d did back in the proof of Proposition 6.6 for the set A^i. We cannot do this directly, because any given tt-reduction f_j^i from A^i to H might make queries on many different columns at once, and our martingales can only act on one column at a time. Instead, for any $q \in \mathbb{N}$ large enough, the martingales $d^{i,j}_{k';q}$ for all $k' \geq i$ will act together to "succeed as a group" on all sets to which A^i reduces via f_j^i.

The martingale $d^{i,j}_{k;q}$ will be split up into infinitely many martingales

$$d^{i,j}_{k;q} = \sum_{\ell=1}^{\infty} d^{i,j}_{k;q;\ell},$$

similar to the proof of Proposition 6.6. Fix i and j. For any $m \in \{0,1\}^*$, let y_m be least such that $v < y_m$ for all queries $\langle u, v\rangle$ made by f_j^i on inputs $\langle j, x\rangle$ for all $x < s_m$. For any language C, let $\mathbf{E}^C(m)$ be the event that $f_j^i(C)[j, < m] = A^i[j, < m]$, i.e., that $f_j^i(C)$ and A^i agree on $\{\langle i, y\rangle \mid y < s_m\}$. For all $w \in \{0,1\}^*$, we define

$$d^{i,j}_{k;q;\ell}(w) = 2^{|w|-\ell} \cdot \Pr_C\left[\xi^H(k, y_{q\ell}) \cup (\{k\} \times w) \sqsubseteq C \mid \xi^H(k, y_{q\ell}) \sqsubseteq C \,\&\, \mathbf{E}^C(q\ell)\right]$$

if $\Pr_C[\xi^H(k, y_{q\ell}) \sqsubseteq C \ \&\ \mathbf{E}^C(q\ell)] > 0$. Otherwise, for all w define $d^{i,j}_{k;q;\ell}(w) = 2^{-\ell}$.

Remark 6.12. The definition of $d^{i,j}_{k;q;\ell}(w)$ above remains unchanged if we replace $y_{q\ell}$ with any $y \geq y_{q\ell}$. This is because $\mathbf{E}^C(q\ell)$ depends on C only for those queries made by f_j^i on inputs $\langle j, \lambda\rangle, \ldots, \langle j, s_{q\ell-1}\rangle$. None of these are of the form $\langle u, y\rangle$ for $y \geq y_{q\ell}$.

We now define H_k. For any $n \in \mathbb{N}$, we assume that $H_k[0..n-1]$ has already been defined, and we set $w = H_k[0..n-1]$. Let

$$H_k[n] = \begin{cases} \alpha_k(\langle k, s_n\rangle) & \text{if } \alpha_k(\langle k, s_n\rangle) \text{ is defined,} \\ 0 & \text{if } \alpha_k(\langle k, s_n\rangle) \text{ is undefined and } d'_k(w0) \leq d'_k(w1), \\ 1 & \text{if } \alpha_k(\langle k, s_n\rangle) \text{ is undefined and } d'_k(w0) > d'_k(w1). \end{cases}$$

Remark 6.13. Actually, we cannot do this exactly as stated. A rec-computable martingale such as d'_k cannot in general be computed exactly, but is only approximated. What we are really comparing are not $d'_k(w0)$ and $d'_k(w1)$, but rather their nth approximations, which *are* computable. Since these approximations are guaranteed to be within 2^{-n} of the actual values, and our sole aim is to make d'_k fail on H_k, it suffices for our purposes to consider only the approximations when doing the comparisons above. The same trick is used in [Lutz94a].

H_k is evidently recursive (given the last remark), and for cofinitely many n, $H_k[n]$ is chosen so that $d'_k(H_k[0..n]) \leq d'_k(H_k[0..n-1]) + 2^{-n}$, the 2^{-n} owing to the error in the approximation of d'_k. Thus d'_k fails on H_k, from which we obtain

Fact 6.14. The martingales $\tilde{d}_k$ and $d^{i,j}_{k;q}$ for all and $i \leq k$, j, and q all fail on H_k.

Thus Conditions 2 and 3 are satisfied. Each H_k also preserves the diagonalization commitments made by the $\alpha_{k'}$ for all $k' \leq k$, so the following is easily checked:

Fact 6.15. $\alpha_0 \sqsubseteq \alpha_1 \sqsubseteq \alpha_2 \sqsubseteq \cdots \sqsubseteq H$.

To verify Condition 1, we show that $A^i \neq f^i_j(H)$ for all i and j. Suppose $A^i = f^i_j(H)$ for some i and j. Let $k_0 = \langle i, j\rangle$, and let $\sigma = H[0..k_0-1] \cup \alpha_{k_0-1}$. By the definition of α_{k_0}, it must be the case that $A^i = f^i_j(C)$ for all C with $\sigma \sqsubseteq C$, otherwise f^i_j would have been diagonalized against by α_{k_0} and would thus fail to reduce A^i to H. Let q_0 be smallest such that $q_0 > i$ and $\sigma(\langle q', y\rangle)$ is undefined for all y and $q' \geq q_0$. We will show that $d^{i,j}_{n;q_0}$ succeeds on H_n for some $n < q_0$, contradicting Fact 6.14 above.

For notational convenience, let $A = A^i$, $f = f^i_j$, and for all $k \geq i$ and ℓ let $d_k = d^{i,j}_{k;q_0}$ and $d_{k;\ell} = d^{i,j}_{k;q_0;\ell}$. For any language C and $m \in \mathbb{N}$, we let y_m and $\mathbf{E}^C(m)$ be as before. For any ℓ and $s_u \geq y_{q_0\ell}$ we have

$$\Pr_C[\mathbf{E}^C(q_0\ell) \mid \xi^H(q_0, s_u) \sqsubseteq C] = 1$$

by the definition of q_0 and $y_{q_0\ell}$, and thus

$$2^{q_0\ell} \cdot \prod_{k=i}^{q_0-1} d_{k;\ell}(H_k[0..u-1])$$

$$\geq \prod_{k=i}^{q_0-1} \left(2^\ell \cdot d_{k;\ell}(H_k[0..u-1])\right)$$

$$= \prod_{k=i}^{q_0-1} 2^u \cdot \Pr_C\left[\xi^H(k, y_{q_0\ell}) \cup (\{k\} \times H_k[0..u-1]) \sqsubseteq C \mid \xi^H(k, y_{q_0\ell}) \sqsubseteq C \,\&\, \mathbf{E}^C(q_0\ell)\right]$$

$$= \prod_{k=i}^{q_0-1} 2^u \cdot \Pr_C\left[\xi^H(k, s_u) \cup (\{k\} \times H_k[0..u-1]) \sqsubseteq C \mid \xi^H(k, s_u) \sqsubseteq C \,\&\, \mathbf{E}^C(q_0\ell)\right]$$

(by Remark 6.12)

$$= \prod_{k=i}^{q_0-1} 2^u \cdot \Pr_C\left[\xi^H(k+1, s_u) \sqsubseteq C \mid \xi^H(k, s_u) \sqsubseteq C \,\&\, \mathbf{E}^C(q_0\ell)\right]$$

$$= \prod_{k=i}^{q_0-1} 2^u \cdot \frac{\Pr_C[\xi^H(k+1, s_u) \sqsubseteq C \,\&\, \mathbf{E}^C(q_0\ell)]}{\Pr_C[\xi^H(k, s_u) \sqsubseteq C \,\&\, \mathbf{E}^C(q_0\ell)]}$$

$$= 2^{(q_0-i)u} \cdot \frac{\Pr_C[\xi^H(q_0, s_u) \sqsubseteq C \,\&\, \mathbf{E}^C(q_0\ell)]}{\Pr_C[\xi^H(i, s_u) \sqsubseteq C \,\&\, \mathbf{E}^C(q_0\ell)]}$$

$$= 2^{(q_0-i)u} \cdot \frac{\Pr_C[\mathbf{E}^C(q_0\ell) \mid \xi^H(q_0, s_u) \sqsubseteq C] \cdot \Pr_C[\xi^H(q_0, s_u) \sqsubseteq C]}{\Pr_C[\mathbf{E}^C(q_0\ell) \mid \xi^H(i, s_u) \sqsubseteq C] \cdot \Pr_C[\xi^H(i, s_u) \sqsubseteq C]}$$

$$= \frac{\Pr_C[\mathbf{E}^C(q_0\ell) \mid \xi^H(q_0, s_u) \sqsubseteq C]}{\Pr_C[\mathbf{E}^C(q_0\ell) \mid \xi^H(i, s_u) \sqsubseteq C]}$$

$$= \frac{1}{\Pr_C\ [\mathbf{E}^C(q_0\ell) \mid \xi^H(i, s_u) \sqsubseteq C]}$$

$$= \frac{1}{\Pr_C\left[\mathbf{E}^C(q_0\ell)\right]}$$

$$\geq 2^{q_0\ell},$$

the last inequality following from Fact 6.11. Therefore,

$$\prod_{k=i}^{q_0-1} d_{k;\ell}(H_k[0..u-1]) \geq 1$$

for all $s_u \geq y_{q_0\ell}$, which implies that $d_{k;\ell}(H_k[0..u-1]) \geq 1$ for at least one k between i and $q_0 - 1$. Since q_0 is fixed and ℓ was chosen arbitrarily, by the Pigeon-Hole Principle there must be some n_0 with $i \leq n_0 < q_0$ such that for infinitely many ℓ, $d_{n_0;\ell}(H_{n_0}[0..u-1]) \geq 1$ for all $s_u \geq y_{q_0\ell}$. This in turn implies that the martingale

$$d_{n_0} = \sum_{\ell=1}^{\infty} d_{n_0;\ell}$$

succeeds on H, contradicting Fact 6.14.

Thus $A^i \neq f_j^i(H)$ for all i and j, and Condition 1 is satisfied. ■

It would be interesting to see how far can the martingale diagonalization technique be pushed, that is, for which classes and reducibilities we can construct languages that are hard but not weakly hard.

Very recently, Ambos-Spies, Terwijn and Zheng have shown that almost every language in E is $\leq^{\mathrm{p}}_{\mathrm{m}}$-weakly-complete and not $\leq^{\mathrm{p}}_{\mathrm{btt}}$-complete [AmboTeZ]. This, together with the results in [JuedLu94a] and [AmboNeT], implies the next result that summarizes the actual knowledge of measure of cones in E

Theorem 6.16. ([JuedLu94a], [AmboTeZ], [AmboNeT]) For almost every $A \in \mathrm{E}$, $\mathrm{P}_{\mathrm{btt}}(A)$ does not have measure 0 in E. For each $k \in \mathbb{N}$, for almost every $A \in \mathrm{E}$, $\mathrm{P}^{-1}_{k\text{-tt}}(A)$ has measure 0 in E.

The next open question is whether this is the case for other reducibilities and classes. As a curiosity, let us mention that in [BuhrMa] we construct a set A such that both $\mathrm{P}_{\mathrm{tt}}(A)$ and $\mathrm{P}^{-1}_{k\text{-tt}}(A)$ (for every $k \in \mathbb{N}$) have measure 0 in E.

Another very recent result by Juedes and Lutz on weakly-complete languages is the following:

Theorem 6.17. Every $\leq^{\mathrm{p}}_{\mathrm{m}}$-weakly-complete for E is $\leq^{\mathrm{p}}_{\mathrm{m}}$-weakly-complete for E_2. There exists a $\leq^{\mathrm{p}}_{\mathrm{m}}$-weakly-complete for E_2 that is not $\leq^{\mathrm{p}}_{\mathrm{m}}$-weakly-complete for E.

Another interesting open problem is the existence of languages A such that $\mu(R^{-1}(A) \mid \mathcal{C}) = 0$ for each class $\mathcal{C}$ and reducibility R, that is, what we denoted above as R-incompressible languages. Notice that the existence of an R-incompressible language for $\mathcal{C}$ is equivalent to the fact that the class of R-complete problems has measure 0 in $\mathcal{C}$. (Here we have seen that there exists a $\leq^{\mathrm{p}}_{\mathrm{tt}}$-incompressible for ESPACE. Recently this has been shown for the class E with reducibility $\leq^{\mathrm{p}}_{\mathrm{btt}}$ in [AmboTeZ].)

6.3 On the robustness of ALMOST-R

If R is a reducibility, then ALMOST-R is defined to be the class

$$\{A \mid \Pr(R^{-1}(A)) = 1\}.$$

Book, Lutz, and Wagner [BookLuW] showed that for every bounded (that is, recursively presentable) reducibility, ALMOST-R = $R(\mathrm{rand}) \cap \mathrm{REC}$, where rand denotes the class of algorithmically random languages in the sense of Martin-Löf [Mart], and REC denotes the class of recursive languages. Book [Book94] extended this characterization for certain bounded reducibilities called “appropriate” (all of the standard reducibilities used in structural complexity theory are appropriate) by showing the Random Oracle Characterization, namely that for every $B \in \mathrm{rand}$, ALMOST-R = $R(B) \cap \mathrm{REC}$, and the Independent Pair Characterization, namely that for every B and C such that $B \oplus C \in \mathrm{rand}$, ALMOST-$R$ = $R(B) \cap R(C)$.

While different classes are obtained in the characterization of ALMOST-R as $R(\mathrm{rand}) \cap \mathrm{REC}$ by considering different reducibilities R, here we are concerned with the possibility of obtaining different classes by considering as parameter values the classes rand and REC. In particular, we investigate the result of substituting specific subclasses of rand for rand itself. We find that if we substitute a class based on Kurtz's notion of "n-randomness" (defined in [Kurt]) and simultaneously substitute the class Δ_n^0 (from the arithmetical hierarchy of languages) for the class REC, then once again the result is ALMOST-R. That is, $R(n\text{-rand}) \cap \Delta_n^0 = \text{ALMOST-}R$ (Theorem 6.20).

Our new characterizations of classes having the form ALMOST-R imply a robustness property of these classes. The parameters $\mathcal{C}$ and $\mathcal{D}$ in ALMOST-$R = R(\mathcal{C}) \cap \mathcal{D}$ may vary, while the result is always ALMOST-R.

Next we develop our results about "n-randomness." First we review the concept of the arithmetical hierarchy of *classes of* languages due to Kleene (see Rogers [Roge] for background).

If L is a language and $\mathcal{C} \subseteq \{0,1\}^\infty$ is a class, then $L \cdot \mathcal{C}$ denotes the class $\{w\psi \bigm| w \in L, \psi \in \mathcal{C}\}$.

A class of languages X is an *open class* when there is an $A \subseteq \{0,1\}^*$ such that

$$X = A \cdot \{0,1\}^\infty.$$

We say that X is a *closed class* when X^c is an open class.

A class of languages is *recursively open* if it is of the form $A \cdot \{0,1\}^\infty$ for some recursively enumerable set $A \subseteq \{0,1\}^*$. A class of languages is *recursively closed* if it is the complement of some recursively open set.

Notice that if $\mathcal{C}$ is a countable union or intersection of (recursively) open or closed sets, then $\mathcal{C}$ is Lebesgue-measurable and so $\Pr(\mathcal{C})$ is defined. Since there are only countably many recursively open sets, every intersection of recursively open sets is a countable intersection of such sets, and hence is Lebesgue-measurable; similarly every union of recursively closed sets is Lebesgue-measurable.

Kleene's arithmetical hierarchy of classes is defined as follows.

(i) Let Σ_1^0 be defined as $\{A \mid A \text{ is recursively open}\}$. We fix an enumeration of Σ_1^0 as follows: let $\{M_i \bigm| i \in \mathbb{N}\}$ be a recursive enumeration of all Turing machines (so that $\{L(M_i) \bigm| i \in \mathbb{N}\}$ is the class of recursively enumerable sets). If $A_i = L(M_i) \cdot \{0,1\}^\infty$, then $\Sigma_1^0 = \{A_i \mid i \in \mathbb{N}\}$.

(ii) We say that $\{C_j \mid j \in \mathbb{N}\}$ is a *uniform sequence in* Σ_1^0 if there exists a total recursive function g such that for every $j \in \mathbb{N}$, $C_j = A_{g(j)}$.

(iii) For every $n \geq 1$, $\Pi_n^0 = \{A \mid A^c \in \Sigma_n^0\}$.

(iv) We say that $\{D_j \mid j \in \mathbb{N}\}$ is a *uniform sequence in* Π_n^0 if there exists a uniform sequence in Σ_n^0, $\{C_j \mid j \in \mathbb{N}\}$, such that for every $j \in \mathbb{N}$, $D_j = (C_j)^c$.

(v) For every $n \geq 1$, $B \in \Sigma_{n+1}^0$ if there exists a uniform sequence in Π_n^0, $\{D_j \mid j \in \mathbb{N}\}$, such that $B = \bigcup_k D_k$.

(vi) We say that $\{C_j \mid j \in \mathbb{N}\}$ is a *uniform sequence in* Σ_{n+1}^0 if there exists a uniform sequence in Π_n^0, $\{D_{\langle j,k\rangle} \mid j,k \in \mathbb{N}\}$, such that for every $j \in \mathbb{N}$, $C_j = \bigcup_k D_{\langle j,k\rangle}$.

Note that classically the same notation is used for both the arithmetical hierarchy of languages defined in Chapter 1 (where Σ_n^0 denotes a set of languages) and the arithmetical hierarchy of classes of languages we just defined (where Σ_n^0 denotes a set of classes). The meaning in each case will be clear from the context.

Now we define the concepts of "n-constructive null cover" and "n-random language" in a similar way to the introduction of null covers and random languages in [BookLuW].

For $n > 0$, a class X of languages has an *n-constructive null cover* if there exists a uniform sequence in Σ_n^0, $\{C_k \mid k \in \mathbb{N}\}$, such that

(i) for every $k \in \mathbb{N}$, $X \subseteq C_k$, and

(ii) for every $k \in \mathbb{N}$, $\Pr(C_k) < 2^{-k}$.

Notice that condition (ii) implies that every class with an n-constructive null cover has probability 0.

Let NULL_n denote the union of all classes that have an n-constructive null cover.

Notice that $\mathrm{NULL}_n \subseteq \mathrm{NULL}_{n+1}$. In the case of $n = 1$, we refer to the class as NULL, that is, $\mathrm{NULL}_1 = \mathrm{NULL}$.

The class rand of *algorithmically random languages* was defined by Martin-Löf [Mart] as $\mathrm{rand} = \{0,1\}^\infty - \mathrm{NULL}$.

Here we define, for each $n > 0$, the class n-rand by $n\text{-rand} = \{0,1\}^\infty - \mathrm{NULL}_n$, and the class ω-rand as $\omega\text{-rand} = \bigcap_n n\text{-rand}$.

Since $\mathrm{NULL}_n \subseteq \mathrm{NULL}_{n+1}$, $n+1\text{-rand} \subseteq n\text{-rand}$. Since $\mathrm{NULL}_1 = \mathrm{NULL}$, $1\text{-rand} = \mathrm{rand}$.

A reducibility R will be called *appropriate* if (i) it is bounded, (ii) for any language A, $R(A)$ is closed under finite variations, and (iii) for any language L, $R^{-1}(L)$ is closed under finite variations and under finite translations, as defined in Chapter 1.

The reader should note that the reducibilities commonly used in structural complexity theory meet the conditions for being appropriate.

If R is a bounded reducibility and $n > 0$, then define $\mathrm{ALMOST}_n\text{-}R$ as the class

$$\mathrm{ALMOST}_n\text{-}R = \{A \mid n\text{-rand} \subseteq R^{-1}(A)\},$$

and the class $\mathrm{ALMOST}_\omega\text{-}R$ by

$$\mathrm{ALMOST}_\omega\text{-}R = \{A \mid \omega\text{-rand} \subseteq R^{-1}(A)\}.$$

In [BookLuW] Book, Lutz, and Wagner studied the classes of the form ALMOST-R and related them to the class rand by showing that

$$\mathrm{ALMOST}\text{-}R = R(\mathrm{rand}) \cap \mathrm{REC}.$$

The main result of this section is that each class $\mathrm{ALMOST}_n\text{-}R$ is related to the class n-rand in a very similar way, and that $\mathrm{ALMOST}_n\text{-}R = \mathrm{ALMOST}\text{-}R$. We also obtain similar results for $\mathrm{ALMOST}_\omega\text{-}R$ and ω-rand.

We begin with a technical lemma stating that for any language B in Δ_n^0, $R^{-1}(B)$ is a class in Σ_{n+1}^0. This will be useful in the proof of our main theorem.

Lemma 6.18. If R is a bounded reducibility and B is a language in Δ_n^0, then $R^{-1}(B)$ is a class in Σ_{n+1}^0.

Proof . We consider only the case where n is odd, the other case being analogous.

Let g be a recursive presentation of R. For every $j \in \mathbb{N}$, let $R_j^{-1}(B) = \{A \mid L(M_{g(j)}, A) = B\}$. Then $R^{-1}(B) = \bigcup_j R_j^{-1}(B)$, and it suffices to show that if $B \in \Delta_n^0$, then $\{R_j^{-1}(B) \mid j \in \mathbb{N}\}$ is a uniform sequence in Π_n^0, or equivalently, $\{\left(R_j^{-1}(B)\right)^c \mid j \in \mathbb{N}\}$ is a uniform sequence in Σ_n^0.

Since $B \in \Delta_n^0$, there exist recursive languages C and D such that for every $x \in \{0,1\}^*$,

(i) $x \in B$ if and only if $\exists m_1 \forall m_2 \ldots \exists m_n(\langle x, m_1, \ldots, m_n\rangle \in C)$,

(ii) $x \notin B$ if and only if $\exists m_1 \forall m_2 \ldots \exists m_n(\langle x, m_1, \ldots, m_n\rangle \in D))$.

Fix $j \in \mathbb{N}$. For each $x, m_1, m_2, \ldots, m_{n-1} \in \{0,1\}^*$, we define the following two classes

$$Y^j_{x,m_1,m_2\ldots,m_{n-1}} = \{A \mid \exists m_n \langle x, m_1, \ldots, m_n\rangle \in C \text{ and } L(M_{g(j)}, A)(x) = 0\},$$

and

$$Z^j_{x,m_1,m_2\ldots,m_{n-1}} = \{A \mid \exists m_n \langle x, m_1, \ldots, m_n\rangle \in D \text{ and } L(M_{g(j)}, A)(x) = 1\}.$$

Using these classes, we can express $R_j^{-1}(B)^c$ as follows

$$R_j^{-1}(B)^c = \bigcup_x \Big(\bigcup_{m_1} \bigcap_{m_2} \cdots \bigcap_{m_{n-1}} (Y^j_{x,m_1,m_2\ldots,m_{n-1}} \cup Z^j_{x,m_1,m_2\ldots,m_{n-1}})\Big) \tag{6.1}$$

Next we show that for fixed $x, m_1, m_2 \ldots, m_{n-1} \in \{0,1\}^*$, the class $Y^j_{x,m_1,m_2\ldots,m_{n-1}}$ is recursively open. To do this we define a partial recursive function $h^j_{x,m_1,m_2\ldots,m_{n-1}}$ as follows. For $m_n, z \in \{0,1\}^*$, if $\langle x, m_1, \ldots, m_n\rangle \in C$, $L(M_{g(j)}, z0^\omega)(x) = 0$ and $L(M_{g(j)}, z0^\omega)(x)$ needs only the initial part z of $z0^\omega$, then $h^j_{x,m_1,m_2\ldots,m_{n-1}}(z, m_n) = z$. Otherwise, $h^j_{x,m_1,m_2\ldots,m_{n-1}}(z, m_n)$ is undefined.

From the definition of $Y^j_{x,m_1,m_2\ldots,m_{n-1}}$ we know that $A \in Y^j_{x,m_1,m_2\ldots,m_{n-1}}$ if and only if there exists a prefix z of A such that $\langle x, m_1, \ldots, m_n\rangle \in C$, $L(M_{g(j)}, z0^\omega)(x) = 0$ and $L(M_{g(j)}, z0^\omega)(x)$ needs only the initial part z of $z0^\omega$. But this is exactly the definition of z being in the range of $h^j_{x,m_1,m_2\ldots,m_{n-1}}$. Thus $Y^j_{x,m_1,m_2\ldots,m_{n-1}} = \text{range}(h^j_{x,m_1,m_2\ldots,m_{n-1}}) \cdot \{0,1\}^\infty$, and $Y^j_{x,m_1,m_2\ldots,m_{n-1}}$ is recursively open. By a similar argument $Z^j_{x,m_1,m_2\ldots,m_{n-1}}$ is recursively open, using functions $f^j_{x,m_1,m_2\ldots,m_{n-1}}$ defined as follows. For $m_n, z \in \{0,1\}^*$, if $\langle x, m_1, \ldots, m_n\rangle \in D$, $L(M_{g(j)}, z0^\omega)(x) = 1$ and $L(M_{g(j)}, z0^\omega)(x)$ needs only the initial part z of $z0^\omega$, then $f^j_{x,m_1,m_2\ldots,m_{n-1}}(z, m_n) = z$. Otherwise, $f^j_{x,m_1,m_2\ldots,m_{n-1}}(z, m_n)$ is undefined.

We define a recursive function F that is the uniform version of all h's and f's as follows. For every $j \in \mathbb{N}$, $x, m_1, m_2 \ldots, m_{n-1}, m_n, z \in \{0,1\}^*$,

$$F(j, x, m_1, m_2 \ldots, m_{n-1}, m_n, z0) = h^j_{x,m_1,m_2\ldots,m_{n-1}}(m_n, z),$$

$$F(j, x, m_1, m_2 \ldots, m_{n-1}, m_n, z1) = f^j_{x,m_1,m_2\ldots,m_{n-1}}(m_n, z).$$

F witnesses the fact that the sequence of classes

$$\{ \text{ range}(h^j_{x,m_1,m_2\ldots,m_{n-1}}) \cdot \{0,1\}^\infty \ \cup \ \text{range}(f^j_{x,m_1,m_2\ldots,m_{n-1}}) \cdot \{0,1\}^\infty$$

$$\mid \ j \in \mathbb{N}, \ x, m_1, m_2, \ldots, m_{n-1} \ \in \{0,1\}^* \ \}$$

is a uniform sequence in Σ^0_1.

To complete the proof note that $\{R_j^{-1}(B)^c \mid j \in \mathbb{N}\}$ can be seen to be a uniform sequence in Σ^0_n by using the expression of $R_j^{-1}(B)^c$ in Equation (6.1), and the facts that

$$Y^j_{x,m_1,m_2\ldots,m_{n-1}} = \text{range}(h^j_{x,m_1,m_2\ldots,m_{n-1}}) \cdot \{0,1\}^\infty,$$

and

$$Z^j_{x,m_1,m_2\ldots,m_{n-1}} = \text{range}(f^j_{x,m_1,m_2\ldots,m_{n-1}}) \cdot \{0,1\}^\infty.$$

■

In the proof of our main theorem we also use the following lemma, Theorem IV.2.2 in [Kaut].

Lemma 6.19. [Kaut] Let X be a class in Σ^0_{n+1} that is closed under finite variations and finite translations. Then either $X \cap n$-rand $= \emptyset$ or n-rand $\subseteq X$.

Now we have our main result.

Theorem 6.20. For any appropriate reducibility R and any $n > 0$,

a) for every $B \in n$-rand, ALMOST$_n$-$R = R(B) \cap \Delta^0_n$;

b) ALMOST$_n$-$R = R(n\text{-rand}) \cap \Delta^0_n$;

c) ALMOST$_n$-R = ALMOST-R.

Proof . a) Fix $B \in n$-rand. First, we show that ALMOST$_n$-$R \subseteq R(B) \cap \Delta^0_n$. Let $A \in$ ALMOST$_n$-R. By definition of ALMOST$_n$-R, n-rand $\subseteq R^{-1}(A)$ thus $A \in R(B)$. Since NULL$_n$ is a countable union of classes having probability 0, Pr(n-rand) $= 1$ which implies that for every $A \in$ ALMOST$_n$-R, $\Pr(R^{-1}(A)) = 1$, and ALMOST$_n$-$R \subseteq$ ALMOST-$R \subseteq$ REC. Thus ALMOST$_n$-$R \subseteq R(B) \cap \text{REC} \subseteq R(B) \cap \Delta^0_n$.

Second, we show that $R(B) \cap \Delta^0_n \subseteq$ ALMOST$_n$-R. Let $A \in R(B) \cap \Delta^0_n$. By Lemma 6.18 since $A \in \Delta^0_n$, $R^{-1}(A) \in \Sigma^0_{n+1}$. R is an appropriate reducibility as defined in the preliminaries, therefore $R^{-1}(A)$ is closed under finite variations and is closed under under finite translations. By Lemma 6.19, either n-rand $\subseteq R^{-1}(A)$ or n-rand $\cap R^{-1}(A) = \emptyset$. But $B \in n$-rand $\cap R^{-1}(A)$, therefore n-rand $\subseteq R^{-1}(A)$ and $A \in$ ALMOST$_n$-R.

b) Is a direct consequence of a).

c) We have argued in a) that ALMOST$_n$-$R \subseteq$ ALMOST-R.

To see that ALMOST-$R \subseteq$ ALMOST$_n$-R, take $A \in$ ALMOST-R. Since

$\Pr(R^{-1}(A)) = 1$ and Pr(n-rand) $= 1$ then $R^{-1}(A) \cap n$-rand $\neq \emptyset$. Thus $A \in R(n\text{-rand}) \cap$ REC and $A \in$ ALMOST$_n$-R by b). ■

Thus, Theorem 6.20 extends the Random Oracle Characterization to classes having the form ALMOST$_n$-R by showing that for every $n > 0$ and every $B \in n$-rand, ALMOST-$R = R(B) \cap \Delta_n^0 = R(n\text{-rand}) \cap \Delta_n^0 =$ ALMOST$_n$-R. Notice that since ALMOST-R is a recursive class, these results show that there are no languages from $\Delta_n^0 - \text{REC}$ in $R(n\text{-rand})$, that is, oracles in n-rand are useless for $\Delta_n^0 - \text{REC}$.

We also show that $R(\omega\text{-rand}) \cap AH =$ ALMOST-R, where AH denotes the arithmetical hierarchy of languages, and ω-rand corresponds to the concept of "ω-randomness" as defined in [Kaut].

Theorem 6.21. For any appropriate reducibility R,

a) for every $B \in \omega$-rand, ALMOST$_\omega$-$R = R(B) \cap AH$;

b) ALMOST$_\omega$-$R = R(\omega\text{-rand}) \cap AH$;

c) ALMOST$_\omega$-$R =$ ALMOST-R.

Proof . The proof uses similar arguments to those in the proof of Theorem 6.20. Remark that since ω-rand is a countable intersection of classes having probability 1, it has probability 1. ■

To end this section, we briefly comment on the possibility of characterizing ALMOST-R in terms of Γ-randomness as we have done with Martin-Löf and n-randomness. This would produce resource-bounded measure characterizations of interesting classes.

This is a difficult problem that relates directly to the measurability of upper cones. If we define

$$\text{ALMOST}_\Gamma\text{-}R = \{A \mid \Gamma\text{-rand} \subseteq R^{-1}(A)\},$$

then since $\Pr(\Gamma\text{-rand}) = 1$, clearly ALMOST$_\Gamma$-$R \subseteq$ ALMOST-R. But to see the converse, that is, ALMOST-$R \subseteq$ ALMOST$_\Gamma$-R, we need that for each $A \in$ ALMOST-R, Γ-rand $\subseteq R^{-1}(A)$. We would have an answer if we knew the Γ-measure of $R^{-1}(A)$ for each language A, that is, if we could show that for every A, $R^{-1}(A)$ is Γ-measurable. But this is not even known for the simplest reducibilities, such as $\leq_{\mathrm{m}}^{\mathrm{p}}$.

Let us only remark a first step in this direction. For all natural reducibilities, it trivially holds that for every A, $R(\emptyset) \subseteq R(A)$. If, besides, R is a reducibility such that ALMOST-$R \subseteq R(\emptyset)$, such as $\leq_{\mathrm{btt}}^{\mathrm{p}}$, then ALMOST-$R =$ ALMOST$_\Gamma$-R and for every $B \in \Gamma$-rand,

$$\text{ALMOST-}R \subseteq R(B) \cap \text{REC}.$$

Lutz and Martin (personal communication) have considered the following situation: take a reducibility R and restrict it so that only a bounded number of queries can be made (making it like a "bounded truth-table" or "bounded Turing" reducibility) while maintaining the bounds on computational complexity. If R_b denotes the result, then $R_b(\text{rand}) \cap \Sigma_1^0 =$ ALMOST-R_b.

Kautz and Lutz (personal communication) went in the other direction. If R is a reducibility that is not bounded truth-table or bounded Turing, then $R(\text{rand}) \cap \Sigma_1^0 \neq$ ALMOST-R (but clearly ALMOST-$R \subset R(\text{rand}) \cap \Sigma_1^0$).

It would be interesting to answer these last two questions in a more general form, that is, does $R(n\text{-rand}) \cap \Sigma^0_{n+1}$ equal ALMOST-R?

6.4 Bidimensional measure

Up to this point, this dissertation has been focusing mainly on measure of classes of languages. Nevertheless, there are some important properties that are better expressed in terms of couples of languages, and in order to study whether one of these properties represents the typical behavior or the exception we need to define a measure of classes of pairs of languages.

For instance consider the class of minimal pairs for $\leq^{\mathrm{p}}_{\mathrm{m}}$, that is, pairs (A, B) such that every language C with $C \leq^{\mathrm{p}}_{\mathrm{m}} A$ and $C \leq^{\mathrm{p}}_{\mathrm{m}} B$ must be in P. Schöning constructs in [Schö84] 'arbitrary complex' minimal pairs, which supports the intuition that almost every pair is minimal. It is an open problem to define a bidimensional resource-bounded measure for which this result holds within, for instance, the class E × E.

So far, we have not found a satisfactory definition of bidimensional resource-bounded measure. In this section we first discuss the properties that are desirable for such a measure and then list the characteristics and inconveniences of the most natural approaches.

A bidimensional measure should be a way of comparing the size of classes $X \subseteq \{0,1\}^\infty \times \{0,1\}^\infty$ with the size of some pattern classes, for instance E × E or ESPACE × ESPACE. It is clearly desirable, too, to have a Kolmogorov 0-1 law avalaible. In addition to this there should be some connection between the bidimensional measure of a class X and the measure of its 'projections', as we explain now.

We define for each class $X \subseteq \{0,1\}^\infty \times \{0,1\}^\infty$ and for each language A the projections X_A and X^A as follows

$$X_A = \{B \mid (A, B) \in X\}$$

$$X^A = \{B \mid (B, A) \in X\}.$$

Classically, the relation between the measure of a set and the measure of its projections is formalized by the Fubini Lemma. This lemma says that if $X \subseteq \{0,1\}^\infty \times \{0,1\}^\infty$ has Lebesgue measure 0, then for almost every A the Ath projections of X, X_A and X^A, must have Lebesgue measure 0, that is

Lemma 6.22. Let $X \subseteq \{0,1\}^\infty \times \{0,1\}^\infty$. If $\Pr(X) = 0$ then the following holds

a) $\Pr(\{A \mid \Pr(X^A) = 0\}) = 1$.

b) $\Pr(\{A \mid \Pr(X_A) = 0\}) = 1$.

Intuitively, the projections of a class X can be viewed as the 1-dimensional slices X is made from, thus it is reasonable that if X is very small, many of this slices have to be small too. Note that, as a consequence of this lemma, if X has Lebesgue measure 1 then for almost every A the Ath projections of X, X_A and X^A, must have Lebesgue measure 1.

We are looking for a bidimensional resource-bounded measure where the Fubini Lemma holds. Let us see the use of such a tool. For instance, consider X to be the class of pairs of languages (A, B) such that A is $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducible to B. The results of Juedes and Lutz in [JuedLu94a] say that almost every language $A \in \mathrm{E}$ has the property that ${\mathrm{P_m}}^{-1}(A)$ has measure 0 in E. Thus for almost every A, X_A has measure 0 in E, therefore X cannot have measure 1 in E × E. By the desired Kolmogorov 0-1 law, there would be only two possibilities for X: either X is not measurable in E × E or X has measure 0 in E × E. But very recently, Ambos-Spies, Terwijn and Zheng have shown in [AmboTeZ] that for almost every $A \in \mathrm{E}$, $\mathrm{P_m}(A)$ does not have measure 0 in E. This implies that for almost every A, X^A does not have measure 0 in E, and X cannot have measure 0 in E × E. The conclusion is that X cannot be measurable in E × E, for any definition of measure that has the Fubini property.

Let us mention some possible definitions. We can define bidimensional martingales, corresponding to strategies in the game where a player bets on a hidden pair of languages (A, B). In step n, the player bets on $(A(s_n), B(s_n))$ with the information $(A[0..n-1], B[0..n-1])$. In this case given such a function d, for each $u, v \in \{0,1\}^*$ with $|u| = |v|$,

$$d(u,v) = \frac{d(u0,v0) + d(u0,v1) + d(u1,v0) + d(u1,v1)}{4}.$$

The class of pairs covered by d would be defined as

$$\mathrm{S}^\infty[d] = \{(A,B) \mid \limsup_{n\to\infty} d(A[0..n], B[0..n]) = \infty\},$$

and a class X has measure 0 if X is included in $\mathrm{S}^\infty[d]$ for some d.

With this definition we can show that the class $X = \{(A,B) \mid A \leq^{\mathrm{p}}_{\mathrm{m}} B\}$ we mentioned above has measure 0 in E × E. A martingale for X is a sum of martingales d_i each of them dealing with a particular $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduction M_i. For each $x \in \{0,1\}^*$, if the query of $M_i(x)$ is bigger than x then d_i bets on the query according to x; if the query of $M_i(x)$ is smaller than x then d_i bets on x depending on the query. But from the discussion above, since X has measure 0 in E × E with this formulation, Fubini Lemma does not hold; therefore we discard this definition.

Another possibility is to define a bidimensional martingale $d(u,v)$ as a product of two regular martingales $d^1(u)$ and $d^2(v)$, each of them dealing with a component, and the set covered by d as

$$\mathrm{S}^\infty[d] = \{(A,B) \mid \limsup_{n\to\infty} d^1(A[0..n]) \cdot d^2(B[0..n]) = \infty\}.$$

We could also generalize this to functions of the form

$$d(u,v) = \sum_i d^1_i(u) \cdot d^2_i(v),$$

for d^1 and d^2 two uniform enumerations of martingales (that is, 1-MS). It can be proven that the resulting measure fulfills the Fubini property, but it does not seem easy to find interesting examples where this measure can be used.

References

[AlleSt] E. Allender, M. Strauss: Towards a Measure for P. *Manuscript.*

[AlonSp] N. Alon, J.H. Spencer: *The Probabilistic Method.* Wiley 1992.

[Ambo] K. Ambos-Spies: Randomness, Relativizations, and Polynomial Reducibilities. *Proceedings First Annual Conference on Structure in Complexity Theory* (1986), 23–34.

[AmboFlH] K. Ambos-Spies, H. Fleischhack, H. Huwig: Diagonalizations over Polynomial Time Computable Sets. *Theoretical Computer Science* **51** (1987), 177–204.

[AmboNeT] K. Ambos-Spies, H.-C. Neis, S.A. Terwijn: Genericity and Measure for Exponential Time. *Manuscript.*

[AmboTeZ] K. Ambos-Spies, S.A. Terwijn, X. Zheng: Resource Bounded Randomness and Weakly Complete Problems. *Manuscript.*

[ArviKöM] V. Arvind, J. Köbler, M. Mundhenk: On Bounded Truth-Table, Conjunctive and Randomized Reductions to Sparse Sets. *Proceedings of the 12th Conference FSTTCS,* Lecture Notes in Computer Science Vol. 652 (1992), 140–151.

[Baba] L. Babai: Random Oracles Separate PSPACE from the Polynomial-Time Hierarchy. *Information Processing Letters* **26** (1987), 51–53.

[BakeGiS] T. Baker, J. Gill, R. Solovay: Relativizations of the P=?NP Question. *SIAM Journal on Computing* **4** (1975), 431–442.

[Balc] J.L. Balcázar: Self-Reducibility. *Journal of Computer and System Sciences* **41** (1990), 367–388.

[BalcDíG] J.L. Balcázar, J. Díaz, J. Gabarró: *Structural Complexity I.* EATCS Monographs on Theoretical Computer Science, Vol. 11, Springer-Verlag 1988.

[BalcHeM] J.L. Balcázar, M. Hermo, E. Mayordomo: Characterizations of Logarithmic Advice Complexity Classes. *Proceedings of the IFIP 12th World Computer Congress* (1992), 315–321.

[BalcSc] J.L. Balcázar, U. Schöning: Bi-Immune Sets for Complexity Classes. *Mathematical Systems Theory* **18** (1985), 1–10.

[BellGo] M. Bellare, S. Goldwasser: The Complexity of Decision Versus Search. *SIAM Journal on Computing,* to appear. See also Technical Memorandum MIT/LCS/TM 444, *MIT Laboratory for Computer Science.*

[BennGi] C.H. Bennett, J. Gill: Relative to a Random Oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$ with Probability 1. *SIAM Journal on Computing* **10** (1981), 96–113.

[Berm] L. Berman: On the Structure of Complete Sets: Almost Everywhere Complexity and Infinitely Often Speed-up. *Proceedings of the 17th Symposium on Foundations of Computer Science* (1976), 76–80.

[BermHa] L. Berman, J. Hartmanis: On Isomorphisms and Density of NP and Other Complete Sets. *SIAM Journal on Computing* **6** (1977), 305–332.

[Book74] R.V. Book: Tally Languages and Complexity Classes. *Information and Control* **26** (1974), 186–193.

[Book94] R.V. Book: On Languages Reducible to Algorithmically Random Languages. *SIAM Journal on Computing* (1994), to appear.

[BookLuW] R.V. Book, J.H. Lutz, K.W. Wagner: An Observation on Probability versus Randomness with Applications to Complexity Classes. *Mathematical Systems Theory* **26** (1994), to appear.

[BookMa] R.V. Book, E. Mayordomo: On the Robustness of ALMOST-*R*. Report de Recerca LSI-93-27-R *Universitat Politècnica de Catalunya* (1993).

[Breu] J.M. Breutzmann: Almost-IP=IP. *Manuscript.*

[BuhrHoT] H. Buhrman, S. Homer, L. Torenvliet: Completeness for Nondeterministic Complexity Classes. *Mathematical Systems Theory* **24** (1991), 177–200.

[BuhrMa] H. Buhrman, E. Mayordomo: *In preparation.*

[BuhrSpT] H. Buhrman, E. Spaan, L. Torenvliet: Bounded Reductions. *Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science,* Lecture Notes in Computer Science Vol. 480 (1991), 410–421.

[BuhrTo] H. Buhrman, L. Torenvliet: On the Structure of Complete Sets. *Proceedings 9th Annual Conference on Structure in Complexity Theory,* to appear.

[Cai] J. Cai: With Probability One, a Random Oracle Separates PSPACE from the Polynomial-Time Hierarchy. *Journal of Computer and System Sciences* **38** (1989), 68–85.

[Cher] H. Chernoff: A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *Annals of Mathematical Statistics* **23** (1952), 493–509.

[Chur33] A. Church: A Set of Postulates for the Foundation of Logic. *Annals of Mathematics* **25** (1933), 839–864.

[Chur36] A. Church: An Unsolvable Problem of Elementary Number Theory. *The American Journal of Mathematics* **58** (1936), 345–363.

[Cook] S. Cook: The Complexity of Theorem Proving Procedures. *Proceedings 3rd Annual Symposium on Theory of Computing* (1971), 151–158.

[Erdö] P. Erdös: Some Remarks on the Theory of Graphs. *Bulletin of the American Mathematical Society* **53** (1947), 292–294.

[ErdöSp] P. Erdös, J. Spencer: *Probabilistic Methods in Combinatorics.* Academic Press 1974.

[Fenn] S.A. Fenner: Notions of Resource-Bounded Category and Genericity. *Proceedings 6th Annual Conference on Structure in Complexity Theory* (1991), 196–211.

[FennLuM] S.A. Fenner, J.H. Lutz, E. Mayordomo: There is a Weakly Useful Set that is Not Useful. *In preparation.*

[FlajSt74a] P. Flajolet, J.M. Steyaert: On Sets Having Only Hard Subsets. *Proceedings of the 1st International Colloquium on Automata, Languages and Programming* (1974), 446–457.

[FlajSt74b] P. Flajolet, J.M. Steyaert: Une Généralisation de la Notion d'Ensemble Immune. *RAIRO Informatique Théorique* **8** (1974), 37–48.

[FortSi] L. Fortnow, M. Sipser: Are there Interactive Protocols for co-NP Languages? *Infor-*

mation Processing Letters **28** (1988), 249–251.

[Fu] B. Fu: With Quasi-Linear Queries EXP is Not Polynomial Time Turing Reducible to Sparse Sets. *Proceedings 8th Annual Conference on Structure in Complexity Theory* (1993), 185–191.

[GasaHo] W.I. Gasarch, S. Homer: Relativizations Comparing NP and Exponential Time. *Information and Control* **58** (1983), 88–100.

[GeskHuS] J.G. Geske, D.T. Huynh, J.I. Seiferas: A Note on Almost-Everywhere-Complex Sets and Separating Deterministic-Time-Complexity Classes. *Information and Computation* **92** (1991), 97–104.

[Göde] K. Gödel: On Formally Undecidable Propositions of Principia Mathematica and Related Systems. *Monatshefte fur Math. und Physik* **38** (1931), 173–198.

[HageRü] T. Hagerup, C. Rüb: A Guided Tour of Chernoff Bounds. *Information Processing Letters* **33** (1990), 305–308.

[Hart] J. Hartmanis: Generalized Kolmogorov Complexity and the Structure of Feasible Computations. *Proceedings of the 24th Symposium on Foundations of Computer Science* (1983), 439–445.

[HartImM] J. Hartmanis, N. Immerman, S. Mahaney: One-Way Log-Tape Reductions. *Proceedings of the 19th Symposium on Foundations of Computer Science* (1978), 65–72.

[HartImS] J. Hartmanis, N. Immerman, V. Sewelson: Sparse Sets in NP-P: EXPTIME Versus NEXPTIME. *Information and Computation* **65** (1985), 159–181.

[HartSt] J. Hartmanis, R.E. Stearns: On the Computational Complexity of Algorithms. *Transactions of the American Mathematical Society* **117** (1965), 285–306.

[Hema] L.A. Hemachandra: The Strong Exponential Hierarchy Collapses. *Journal of Computer and System Sciences* **39** (1989), 299–322.

[HemaOgW] L.A. Hemachandra, M. Ogihara, O. Watanabe: How Hard are Sparse Sets? *Proceedings 7th Annual Conference on Structure in Complexity Theory* (1992), 222–238.

[HermMa] M. Hermo, E, Mayordomo: A Note on Polynomial Size Circuits with Low Resource-Bounded Kolmogorov Complexity. *Mathematical Systems Theory,* to appear.

[Home] S. Homer: Structural Properties of Nondeterministic Complete Sets. *Proceedings 5th Annual Conference on Structure in Complexity Theory* (1990), 3–10.

[HomeKuR] S. Homer, S. Kurtz, J. Royer: A Note on 1-Truth-Table Hard Languages. *Theoretical Computer Science* **115** (1993), 383–389.

[HopcUl] J. Hopcroft, J. Ullman: *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley 1979.

[JuedLaL] D.W. Juedes, J.I. Lathrop, J.H. Lutz: Computational Depth and Reducibility. *Proceedings of the 20th International Colloquium on Automata, Languages and Programming* , Lecture Notes in Computer Science Vol. 700 (1993), 277–288. *Theoretical Computer Science,* to appear.

[JuedLu92] D.W. Juedes, J.H. Lutz: Kolmogorov Complexity, Complexity cores, and the Distribution of Hardness. In: *Kolmogorov Complexity and Computational Complex-*

ity (O. Watanabe, editor). EATCS Monographs on Theoretical Computer Science, Springer-Verlag 1992.

[JuedLu94a] D.W. Juedes, J.H. Lutz: The Complexity and Distribution of Hard Problems. *Proceedings of the 34th Symposium on Foundations of Computer Science* (1993), 177–185. *SIAM Journal on Computing,* to appear.

[JuedLu94b] D.W. Juedes, J.H. Lutz: Weak Completeness in E and E_2. *Manuscript.*

[Kann] R. Kannan: Circuit-Size Lower Bounds and Non-Reducibility to Sparse Sets. *Information and Control* **55** (1982), 40–56.

[Karp] R.M. Karp: Reducibility Among Combinatorial Problems. In: *Complexity of Computer Computations* (R. Miller and J. Thatcher, editors). Plenum Press 1972, 85–104.

[KarpLi] R.M. Karp, R.J. Lipton: Some Connections Between Nonuniform and Uniform Complexity Classes. *Proceedings 12th Annual Symposium on Theory of Computing* (1980), 302–309. Also published as: Turing machines that take advice, *L'Enseignement Mathematique* **28** (1982), 191–209.

[Kaut] S. Kautz: *Degrees of Random Sets.* Ph.D.Dissertation, Cornell University, 1991.

[KautMi] S. Kautz, P.B. Miltersen: Relative to a Random Oracle, NP is not Small. *Proceedings 9th Annual Conference on Structure in Complexity Theory,* to appear.

[Klee] S. Kleene: General Recursive Functions of Natural Numbers. *Mathematische Annalen* **112** (1936), 727–742.

[KoMo] K. Ko, D. Moore: Completeness, Approximation and Density. *SIAM Journal on Computing* **10** (1981), 787–796.

[KolmUs] A.N. Kolmogorov, V.A. Uspenskii: Algorithms and Randomness. Translated in *Theory of Probability and its Applications* **32** (1987), 389–412.

[Kurt] S. Kurtz: *Randomness and Genericity in the Degrees ofUnsolvability.* Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 1981.

[KurtMaR] S. Kurtz, S. Mahaney, J. Royer: The Structure of Complete Degrees. In: *Complexity Theory Retrospective* (A. Selman, editor). Springer-Verlag 1990, 108–146.

[Ladn] R. Ladner: The Circuit Value Problem is Log Space Complete for P. *SIGACT News* **7** (1975), 18–20.

[LadnLyS] R. Ladner, N. Lynch, A. Selman: A Comparison of Polynomial-Time Reducibilities. *Theoretical Computer Science* **1** (1975), 103–123.

[Levi] L.A. Levin: Universal Sequential Search Problems. *Problems of Information Transmission* **9** (1973), 265–266.

[LewiStH] P.M. Lewis, R.E. Stearns, J. Hartmanis: Memory Bounds for Recognition of Context-Free and Context Sensitive Languages. *Proceedings of the 6th Annual Syposium on Switching Circuit Theory and Logical Design* (1965), 191–202.

[LongYo] L. Longpré, P. Young: Cook Reducibility is Faster than Karp Reducibility in NP. *Journal of Computer and System Sciences* **41** (1990), 389–401.

[Lutz90] J.H. Lutz: Category and Measure in Complexity Classes. *SIAM Journal on Computing* **19** (1990), 1100–1131.

[Lutz91a] J.H. Lutz: A Pseudorandom Oracle Characterization of BPP. *Proceedings 6th Annual Conference on Structure in Complexity Theory* (1991), 190–195.

[Lutz91b] J.H. Lutz: An Upward Measure Separation Theorem. *Theoretical Computer Science* **81** (1991), 127–135.

[Lutz92] J.H. Lutz: Almost Everywhere High Nonuniform Complexity. *Journal of Computer and System Sciences* **44** (1992), 220–258.

[Lutz93] J.H. Lutz: The Quantitative Structure of Exponential Time. *Proceedings 8th Annual Conference on Structure in Complexity Theory* (1993), 158–175.

[Lutz94a] J.H. Lutz: Weakly Hard Problems. *Proceedings 9th Annual Conference on Structure in Complexity Theory,* to appear.

[Lutz94b] J.H. Lutz: Resource-Bounded Measure. *In preparation.*

[Lutz94c] J.H. Lutz: Intrinsically Pseudorandom Sequences. *In preparation.*

[LutzMa94a] J.H. Lutz, E. Mayordomo: Measure, Stochasticity, and the Density of Hard Languages. *Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science,* Lecture Notes in Computer Science Vol. 665 (1993), 38–47. *SIAM Journal on Computing,* to appear.

[LutzMa94b] J.H. Lutz, E. Mayordomo: Cook Versus Karp-Levin: Separating Reducibilities if NP is not Small. *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science,* Lecture Notes in Computer Science Vol. 775 (1994), 415–426. Accepted in *Theoretical Computer Science.*

[LutzSc] J.H. Lutz, W.J. Schmidt: Circuit Size Relative to Pseudorandom Oracles. *Theoretical Computer Science* **107** (1993), 95–120.

[Lync] N. Lynch: On Reducibility to Complex or Sparse Sets. *Journal of the ACM* **22** (1975), 341–345.

[Maha] S.R. Mahaney: Sparse Complete Sets for NP: Solution of a Conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences* **25** (1982), 130–143.

[Mart] P. Martin-Löf: On the Definition of Infinite Random Sequences. *Information and Control* **9** (1966), 602–619.

[Mayo92a] E. Mayordomo: Almost Every Set in Exponential Time is P-bi-immune. *Proceedings of the 17th International Symposium on Mathematical Foundations of Computer Science,* Lecture Notes in Computer Science Vol. 629 (1992), 392–400. *Theoretical Computer Science,* to appear.

[Mayo92b] E. Mayordomo: Measuring in PSPACE. *Proceedings of the 7th International Meeting of Young Computer Scientists* (1992), 122–129. (Topics in Computer Science, Gordon & Breach, to appear.)

[Meye] A.R. Meyer: Reported in [BermHa].

[NisaWi] N. Nisan, A. Wigderson: Hardness versus Randomness. *Proceedings of the 29th Symposium on Foundations of Computer Science* (1988), 2–11.

[OgihWa] M. Ogihara, O. Watanabe: On Polynomial Bounded Truth-Table Reducibility of NP Sets to Sparse Sets. *SIAM Journal on Computing* **20** (1991), 471–483.

[OrpoSc] P. Orponen, U. Schöning: The Density and Complexity of Polynomial Cores for Intractable Sets. *Information and Control* **70** (1986), 54–68.

[Oxto] J.C. Oxtoby: *Measure and Category.* Graduate Texts in Mathematics, Vol. 2, Springer-Verlag 1980.

[Post36] E.L. Post: Finite Combinatory Process. *Journal of Symbolic Logic* **1** (1936), 103–105.

[Post44] E.L. Post: Recursively Enumerable Sets of Integers and their Decision Problems. *Bulletin American Mathematical Society* **50** (1944), 284–316.

[Roge] H. Rogers: *Theory of Recursive Functions and Effective Computability.* McGraw-Hill 1967.

[Sava] J.E. Savage: Computational work and time of finite machines. *Journal of the ACM* **19** (1972), 660–674.

[Schö84] U. Schöning: Minimal Pairs for P. *Theoretical Computer Science* **31** (1984), 41–48.

[Schö86] U. Schöning: *Complexity and Structure.* Springer-Verlag 1986.

[Schn70] C.P. Schnorr: Klassifikation der Zufallsgesetzenach Komplexität und Ordnung. *Z. Wahrscheinlichkeitstheorie verw. Geb.* **16** (1970), 1–21.

[Schn71a] C.P. Schnorr: A Unified Approach to the Definition of Random Sequences. *Mathematical Systems Theory* **5** (1971), 246–258.

[Schn71b] C.P. Schnorr: Zufälligkeit und Wahrscheinlichkeit. *Lecture Notes in Mathematics***218**(1971).

[Schn73] C.P. Schnorr: Process Complexity and Effective Random Tests. *Journal of Computer and System Sciences* **7** (1973), 376–388.

[Selm79] A.L. Selman: P-Selective Sets, Tally Languages, and the Behavior of Polynomial Time Reducibilities on NP. *Mathematical Systems Theory* **13** (1979), 55–65.

[Selm82] A.L. Selman, Reductions on NP and P-Selective Sets. *Theoretical Computer Science* **19** (1982), 287–304.

[Sham] A. Shamir: IP = PSPACE. *Journal of the Association for Computing Machinery* **39** (1992), 869–877.

[Shan48] C.E. Shannon: A Mathematical Theory of Communication. *Bell System Technical Journal* **27** (1948), 379–423, 623–656.

[Shan49] C.E. Shannon: The Synthesis of Two-Terminal Switching Circuits. *Bell System Technical Journal* **28** (1949), 59–98.

[Spen] J.H. Spencer: *Ten Lectures on the Probabilistic Method.* SIAM 1987.

[SteaHaL] R.E. Stearns, J. Hartmanis, P.M. Lewis: Hierarchies of Memory-Limited Computations. *Proceedings of the 6th Annual Syposium on Switching Circuit Theory and Logical Design* (1965), 179–190.

[Stoc77] L.J. Stockmeyer: The Polynomial-Time Hierarchy, *Theoretical Computer Science* **3** (1977), 1–22.

[Stoc85] L. Stockmeyer: On Approximation Algorithms for #P. *SIAM Journal on Computing* **14** (1985), 849–861.

[StocCh] L. Stockmeyer, A.K. Chandra: Provably Difficult Combinatorial Games. *SIAM Journal on Computing* **8** (1979), 151–174.

[TangBo] S. Tang, R.V. Book: Polynomial-Time Reducibilities and "Almost-all" Oracle Sets. *Theoretical Computer Science* **81** (1991), 36–47.

[Toda] S. Toda: PP is as Hard as the polynomial-Time Hierarchy. *SIAM Journal on Computing* **20** (1991), 865–877.

[Turi36] A. Turing: On Computable Numbers with an Application to the 'Entscheidungsproblem'. *Proceedings of the London Mathematical Society* **2** (1936), 230–265.

[Turi37] A. Turing: Rectification to 'On Computable Numbers...'. *Proceedings of the London Mathematical Society* **4** (1937), 544–546.

[UspeSeS] V.A. Uspenskii, A.L. Semenov, A.Kh. Shen': Can an Individual Sequence of Zeros and Ones be Random? *Russian Mathematical Surveys* **45** (1990), 121–189.

[Wata87a] O. Watanabe: *On the Structure of Intractable Complexity Classes.* PhD Dissertation, Tokyo Institute of Technology, 1987.

[Wata87b] O. Watanabe: A Comparison of Polynomial Time Completeness Notions. *Theoretical Computer Science* **54** (1987), 249–265.

[Wata87c] O. Watanabe: Polynomial Time Reducibility to a Set of Small Density. *Proceedings 2nd Annual Conference on Structure in Complexity Theory* (1987), 138–146.

[WataTa] O. Watanabe, S. Tang: On Polynomial Time Turing and Many-One Completeness in PSPACE. *Theoretical Computer Science* **97** (1992), 199–215.

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

[Youn] P. Young: Some Structural Properties of Polynomial Reducibilities and Sets in NP, *Proceedings 15 Annual Symposium on Theory of Computing* (1983), 392–401.