Almost Every Set in Exponential Time is P-Bi-Immune *

Elvira Mayordomo

Departament de Llenguatges i Sistemes Informàtics, Univ. Politècnica de Catalunya
E-mail mayordomo@lsi.upc.es.
08028 Barcelona, Spain

Abstract. A set A is P-bi-immune if neither A nor its complement has an infinite subset in P. We investigate here the *abundance* of P-bi-immune languages in linear-exponential time (E). We prove that the class of P-bi-immune sets has measure 1 in E. This implies that 'almost' every language 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. A bit further, we show that every p-random (pseudo-random) language is E-bi-immune. Regarding the existence of P-bi-immune sets in NP, we show that if NP does not have measure 0 in E, then NP contains a P-bi-immune set. Another consequence is that the class of $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete languages for E has measure 0 in E.

In contrast, it is shown that in E, and even in REC, the class of P-bi-immune languages lacks the property of Baire (the Baire category analogue of Lebesgue measurability).

* This work was supported by a Spanish Government grant FPI PN90, by the ESPRIT EC project 3075 (ALCOM) and by Accion Integrada HA-047. It was in part done while visiting Iowa State University.

A preliminary version of this paper appeared in Proceedings of the Seventeenth International Symposium on Mathematical foundations of Computer Science (Prague, August 24–28). Springer Lecture Notes in Computer Science 629, 392–400 (1992).

1 Introduction

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 infinite subset of A or its complement is in $\mathcal{C}$.

The notion of immunity was first introduced by Post [18] in recursive function theory. Flajolet and Steyaert transformed it into the complexity theoretic setting in [6] and [7]. Hartmanis and Berman show that E (linear exponential time) contains a P-bi-immune set ([10], observed in [12]), and an application of [9] 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 [2], 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'. To do this we have first to give a precise meaning to this 'almost every' equivalently, we have to classify by size criteria the classes included in E. Secondly we must prove that P-bi-immunity defines a 'large' class within E.

There are mainly two ways of size-classification of classes within E, namely measure in E and category in E, which are particular cases of resource-bounded measure and resource-bounded category, respectively. These two theories were introduced by Lutz in [13] and [14]. (Resource-bounded measure was introduced incorrectly in [13] and corrected in [14].) We will analyze the size of the class of P-bi-immune languages in both settings.

Resource-bounded measure is a generalization of a powerful mathematical tool, Lebesgue measure. Lebesgue measure is useless within recursive classes such as E, because it associates to every countable class measure 0 (that is, minimal size) and every recursive class has a countable amount of languages, so there can be no size distinction among subclasses of E. To introduce his resource-bounded measure, Lutz takes a constructive way of defining Lebesgue measure and bounds the resources allowed in the construction, obtaining meaningful measures in exponential classes, that is to say, measures that allow existence of both small and large subclasses.

We will prove that the class of P-bi-immune languages has measure 1 in E. (This means that its intersection with E is a 'large' subclass of E, see section 3 for definitions.) This implies that almost every language in E is P-bi-immune, and so it extends the previously mentioned result from [10] (in fact, it extends [9] since we will see that for any $c > 0$ almost every language in E is DTIME(2^{cn})-bi-immune). As a corollary, 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 (the class defined by polynomial exponential time, following Lutz's notation). So almost every language in E_2 is E-bi-immune.

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 [8].) We obtain here a sufficient condition for the existence of P-bi-immune sets in NP: if NP does not have measure 0 in E, then NP contains a P-bi-immune set. Lutz has proposed investigation of the hypothesis that NP does not have measure 0 in E, suggesting that it is reasonable relative to our current knowledge, insofar as it has widely believed consequences. It is also shown here that if NP does not have measure 0 in E_2 then NP contains an E-bi-immune set. (Recall that E_2 is the smallest deterministic time complexity class known to contain NP.)

The second method to estimate the size of a class of languages is resource-bounded category. Lutz defines this in [13] by bounding resources in topological Baire category. Classical Baire category differs drastically from Lebesgue measure [17], in the sense that 'large' classes for Baire can be 'small' for Lebesgue, and vice versa. 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 using resource-bounded measure.

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 third and fourth sections of this paper, dealing respectively with resource-bounded measure and resource-bounded category, can be read independently.

2 Preliminaries

First, we review the notion of immunity.

Definition 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 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.

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

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

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. 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.

Let $s_0, s_1, s_2, \ldots$ be the standard enumeration of the strings in $\{0,1\}^*$ in lexicographical order.

From now on we will use the characteristic sequence χ_L of a language L to denote it, where χ_L is defined as follows:

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

So we identify the set $\{0,1\}^\infty$ of all languages over $\{0,1\}$ with the set $\{0,1\}^\infty$ of all sequences. The complement of a set of languages X is $X^c = \{0,1\}^\infty - X = \{0,1\}^\infty - X$.

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 $x \neq y$.

Definition 3. Let $w \in \{0,1\}^*$. $\mathbf{C}_w$ denotes the class of languages $\{x \in \{0,1\}^\infty \mid w \sqsubseteq x\}$.

Let **all** be the class of all functions $f : \{0,1\}^* \to \{0,1\}^*$, and rec be the class of recursive functions in **all**. Let p be the class of polynomial time computable functions, $\mathrm{p}_2$the class of functions computable in time $2^{(\log n)^k}$ for some k, pspace be the class of polynomial space computable functions, and $\mathrm{p}_2$spacethe class of functions computable in space $2^{(\log n)^k}$ for some k. (In the last definitions the output space is bounded in the same way as the working space.)

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.

We let $\mathbf{D} = \{m2^{-n} \mid m,n \in \mathbb{N}\}$ be the set of *nonnegative dyadic rational numbers*.

With the exception of functions mapping into $[0,\infty)$ all our functions are of the form $f : X \to Y$, where each of the sets X, Y is $\mathbb{N}, \{0,1\}^*, \mathbf{D}$, or some cartesian product of these sets. For purposes of computational complexity we regard such functions as mapping $\{0,1\}^*$ into $\{0,1\}^*$. For example, a function $f : \mathbb{N}^2 \times \{0,1\}^* \to \mathbb{N} \times \mathbf{D}$ is interpreted as a function $\tilde{f} : \{0,1\}^* \to \{0,1\}^*$ in order to compute resources. Under this interpretation, $f(i,j,w) = (k,q)$ means that $\tilde{f}(\langle 0^i, \langle 0^j, w\rangle\rangle) = \langle 0^k, \langle u,v\rangle\rangle$, where u and v are the binary representations of the integer and fractional parts of q, respectively.

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

Let RE be the class of recursively enumerable languages, and REC be the class of recursive languages. Let $\mathrm{E} = \bigcup_c \mathrm{DTIME}(2^{cn})$, and $\mathrm{E}_2 = \bigcup_c \mathrm{DTIME}(2^{n^c})$.

We will use Γin the rest of the paper to refer to a class of functions inside **all**.

Next, we can associate with each class of functions Γa class of languages $\mathrm{R}(\Gamma)$. This association will be used in the next two sections, when bounding resources from Lebesgue measure to obtain resource-bounded measure and from Baire category in order to obtain resource-bounded category.

Definition 4. $f \in \Gamma$ is a *constructor* iff $\forall w \in \{0,1\}^*$, $w \sqsubsetneq f(w)$.

Definition 5. 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 6. $\mathrm{R}(\Gamma)$ is the class of languages $\{\mathrm{R}(h) \mid h \text{ a constructor in } \Gamma\}$.

From the classes of functions we mentioned, we obtain well-known classes, as proven in [13]:

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

3 P-bi-immunity and resource-bounded measure

In this section we introduce the method of resource-bounded measure to classify complexity classes depending on their size, and then apply it to prove that the class of P-bi-immune sets has maximal size within 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.

At the end of the section we explain some consequences of this result for $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete languages and for the existence of P-bi-immune sets in NP.

First, we review resource-bounded measure, introduced by Lutz in [14]. Resource-bounded measure is a generalization of classical Lebesgue measure on the real unit interval $[0,1]$. Let us explain the meaning of 'generalization' here.

Lebesgue measure is a function that associates to each 'measurable' subset X of $[0,1]$ a real number $\mathrm{m}(X)$, with $0 \leq \mathrm{m}(X) \leq 1$. The value of $\mathrm{m}(X)$ represents the length of X, in the sense that the measure of an interval $[a\,,b]$ is $b-a$, and for more general subsets X, $\mathrm{m}(X)$ is based on the length of the intervals that approximate X. Lebesgue measure is not defined for every subset of $[0,1]$, and we say that a set X is Lebesgue-measurable if $\mathrm{m}(X)$ is defined.

The class of all languages on $\{0,1\}$, that we have denoted $\{0,1\}^\infty$, can be identified with the interval $[0,1]$ by associating to each $x \in \{0,1\}^\infty$ the real number that has $0.x$ as its standard binary representation. (We do not pay attention to the problem of having two different binary representations for the same number, such as $0.01^\infty = 0.1$, that can be avoided by considering only infinite languages on $\{0,1\}$.) Using this identification Lebesgue measure can be viewed as a measure on $\{0,1\}^\infty$, simply by defining the measure of a class $X \subseteq \{0,1\}^\infty$ as the measure of the subset of $[0,1]$ formed by the images of the elements of X via the above identification.

Given a recursive class $\mathcal{C}$, we could use Lebesgue measure on $\{0,1\}^\infty$ to define μ, a measure on $\mathcal{C}$ as follows

$$\mu(X) = \mathrm{m}(X \cap \mathcal{C}).$$

But this would be useless, because since Lebesgue measure is always 0 for countable sets and recursive classes are always countable, μ would be identically 0.

In order to obtain a non-trivial measure on recursive classes such as REC, E, E_2, ESPACE or E_2SPACE, 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 1: With the information $[\![\lambda \in L]\!]$, the player bets a_1, a part of c_1, either that $0 \in L$ or that $0 \notin L$. If he wins, he gets double, that is $2 \times a_1$, and his capital is now $c_2 = c_1 + a_1$. If he loses, he gets nothing and his capital is now $c_2 = c_1 - a_1$.

*Step n:*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 limit of $\{c_n\}$ as n goes to infinity is infinite.

A *strategy* for this game is a function $a : \{0,1\}^* \to \{0,1\} \times [0, \infty)$ 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 $d_a : \{0,1\}^* \to [0, \infty)$, 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 7. A *martingale* is a function $d : \{0,1\}^* \to [0, \infty)$ satisfying

$$d(w) = \frac{d(w0) + d(w1)}{2} \tag{1}$$

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

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

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

We are now ready to define Lebesgue measure. (Recalling our identification of $\{0,1\}^\infty$ with $[0,1]$, the following is just a restatement of more classical formulations of Lebesgue measure, for instance those in [17].)

Definition 9. A class $X \subseteq \{0,1\}^\infty$ has *Lebesgue-measure 0* iff there exists a martingale d such that, 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 10. 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 [17]), these classes can only have measure 0 or measure 1, if they are measurable at all.

Going back to the initial problem of defining a non trivial measure inside REC, E, E_2 or ESPACE, what we do next is to restrict the martingales that can witness that a class has measure 0. Since martingales are real-valued functions, we cannot require them to be recursive in the ordinary sense. What we do is require them to be approximated by recursive functions, perhaps with additional complexity constraints.

In this way for a recursive class of functions Γ, we define μ_Γ as a restriction of Lebesgue measure to Γ-approximable martingales. We then use μ_Γ to define a nontrivial measure μ on a suitable recursive class $\mathcal{C}$ as $\mu(X) = \mu_\Gamma(X \cap \mathcal{C})$. Let us formalize

Definition 11. A martingale d is Γ-*computable* iff there is a function $\widehat{d} \in \Gamma$, $\widehat{d} : \mathbb{N} \times \{0,1\}^* \to \mathbf{D}$, such that

$$|\widehat{d}(k,w) - d(w)| \leq 2^{-k}$$

for all $k \in \mathbb{N}$, and $w \in \{0,1\}^*$.

We now come to the key definition of resource-bounded measure theory.

Definition 12. A class $X \subseteq \{0,1\}^\infty$ has Γ-*measure 0* iff there exists a Γ-computable martingale d such that, for any $L \in X$, d is successful for L.

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

Definition 13. A set $X \subseteq \{0,1\}^\infty$ has Γ-*measure 1* iff X^c has Γ-measure 0.

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

There exists a resource-bounded generalization of Kolmogorov 0-1 law (see [15]). For this reason we only define Γ-measure 0 and Γ-measure 1.

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 (Theorem 3.13 in [14]), which states that $\mathrm{R}(\Gamma)$ does not have Γ-measure 0.

Theorem 14. ([14]) Measure Conservation Theorem. For every Γ-computable martingale d, there exists a language $L \in \mathrm{R}(\Gamma)$ such that d is not successful for L.

We finally define a meaningful measure in $\mathrm{R}(\Gamma)$.

Definition 15. A set $X \subseteq \{0,1\}^\infty$ has *measure 0 in* $\mathrm{R}(\Gamma)$ iff $X \cap \mathrm{R}(\Gamma)$ has Γ-measure 0.

Definition 16. A set $X \subseteq \{0,1\}^\infty$ has *measure 1 in* $\mathrm{R}(\Gamma)$ iff X^c has measure 0 in $\mathrm{R}(\Gamma)$.

Recall from the preliminaries that taking $\Gamma = \mathrm{p}, \mathrm{p}_2$, pspace and $\mathrm{p}_2$space we obtain $\mathrm{R}(\Gamma) =$ E, E_2, ESPACE and E_2SPACE, respectively. This implies that we have defined a nontrivial measure on the classes E, E_2, ESPACE and E_2SPACE.

As a last measure concept, we introduce the pseudo-random languages, which represent the notion of 'typical' language in this setting.

Definition 17. A language L is Γ-*random* iff it belongs to every Γ-measure-1 class.

The following lemma contains the first elemental properties of resource-bounded measure and is straightforward from the above definitions.

Lemma 18. Let $X,\ Y \subseteq \{0,1\}^\infty$. If $Y \subseteq X$ and X has Γ-measure 0 then Y has Γ-measure 0. Let $X \subseteq \{0,1\}^\infty$. If X has Γ-measure 0 then X has measure 0 in $\mathrm{R}(\Gamma)$.

For the nontrivial measure on E we just defined we can prove our main result on the class of P-bi-immune languages.

Theorem 19. The class of P-bi-immune languages has measure 1 in E.

Proof. Let Y be the class of non-P-bi-immune languages. By Definition 13, if we prove that Y is a p-measure 0 class we have the theorem.

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

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.

Next for each $m \in \mathbb{N}$ we define a martingale $d_m \in \mathrm{p}$ that shows that Y_m has p-measure 0. From these martingales we construct a single p-computable martingale d that witnesses that Y has p-measure 0. This is a standard technique to prove that a set Z that is a 'uniform' union of the p-measure 0 sets $\{Z_m\}$ has p-measure 0. The technique is fully developed by Lutz in Lemma 3.10 of [14].

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

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

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 for each $m \in \mathbb{N}$, d_m fulfills condition (1) in Definition 7, thus it is a martingale. Since $A \in \mathrm{E}$ there exists $c > 0$ such that for each $m \in \mathbb{N}$ d_m can be computed in time $2^{c(\log(|w|))} = |w|^c$ on input w, and thus $d_m \in \mathrm{p}$.

Let us see that for each $m \in \mathbb{N}$, d_m is succesful on all languages in Y_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 $B \in Y_{2i-1}$ $s_n \in B$, ant 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 $\lim_{n\to\infty} d_{2i-1}(\chi_B[0..n]) = \infty$, and d_{i-1} is succesful on B. The proof that d_{2i} witnesses that Y_{2i} has p-measure 0 is analogous.

Now we define a martingale d as follows, for each $w \in \{0,1\}^*$

$$d(w) = \sum_{i=1}^{\infty} 2^{-m} d_m(w).$$

d is well defined because $d(\lambda) = \sum_{i=1}^{\infty} 2^{-m} < \infty$ and $d(w) \leq 2^{|w|} d(\lambda) < \infty$.

Let us see that d is successful on all languages in Y. If $B \in Y$ then there exist $m \in \mathbb{N}$ such that $B \in Y_m$, which implies that $\lim_{n\to\infty} d_m(\chi_B[0..n]) = \infty$. But $\lim_{n\to\infty} d(\chi_B[0..n]) \geq 2^{-m} \cdot \lim_{n\to\infty} d_m(\chi_B[0..n]) = \infty$, thus d is successful on B.

Let us see that d is p-computable. We define $\widehat{d} : \mathbb{N} \times \{0,1\}^* \to \mathbf{D}$ as follows. Let $k \in \mathbb{N}, w \in \{0,1\}^*$

$$\widehat{d}(k,w) = \sum_{i=1}^{|w|+k} 2^{-m} d_m(w).$$

Since for every m, d_m can be computed in time n^c, then $\widehat{d} \in \mathrm{p}$. Let us see that $\widehat{d}$ is a p-approximation of d. Let $k \in \mathbb{N}, w \in \{0,1\}^*$

$$|\widehat{d}(k,w) - d(w)| = \sum_{i=|w|+k+1}^{\infty} 2^{-m} d_m(w) \leq \sum_{i=|w|+k+1}^{\infty} 2^{-m} 2^{|w|} = 2^{-k}.$$

We have that Y has p-measure 0, then Y^c has p-measure 1 and we have completed the proof of the theorem. ∎

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 [16].

Definition 20. 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 [2] 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 21. Almost every set in E has $\{0,1\}^*$ as a complexity core.

In the next Theorem we extend Theorem 19 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 22. Let $\mathcal{C}$ be a complexity class such that there exists $A \in \mathrm{E}$ with $\mathcal{C} \subseteq \{A_i \bigm| i \in \mathbb{N}\}$. Then the class of $\mathcal{C}$-bi-immune languages has p-measure 1, and thus measure 1 in E.

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

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

Proof . Similar to the proof of Theorem 19. ■

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 [19], 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 from [11] and [4] is exactly the same as $\leq^{\mathrm{p}}_{1-\mathrm{tt}}$-completeness.

Corollary 23. 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 [3], 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 19, and from Lemma 18 it has measure 0 in E. The second part is analogous, using Theorem 22 and the fact that no $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set for NE is E-bi-immune (from [3]). ■

Notice that it is not known whether $\mathrm{NE} \subseteq \mathrm{E}_2$. Also, from [11] and [4] every $\leq^{\mathrm{p}}_{1-\mathrm{tt}}$-complete set for NE is $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete.

An important consequence dealing with the existence of P-bi-immune sets in NP states:

Corollary 24. If NP does not have measure 0 in E 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 . The results follow from Theorems 19 and 22 and Lemma 18. ■

We finish this section by seeing that the typical languages for resource-bounded measure are E-bi-immune

Corollary 25. 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 22 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. ∎

4 P-bi-immunity and resource-bounded category

In this section we introduce resource-bonded 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 classes 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 [17]). Lutz defines a resource-bounded category in [13], later studied by Fenner ([5]), 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 26. Let X be a class of languages, let Γ_1 and Γ_2 be two classes of functions. 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 $h \in \Gamma_2$. (recall from section 2 the definition of constructor, $w \subsetneq g(w)$ for every w). 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 4 and 5 the composition of g and h, $h \circ g$, is a constructor and $\mathrm{R}(g,h) = \mathrm{R}(h \circ g)$.

Definition 27. 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 28. 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 29. 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 12 and Definition 28), 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 know.

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

Definition 30. Let X be a class of languages. X is *meager in* $\mathrm{R}(\Gamma)$ iff $X \cap \mathrm{R}(\Gamma)$ is Γ-meager.

Definition 31. 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 [13] 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 30 and 31 we get classical Baire category. In that context, typical languages are called generic. We define here Γ-generic or pseudo-generic languages.

Definition 32. Let L be a language. L is Γ-*generic* iff L belongs to every Γ-co-meager class.

(There exist more restrictive notions of genericity, studied among other people by Ambos-Spies et al. in [1].)

The following lemma states some basic properties of meager sets, and is proved by Lutz in [13].

Lemma 33. 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 34. 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 30 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 19, that is, if 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

$$\exists s_n, s_m \in A_i \text{ with } 0 \leq n, m < |\gamma| \text{ and } \gamma[n] \neq \gamma[m]. \tag{2}$$

We say that index i has been diagonalized in γ, and denote it with the boolean condition $\mathrm{Diagonalized}(i, \gamma)$, when condition (2) holds for this i and γ, that is,

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

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 $\mathrm{Diagonalized}(i, \gamma) =$True.

For $\gamma \in \{0,1\}^*$ and $q \geq |\gamma|$, the set $\mathrm{Diagonalizable}(\gamma, q)$ 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

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

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 $\mathrm{Diagonalized}\big(i, g(\alpha)\big) =$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 against 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 $\mathrm{Diagonalized}(i, \gamma)$=True.

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

$$\mathrm{Diagonalizable}(g(\alpha), 2^{|g(\alpha)|}) \cap \{1, \ldots, |\alpha|\} = \emptyset. \tag{3}$$

```
BEGIN
  INPUT α
  γ := α
  IF Diagonalizable(α, 2^|α|) ∩ {1, ..., |α|} = ∅ THEN γ := α0  {This is to ensure α ⊊ g(α)}
  WHILE Diagonalizable(γ, 2^k) ∩ {1, ..., |α|} ≠ ∅ DO
      k := |γ|
      IF Diagonalizable(γ, k) ∩ {1, ..., |α|} ≠ ∅
      THEN
          i := min{j | j ∈ Diagonalizable(γ, k)}
          n := min{r | s_r ∈ A_i}
          IF γ[n] = 0 THEN γ := γ1
          IF γ[n] = 1 THEN γ := γ0
          {At this point we know that Diagonalized(γ, i) = True,
          since s_n, s_k ∈ A_i and γ[n] ≠ γ[k]}
      ELSE γ := γ0
  END WHILE
  OUTPUT γ
END .
```

Figure 1: Algorithm that computes g.

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) 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) $i \notin$ 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 $\gamma \sqsubseteq g(\alpha_l)$, with $|\gamma| \geq |\alpha|$

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

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 $A_1, \ldots, A_{2^{|z|+1}-2}$ on inputs $s_0, \ldots, s_{2^{|z|+1}-2}$, 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 33 we have that the class of P-bi-immune languages is not p-meager.

Theorem 35. 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 31 we have to see that there is no winning strategy for player II in the game $G[Y \cap \mathrm{E}; \mathbf{all}, \mathrm{p}]$, that is to say: for every constructor $h \in \mathrm{p}$, there exists a constructor $g \in \mathbf{all}$, such that $\mathrm{R}(g, h) \in Y \cap \mathrm{E}$.

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 $\{a_n\}$:

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

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 α

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 γ

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 36. 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 37. For any recursively presentable class $\mathcal{C}$ with $P \subseteq \mathcal{C}$, 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 38. 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 39. 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 40. 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 [17], 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 41. 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 $\widehat{h} : \{0,1\}^* \to \{0,1\}^*$ 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 33 implies that X is meager in $\mathrm{R}(\Gamma)$. This completes the proof. ∎

Lemma 42. 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 43. 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 34 and 35 (extended to the classes E, E_2 and ESPACE), Theorem 37 and Lemma 42. ∎

In contrast with Theorem 43, 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 43 and the remark following Lemma 3.9 in [5] 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.

5 Acknowledgements

I would like to thank Jack Lutz for many helpful ideas and discussions, as well as for providing me with the preliminary version of [14], José Luis Balcázar, who proposed this problem and gave a lot of ideas for the introduction, and Stephen Fenner who remarked that the second part of Theorem 37 also holds. I also thank two anonymous referees for their valuable and thorough comments that have improved the first version of this manuscript.

References

[1] K. Ambos-Spies, H. Fleischhack, H. Huwig: Diagonalizations over Polynomial Time Computable Sets. *Theoretical Computer Science* **51** (1987), 177–204.

[2] J.L. Balcázar, U. Schöning: Bi-Immune Sets for Complexity Classes. *Mathematical Systems Theory* **18** (1985), 1–10.

[3] L. Berman: On the Structure of Complete Sets: Almost Everywhere Complexity and Infinitely Often Speed-up. *Proceedings 17th IEEE Symposium on Foundations of Computer Science* (1976), 76–80.

[4] H. Buhrman, E. Spaan, L. Torenvliet: Bounded Reductions. *Proceedings 8th Annual Symposium on Theoretical Aspects of Computer Science* (1991), 410–421.

[5] S.A. Fenner: Notions of Resource-Bounded Category and Genericity. *Proceedings 6th Annual Conference on Structure in Complexity Theory* (1991), 196–211.

[6] P. Flajolet, J.M. Steyaert: On Sets Having Only Hard Subsets. *Proceedings 1st International Colloquium on Automata, Languages and Programming* (1974), 446–457.

[7] P. Flajolet, J.M. Steyaert: Une Généralisation de la Notion d'Ensemble Immune. *RAIRO Informatique Théorique* **8** (1974), 37–48.

[8] W.I. Gasarch, S. Homer: Relativizations Comparing NP and Exponential Time. *Information and Control* **58** (1983), 88–100.

[9] 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.

[10] J. Hartmanis, L. Berman: On Isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing* **6** (1977), 305–332.

[11] S. Homer, S. Kurtz, J. Royer: A Note on 1-Truth-Table Hard Languages. *Theoretical Computer Science,* to appear.

[12] K. Ko, D. Moore: Completeness, Approximation and Density. *SIAM Journal on Computing* **10** (1981), 787–796.

[13] J.H. Lutz: Category and Measure in Complexity Classes. *SIAM Journal on Computing* **19** (1990), 1100–1131.

[14] J.H. Lutz: Almost Everywhere High Nonuniform Complexity. *Journal of Computer and System Sciences* **44** (1992), 220–258.

[15] J.H. Lutz: Resource-Bounded Measure, in preparation.

[16] N. Lynch: On Reducibility to Complex or Sparse Sets. *Journal of the ACM* **22** (1975), 341–345.

[17] J.C. Oxtoby: *Measure and Category.* Graduate Texts in Mathematics, Vol. 2, Springer-Verlag 1980.

[18] E.L. Post: Recursively Enumerable Sets of Integers and their Decision Problems. *Bulletin American Mathematical Society* **50** (1944), 284–316.

[19] L. Stockmeyer, A.K. Chandra: Provably Difficult Combinatorial Games. *SIAM Journal on Computing* **8** (1979), 151–174.