Resource-Bounded Baire Category: A Stronger Approach

Stephen A. Fenner*
Computer Science Department
University of Southern Maine

August 2, 1996

Abstract

This paper introduces a new definition of resource-bounded Baire category in the style of Lutz. This definition gives an almost-all/almost-none theory of various complexity classes. The meagerness/comeagerness of many more classes can be resolved in the new definition than in previous definitions. For example, almost no sets in **EXP** are **EXP**-complete, and **NP** is **PF**-meager unless **NP** = **EXP**. It is also seen under the new definition that no rec-random set can be (recursively) tt-reducible to any **PF**-generic set. We weaken our definition by putting arbitrary bounds on the length of extension strategies, obtaining a spectrum of different theories of Baire Category that includes Lutz's original definition.

1 Introduction

Both Lebesgue measure and Baire category provide useful and differing notions of "smallness" for a set of reals.[1] In Lebesgue measure, the small sets are those with measure zero (null sets). In Baire category, the small sets are those of first category (meager sets). Both notions of smallness satisfy the following basic properties (see [Oxt80]):

1. All countable sets are small.
2. Any subset of a small set is small.
3. The countable union of small sets is small.
4. The complement of a small set is dense.

We say that a subset of 2^ω is 'large' if its complement is small. An element $A \in 2^\omega$ is 'typical' with respect to a countable collection Γ of large subsets of 2^ω if $A \in \cap\Gamma$. By properties 3 and 4 above, the elements of 2^ω typical for any fixed countable Γ comprise a dense subset of 2^ω. In Lebesgue measure, typical elements of 2^ω are called 'random'; in Baire category, they are called 'generic'.

In a series of seminal papers, Lutz [Lut87, Lut92, Lut90] has introduced resource-bounded versions of both measure and category, then studied the notions of typicality that result. The chief advantage of all these ideas is that subrecursive complexity classes have a nontrivial "almost-all" structure, even though they are countable, and one can then study the typical properties of sets within a given complexity class. Lutz has concentrated mostly on the measure theoretic side; in the present paper, we are interested in the category side.

In [Fen91], we made improvements to Lutz's definition of resource-bounded category, with an eye toward pseudogeneric oracle results. We strengthened Lutz's notion and gave a more general characterization in terms of resource-bounded Banach-Mazur games. Banach-Mazur games are a particularly nice way to reason about

*Partially supported by NSF Grant CCR 92-09833. Earlier versions of some of these results also appeared in [Fen91], and were proved while the author was a graduate student in the University of Chicago Computer Science Department, with the support of a University of Chicago Fellowship.

[1] We identify reals with elements of 2^ω, the set of all infinite binary sequences.

category without having to use the burdensome notation of the original definition. The proofs in [Fen91], however, used a somewhat complicated formalism, and our definitions still did not provide the strongest possible notion of resource-bounded category.

In the present paper, we give a simpler and much more powerful definition of resource-bounded category, and prove a number of properties to be meager or comeager under this new definition—properties whose status with the old definition was either unknown or neither meager nor comeager. We also show that the new definition has all the desirable characteristics of Lutz's old definition. In Section 4 we show just how powerful this new definition is by giving a simple and complete characterization of **PF**-meagerness for all Σ_2^0 classes that are closed under finite variations. For example, we show that either **NP** = **EXP** or **NP** is **PF**-meager.

We also look at the new notion of genericity that arises from our definition, showing that no rec-random set is (recursively) truth-table reducible to any **PF**-generic set. We next define length-bounded category. The length bounds give a whole unified spectrum of notions of category of varying strengths. This spectrum includes Lutz's original definition on one end, and ours on the other. We present various known facts about the spectrum in Figure 5.

2 Definitions and Notation

We let ω be the set $\{0,1,2,\ldots\}$ of natural numbers. We set $\Sigma = \{0,1\}$ and let $2^{<\omega}$ be the set of all finite strings of 0's and 1's (*binary strings*). We will interpret a binary string in two ways: (1) as a natural number via the usual dyadic representation, and (2) as a partial function $\omega \to \Sigma$ whose domain is a finite initial segment of ω. We use λ to denote the empty string. For $A, B \subseteq \omega$, we write $A \triangle B$ for the symmetric difference of A and B.

We use 2^ω to denote the set of infinite sequences of 0's and 1's, which we identify both with total functions $\omega \to \Sigma$ and with subsets of ω in the usual way. Unless otherwise specified, we use $X, Y, Z, \ldots$ to denote subsets of 2^ω, $A, B, C, \ldots$ to denote subsets of ω (elements of 2^ω), lower case Roman letters to denote natural numbers, and lower case Greek letters to denote (binary) strings.

Let σ be a string. We let $|\sigma|$ denote the length of σ, i.e., the cardinality of domain(σ). If $f \in 2^{<\omega} \cup 2^\omega$, we write $\sigma \preceq f$ to mean that f extends σ (σ is a prefix of f), and we write $\sigma \prec f$ to mean $\sigma \preceq f$ and $\sigma \neq f$. If τ is a string, then $\sigma\hat{\ }\tau$ denotes the concatenation of σ followed by τ. (The same holds for numbers via the dyadic representation.) We say σ and τ are *compatible* if either $\sigma \preceq \tau$ or $\tau \preceq \sigma$. We will also sometimes have occasion to use a string σ as an oracle, in which case we interpret σ as the set $\{x \mid \sigma(x)\downarrow = 1\}$.

We can further identify a string σ with the 'interval' $C_\sigma \stackrel{\text{df}}{=} \{A \in 2^\omega \mid \sigma \prec A\}$. The collection of all the C_σ form a basis for the Cantor topology on 2^ω. A set $X \subseteq 2^\omega$ is *dense* if every string has an extension in X, i.e.,

$$(\forall \tau \in 2^{<\omega})(\exists A \in X)\ \tau \prec A.$$

Note that this is the usual definition of density under the Cantor topology.

We fix a standard pairing function $\langle\cdot,\cdot\rangle\colon 2^{<\omega} \times 2^{<\omega} \to 2^{<\omega}$ that is a bijection computable and invertible in polynomial time. All our functions will have one argument, but we will often write $f(x,y)$ for $f(\langle x,y\rangle)$. If f is a function with domain $2^{<\omega}$ and k is a natural number, we define $f_k(x) \stackrel{\text{df}}{=} f(\langle 0^k, x\rangle)$ for all $x \in 2^{<\omega}$.

Finally, for $a, b \in \omega$ we let $a \dot{-} b$ denote $\max(0, a-b)$.

2.1 Finite Extension Strategies

The central concept in the study of resource-bounded category is that of a finite extension strategy.

Definition 2.1 *A* finite extension strategy *is a total function $h\colon 2^{<\omega} \to 2^{<\omega}$ such that $\sigma \preceq h(\sigma)$ for all $\sigma \in 2^{<\omega}$. A string τ* meets *h if there is a $\sigma \in 2^{<\omega}$ such that $h(\sigma) \preceq \tau$. If $A \in 2^\omega$, we say that A* meets *h if some $\tau \prec A$ meets h. The set A* avoids *h if A does not meet h.*

For brevity, we will use the word 'strategy' to refer to a finite extension strategy. The following definitions are adapted from classical point-set topology.

Definition 2.2 *A set $X \subseteq 2^\omega$ is* nowhere dense *if there exists a strategy h such that for all $\sigma \in 2^{<\omega}$, $h(\sigma) \not\prec A$ for all $A \in X$. That is, X is nowhere dense if and only if there is a single strategy that is avoided by every element of X.*

Definition 2.3 *A set $X \subseteq 2^\omega$ is* meager *(or of* first category*) if X is contained in a countable union of nowhere dense sets. Equivalently, X is meager if and only if there is a countable family $h_0, h_1, h_2, \ldots$ of strategies such that every $A \in X$ avoids one or more of the h_i's.*

2.2 Encodings of Strings

We are interested in resource-bounded versions of these notions. We can do this by restricting the resources allowed by a Turing machine to compute finite extension strategies. We must consider, however, the mechanism the machine actually uses to compute the strategy. Lutz [Lut87, Lut90] considered an obvious one: given a string as input, the extension is simply written explicitly on the output tape. Lutz's definition leads to an elegant theory having a number of desirable properties (given below), which lend weight to the thesis that it is a legitimate analogue to classical Baire category; the most important of these properties is that certain complexity classes (although countable) are not meager in a resource-bounded sense, and thus have a nontrivial almost-all structure. Lutz's definition is weak, however. For example, a polynomial time bounded strategy on input σ can only produce an extension whose length is as most a polynomial in $|\sigma|$. Viewing strings as initial segments of oracles, under this scheme we will not in general be able to extend the oracle far enough to diagonalize against a polynomial-time oracle machine. Thus we will not be able to show, for instance, that $\mathbf{P} \neq \mathbf{NP}$ relative to the typical exponential-time set, and indeed this is not the case under Lutz's definitions [Fen91].

In [Fen91] we partially addressed this problem by suggesting an alternate scheme for encoding the outputs of machines computing strategies. There, it was shown that under the alternate scheme, one *did* have enough power to perform the diagonalizations above. Furthermore, the definition of resource-bounded category given by the new scheme retained all the desirable properties of Lutz's original scheme, making it a strictly stronger and equally legitimate theory. The encoding scheme in [Fen91] still falls short of the best possible, however. Moreover, instead of strings, we needed to work with functions with arbitrary finite domain—slightly more awkward and confusing than strings. In the present paper we remedy these problems. We define a mechanism for computing total strategies that is *optimal* in the sense that it is at least as powerful as any "sensible" polynomial-time scheme for decoding outputs, and it retains all of the desirable properties of Lutz's original definition. Furthermore, it will no longer be necessary to deal with arbitrary finite functions, so we can return the emphasis simply to strings.

To define our new notions, we will take a completely different approach than the one taken in [Lut87] or [Fen91]. There, a machine is given an input string σ and outputs some encoding of a string $\tau \succeq \sigma$. The string τ can be recovered via some fixed decoding algorithm applied to the machine's output. In [Lut87], the machine outputs τ unencoded, so the decoding algorithm is the identity function. In [Fen91], the machine outputs a list of pairs of the form (x, b) where $x \in \omega - \text{domain}(\sigma)$ and $b \in \Sigma$; the extension τ (now a finite function not necessarily a string) is the minimum extension of σ such that $\tau(x){\downarrow}= b$ for all pairs (x, b) on the list. Our present approach is to consider τ as a function to be computed one bit at a time, i.e., the machine now takes as inputs both σ and a number n in unary and outputs $\tau(n)$ (or $\perp$ if $\tau(n){\uparrow}$).

Consider the strategy, "given a string σ, extend with Ackerman($|\sigma|$) many zeros." Such a strategy is natural and may be very useful for some types of diagonalizations, but has no hope of being computable in polynomial time if one had to write down the entire output. In fact, none of the encoding schemes described in [Fen91] allow for such a strategy in polynomial time, either. Nevertheless, the key property of this strategy, which will make it legitimate under our definition of polynomial-time category, is that for any σ, the extension—though possibly quite long—is easy to compute *locally.* That is, if τ is the extension and n is any natural number, then the value of $\tau(n)$—0, 1, or undefined—can be computed quickly (time polynomial in $|\sigma| + n$). This method for computing extensions will allow us to use strategies which are at least as powerful as any we could define via a fixed decoding algorithm. Furthermore, we will show that the theory resulting from this approach preserves all the desirable properties of resource-bounded category.

We now make our notions precise. For the rest of the paper we let $\perp$ be some fixed integer other than 0 or 1 (two, for concreteness).

Definition 2.4 *Let h be a strategy and $f: 2^{<\omega} \to 2^{<\omega}$ a total function. The function f* locally computes *h if, for all $\sigma \in 2^{<\omega}$ and $n \in \omega$,*

$$f(\sigma, 0^n) = \begin{cases} \tau(n) & \text{if } \tau(n)\downarrow, \\ \bot & \text{otherwise,} \end{cases}$$

where $\tau \stackrel{\rm df}{=} h(\sigma)$. Given f we can in turn denote the strategy h by $f^{\rm LC}$.

Note that the string σ is given to f in its entirety unencoded, and n is in unary. Note also that we cannot in general effectively enumerate all local strategy computations in some given complexity class. This will concern us later when we define resource-bounded genericity.

2.3 Function Classes

Definition 2.5 *Define*

$$\begin{aligned} \omega^\omega &\stackrel{\rm df}{=} \{f \mid f: \omega \to \omega\} \\ \text{arith} &\stackrel{\rm df}{=} \{f \in \omega^\omega \mid f \text{ is definable in first-order} \\ &\qquad\qquad\qquad\qquad \text{Peano arithmetic}\} \\ \text{rec} &\stackrel{\rm df}{=} \{f \in \omega^\omega \mid f \text{ is recursive}\} \\ \mathbf{PF} &\stackrel{\rm df}{=} \mathbf{DTIMEF}(\text{poly}), \end{aligned}$$

We are primarily interested in the classes above, although our basic results also hold for all function classes closed under a certain natural type of functional tt-reduction.

Definition 2.6 *Let f and g be total functions. We say that f is* polynomial-time functionally truth-table reducible *to g ($f \le_{tt}^{\rm pf} g$) if there exist **PF** functions $Q(x)$ and $D(x, y)$ such that, for all $x \in 2^{<\omega}$,*

$$f(x) = D(x, g(q_1)\#g(q_2)\#\cdots\#g(q_m)),$$

where

$$Q(x) = q_1\#q_2\#\cdots\#q_m.$$

We call the pair (Q, D) a pftt-reduction.

In the rest of the paper, unless stated otherwise, we will assume that Δ is any nonempty function class closed under pftt-reductions. All the classes of Definition 2.5 are easily seen to be closed under pftt-reductions.

2.4 Δ-Category

We will now define meagerness with respect to Δ, and describe its basic properties in Section 3.

Definition 2.7 *A set $X \subseteq 2^\omega$ is* Δ-meager *if there is a function $h \in \Delta$ such that*

- *for all $i \in \omega$, h_i locally computes a strategy, and*
- *for every $A \in X$, there exists a $k \in \omega$ such that A avoids $h_k^{\rm LC}$.*

A set X is Δ-comeager *if $2^\omega - X$ is Δ-meager.*

Note that if $\Delta = \omega^\omega$, then Definition 2.7 coincides with Definition 2.3. Also, if $\Delta = \text{rec}$, then Definition 2.7 coincides with earlier definitions of rec-category [Lut87, Lut90, Fen91]. We now define resource-bounded genericity for countable classes Δ.

Definition 2.8 *Let Δ be a countable function class. A set $G \in 2^\omega$ is* Δ-generic *if G is an element of every Δ-comeager set (equivalently, the singleton $\{G\}$ is not Δ-meager).*

The arith-generic sets are exactly what are typically called 'generic sets' (see [Joc80]). The rec-generic sets turn out to be exactly the weakly 1-generic sets defined by Kurtz in [Kur83] (see [Fen91]). All 1-generic sets [Joc80] are weakly 1-generic, but not vice versa.

3 Basic Properties of Δ-Meager Sets

In this section we present the properties of Δ-meager sets which make them analogous to those of classical meager sets mentioned in the introduction (Theorems 3.3 and 3.4). The point of this section is that our new notion of category retains all the desirable properties of Lutz's original definition [Lut87, Lut90]. Therefore, the results in this section generally follow Lutz with some occasional alterations.

For any function class Δ, we first define a corresponding language class $R(\Delta)$, an arena in which the Δ-bounded version of the Baire category theorem holds. We next show that our definition of Δ-category can be characterized entirely in terms of resource-bounded winning strategies in a Banach-Mazur game. Finally, we discuss Δ-generic sets, where we will find it useful to place restrictions on our definition of category. In doing so, we introduce a wide, unified spectrum of different notions of resource-bounded category, where Lutz's notion is at one end, and ours is at the other.

3.1 Resource-Bounded Baire Category Theorem

A *constructor* is a function $\delta \in \Delta$ such that for all binary strings σ, $\delta(\sigma) = \sigma\hat{}b$, for some $b \in \{0,1\}$. Given a binary string ι, define $R_\iota(\delta)$ to be the unique element of 2^ω extending $\delta^i(\iota)$ for all $i \in \omega$. Define $R(\delta)$ to be $R_\lambda(\delta)$, and let

$$R(\Delta) \stackrel{\mathrm{df}}{=} \{R(\delta) \mid \delta \in \Delta \text{ is a constructor}\}.$$

Notice that for all ι, $\iota \prec R_\iota(\delta)$ and that

$$\{R_\iota(\delta) \mid \delta \in \Delta \text{ is a constructor}\} = R(\delta) \cap C_\iota.$$

Lemma 3.1 (Lutz [Lut87, Lut92])

1. $R(\omega^\omega) = 2^\omega$.
2. $R(\text{arith}) = \{A \subseteq \omega \mid A \textit{ is arithmetic}\}$.
3. $R(\text{rec}) = \{A \subseteq \omega \mid A \textit{ is recursive}\}$.
4. $R(\mathbf{PF}) = \mathbf{E} = \mathbf{DTIME}(2^{\text{linear}})$.

Definition 3.2 ([Lut87, Lut92])

- *A subset $X \subseteq R(\Delta)$ is* Δ-countable *if there is a function $\delta \in \Delta$ such that δ_k is a constructor for each $k \in \omega$ and $X \subseteq \{R(\delta_k) \mid k \in \omega\}$.*
- *A Δ-union of Δ-meager sets is a subset $X \subseteq 2^\omega$ such that $X = \bigcup_{i\in\omega} X_i$ and there is an $h \in \Delta$ such that h_i witnesses the Δ-meagerness of X_i for each i.*

Part 3 of the next theorem is an adaptation of Lemma 3.11 in [Lut90] to locally computable strategies.

Theorem 3.3 *Let Δ be a function class closed under pftt-reductions.*

1. *All Δ-countable sets are Δ-meager.*
2. *All subsets of a Δ-meager set are Δ-meager.*
3. *The Δ-union of Δ-meager sets is Δ-meager.*

Proof Sketch:

1. Let $X \subseteq R(\Delta)$ be Δ-countable as witnessed by the function $\delta \in \Delta$ (so δ_i is a constructor for all i). Let h be a function such that for all $i \in \omega$ and $\sigma \in 2^{<\omega}$,
$$h_i^{\mathrm{LC}}(\sigma) = \sigma\hat{\ }[1 - [\delta_i(\sigma)](|\sigma|)] .$$
It is easy to see that $h \leq_{tt}^{\mathrm{pf}} \delta$, and so it follows that $h \in \Delta$. Also, $R(\delta_i)$ avoids h_i^{LC} for all $i \in \omega$. Thus X is Δ-meager as witnessed by h.

2. Immediate by the definition.

3. Let $X = \bigcup_{i\in\omega} X_i$ and $h \in \Delta$ such that X_i is Δ-meager witnessed by h_i for all $i \in \omega$. Define g so that
$$g_{\langle i,j\rangle}(x) = h_i(\langle 0^j, x\rangle)$$
for all $i, j, x \in \omega$. Clearly, $g \in \Delta$, and g_k^{LC} is a strategy for all k. Given $A \in X$, choose i_0 such that $A \in X_{i_0}$. Then X_{i_0} is Δ-meager via h_{i_0}, so there is a j_0 such that A avoids $h_{i_0 j_0}^{\mathrm{LC}} = g_{\langle i_0,j_0\rangle}^{\mathrm{LC}}$. Thus g witnesses that X is Δ-meager.

□

We now present the resource-bounded version of the Baire category theorem. Theorem 3.4 again adapts Theorem 3.12 in [Lut90] to locally computable strategies. Theorem 3.4 is particularly important, saying that even strategies locally computable in Δ are still not powerful enough to make $R(\Delta)$ Δ-meager. Since locally computable strategies can be much more powerful than ones that must be globally computed as in previous settings [Lut87, Fen91], Theorem 3.4 is a considerable strengthening of Theorem 3.12 in [Lut87] and of Theorem 3.4 in [Fen91].

Theorem 3.4 *Let Δ be closed under pftt-reductions. If $X \subseteq 2^\omega$ is Δ-meager, then $R(\Delta) - X$ is dense.*

Proof: Given $h \in \Delta$ witnessing that X is Δ-meager, we describe a constructor $\delta \in \Delta$ such that for any initial binary string ι, $R_\iota(\delta) \notin X$. For this, it suffices to build δ so that $R_\iota(\delta)$ meets all of $h_0^{\mathrm{LC}}, h_1^{\mathrm{LC}}, h_2^{\mathrm{LC}}, \ldots$, that is, for every $n \in \omega$ there exists a binary string σ such that $h_n^{\mathrm{LC}}(\sigma) \prec R_\iota(\delta)$. The constructor δ is computed by the following algorithm, which uses delayed diagonalization:

1. Given as input a binary string σ of length ℓ, compute
$$n_0 \leftarrow \min_n \left[\begin{array}{l} n \leq \ell \ \& \\ (\forall \alpha \preceq \sigma)\ h_n^{\mathrm{LC}}(\alpha) \not\preceq \sigma \end{array} \right].$$
($h_{n_0}^{\mathrm{LC}}$ is the next strategy that $R(\delta)$ must meet.)

2. If no such n_0 exists, return $\sigma\hat{\ }0$.

3. Otherwise, find the shortest prefix $\tau \preceq \sigma$ such that $\sigma \prec h_{n_0}^{\mathrm{LC}}(\tau)$. (Since $\sigma \prec h_{n_0}^{\mathrm{LC}}(\sigma)$, τ must exist.)

4. Return $\sigma\hat{\ }b$ where $b \stackrel{\mathrm{df}}{=} h_{n_0}(\tau, 0^\ell)$. (We have $b \in \{0,1\}$ since $\sigma \prec h_{n_0}^{\mathrm{LC}}(\tau)$.)

The preceding algorithm can be formulated as a pftt-reduction to h. The important thing to note is that, although the strings $h_n^{\mathrm{LC}}(\alpha)$ may be quite long in their entirety, one only needs to compute them locally on the first $\ell + 1$ arguments. Thus the algorithm only needs a table of values for $h_n(\alpha, 0^j)$ for all $n, j \leq \ell$ and all $\alpha \preceq \sigma$, and runs in polynomial time given such a table. Therefore $\delta \leq_{tt}^{\mathrm{pf}} h$, and hence $\delta \in \Delta$. A straightforward induction argument shows that $R_\iota(\delta)$ meets all the h_n^{LC}. □

Lemma 3.5 (Lutz [Lut87]) *If X is Δ-meager, and $\Delta \subseteq \Delta'$, then X is Δ'-meager.*

Proof: Immediate by Definition 2.7. □

3.2 Resource-Bounded Banach-Mazur Games

Meager sets can be alternatively defined in terms of winning strategies of Banach-Mazur games (see [Oxt80]). An effective version of the Banach-Mazur game was introduced by Lisagor in 1979 (see [Lut90]). Lutz [Lut87, Lut90] showed that if Δ is one of the space-bounded classes, then Δ-meager sets can likewise be characterized in terms of Banach-Mazur games where the second player must follow a strategy computable in Δ. The equivalence was subsequently extended to the time-bounded classes [Fen91]. Here, we prove that our current notion of resource-bounded category also admits a Banach-Mazur game characterization for all our classes Δ. Thus resource-bounded category and resource-bounded Banach-Mazur games are equivalent for all of the classes (and notions) we have encountered. Banach-Mazur games are extremely useful in studying resource-bounded category because they allow us to work with single strategies rather than uniform families of them.

The following definitions are adapted from [Lut87] and [Lut90].

Definition 3.6 ([Lut90])

- *A* play *of a Banach-Mazur game is an ordered pair (f, g) of finite extension strategies such that $\sigma \prec g(\sigma)$ for every $\sigma \in 2^{<\omega}$.*
- *The* result *$R(f, g)$ of the play (f, g) is the unique element of 2^ω that extends $(g \circ f)^i(\lambda)$ for all $i \in \omega$.*

If $X \subseteq 2^\omega$ and $\Delta_{\mathrm{I}}, \Delta_{\mathrm{II}}$ are function classes, then $G[X; \Delta_{\mathrm{I}}, \Delta_{\mathrm{II}}]$ is the Banach-Mazur game with distinguished set X, where Player I must choose a finite extension strategy of the form f^{LC} for some $f \in \Delta_{\mathrm{I}}$, and Player II must choose a finite extension strategy of the form g^{LC} for some $g \in \Delta_{\mathrm{II}}$. Player I *wins* the play $(f^{\mathrm{LC}}, g^{\mathrm{LC}})$ if $R(f^{\mathrm{LC}}, g^{\mathrm{LC}}) \in X$; Player II wins otherwise. A *winning strategy* for Player II is a strategy g^{LC} with $g \in \Delta_{\mathrm{II}}$ such that Player II wins $(f^{\mathrm{LC}}, g^{\mathrm{LC}})$ for every $f \in \Delta_{\mathrm{I}}$ such that f^{LC} is a strategy. For more intuition about resource-bounded Banach-Mazur games, see [Fen91] for example.

As in [Fen91], we prove a result stronger than Theorem 4.3 of [Lut90] (also Theorem 2 of [Lut87]) by allowing our requirements to be satisfied slowly.

Theorem 3.7 *Let Δ be closed under pftt-reductions, and let X be a subset of 2^ω. The following are equivalent:*

1. *Player II has a winning strategy for $G[X; \omega^\omega, \Delta]$.*
2. *X is Δ-meager.*

Proof: ($1 \Longrightarrow 2$): The classical form of this implication (where $\Delta = \omega^\omega$) was first proved by Banach in a different setting (see [Oxt80, pages 27–30]). The proof here uses the essential ideas of Theorem 2 in [Lut87], adapted to handle locally computed strategies. Assume $g \in \Delta$ and g^{LC} is a winning strategy for Player II. We define h as follows: for all $k \in \omega$ and all $\sigma \in 2^{<\omega}$ of length ℓ, let

$$h_k(\sigma, 0^n) \stackrel{\mathrm{df}}{=} g(\sigma', 0^n)$$

for all $n \in \omega$, where $\sigma' \stackrel{\mathrm{df}}{=} \sigma \hat{\ } 0^{k \dot{-} \ell}$. It follows immediately that

$$h_k^{\mathrm{LC}}(\sigma) = g^{\mathrm{LC}}(\sigma').$$

Clearly, $h \leq_{tt}^{\mathrm{pf}} g$ so $h \in \Delta$. Suppose $A \in 2^\omega$ meets h_k^{LC} for all k. We show that $A \notin X$, which implies that X is Δ-meager as witnessed by h. To do this, it suffices to show that for every $\alpha \prec A$ there is a β such that

$$\alpha \preceq \beta \preceq g^{\mathrm{LC}}(\beta) \prec A.$$

For if this is the case, then Player I has a strategy f^{LC} to ensure that $R(f^{\mathrm{LC}}, g^{\mathrm{LC}}) = A$: given α, Player I simply extends to a corresponding β, which always forces Player II to extend to a prefix of A. If $A \in X$, then g^{LC} would not be a winning strategy for Player II, so it follows that $A \notin X$.

Given $\alpha \prec A$, we find β as follows: Let $k \stackrel{\mathrm{df}}{=} |\alpha|$. We know that A meets h_k, which implies that there is a $\sigma \prec A$ of length ℓ such that

$$\sigma' \preceq g^{\mathrm{LC}}(\sigma') = h_k^{\mathrm{LC}}(\sigma) \prec A,$$

where again we set $\sigma' \stackrel{\mathrm{df}}{=} \sigma\hat{\ }0^{k \dot{-} \ell}$. Since $|\alpha| = k \leq |\sigma'|$, and both α and σ' are extended by A, it must be that $\alpha \preceq \sigma'$. We can then take β to be σ'.

($2 \Longrightarrow 1$): Suppose X is Δ-meager as witnessed by $h \in \Delta$, that is, for every $A \in X$ there exists $i \in \omega$ such that A avoids h_i^{LC}. We define a $g \in \Delta$ such that g^{LC} is a winning strategy for Player II in the game $G[X; \omega^\omega, \Delta]$. For this, it suffices to ensure that for any strategy f^{LC}, $R(f^{\mathrm{LC}}, g^{\mathrm{LC}})$ meets all the h_i^{LC}, and hence $R(f^{\mathrm{LC}}, g^{\mathrm{LC}}) \notin X$. It is no problem to force $R(f^{\mathrm{LC}}, g^{\mathrm{LC}})$ to meet any given h_i^{LC} just by letting g simulate h_i at some turn of the play. The only problem is knowing which h_i to simulate at each turn, in order to simulate all of them eventually.

The function g is computed in a fashion similar to the the constructor δ in the proof of Theorem 3.4. Given an input $\sigma \in 2^{<\omega}$ and $j \in \omega$ we define $g(\sigma, 0^j)$ as follows:

1. Compute
$$n_0 \leftarrow \min_n \left[\begin{array}{l} n < |\sigma| \ \ \& \\ (\forall \tau \preceq \sigma)[h_n^{\mathrm{LC}}(\tau) \not\preceq \sigma] \end{array} \right].$$
($h_{n_0}^{\mathrm{LC}}$ is the next strategy we must simulate.)
2. If no such n_0 exists, return $\sigma(j)$ if $j < |\sigma|$; 0 if $j = |\sigma|$; and $\perp$ otherwise.
3. If n_0 exists, return $h_{n_0}(\sigma\hat{\ }0, 0^j)$.

The function g is in Δ for similar reasons as in Theorem 3.4. A simple induction shows that if τ is the state of the game after Player II's ith move, then τ meets the strategies $h_0^{\mathrm{LC}}, \ldots, h_{i-1}^{\mathrm{LC}}$ regardless of Player I's strategy f^{LC}. Thus $R(f^{\mathrm{LC}}, g^{\mathrm{LC}})$ meets all the h_i^{LC}. □

4 Results about PF-category

Our main theorem of this section, Theorem 4.1, shows that our definition of unrestricted **PF**-category is exceedingly strong. The categorical "size" of many classes can be characterized trivially. This motivates the study of length-bounded **PF**-category, and we also present facts about various length bounds.

Theorem 4.1 *Suppose $X \subseteq 2^\omega$ is a Σ_2^0 class. If there is a set $A \in \mathbf{E}$ such that $A' \notin X$ for all A' with $A' \triangle A$ finite, then X is* **PF**-*meager.*

Proof: Let X be a Σ_2^0 class, and let $A \in \mathbf{E}$ be as in the statement of the theorem. There is an oracle machine M that is total for all oracles, such that

$$X = \{B \mid (\exists x)(\forall y) M^B(x, y) = 0\}.$$

By standard tricks, we can assume without loss of generality that M runs in polynomial time. In the Banach-Mazur game on X, Player II extends its input by computing A locally until it "notices" that

$$(\forall x \leq n)(\exists y) M^\tau(x, y) \neq 0,$$

where n is the length of Player II's input, and τ is Player II's extension so far. Player II can run in polynomial time, and will always eventually stop extending. More formally, the function $f \in \mathbf{PF}$ is computed for all σ and $n \geq |\sigma|$ as follows:

1. Compute the unique $\tau \succeq \sigma$ of length n such that $\tau(z) = A(z)$ for all $z \in \text{domain}(\tau) - \text{domain}(\sigma)$.

2. Compute $M^\tau(x, y)$ for all $x < \log|\sigma|$ and $y < \log n$. Since x and y are so small, we may assume that all of M's queries are in the domain of τ. (This is the only reason why we use log's.)

3. If $(\forall x < \log|\sigma|)(\exists y < \log n) M^\tau(x, y) \neq 0$, then output $\perp$. Otherwise, output $A(n)$.

It is easy to show that f^{LC} is a strategy, for if f^{LC} extends σ infinitely, then the result is a finite variant of A, and hence not in X. Therefore, there is a prefix τ of f^{LC}'s extension such that $(\forall x < \log|\sigma|)(\exists y < \log|\tau|) M^\tau(x, y) \neq 0$, in which case f^{LC} should not have extended past τ—contradiction. Now let B be the result of any game where Player II uses f^{LC}. If x is arbitrary, then on the first turn for Player II where the state of the game is of length at least $2^x + 1$, Player II will extend to ensure that there is a y such that $M^B(x, y) \neq 0$. This guarantees that $B \notin X$. □

Corollary 4.2 *If X is a Σ_2^0 class and closed under finite variations, then*

$$X \text{ is } \mathbf{PF}\text{-meager} \iff \mathbf{E} \not\subseteq X.$$

Note that all recursively presentable classes are Σ_2^0.

Corollary 4.3 *If* $\mathbf{NP} \neq \mathbf{EXP}$, *then* $\mathbf{NP}$ *is* $\mathbf{PF}$*-meager.*

The upper (lower) r-cone of a set A is the class of all languages to which A reduces (reducible to A) via the reducibility r.

Corollary 4.4 *Let A be any set. The lower $\leq_{\text{T}}^p$-cone of A is $\mathbf{PF}$-meager if and only if A is not $\leq_{\text{T}}^p$-hard for $\mathbf{E}$. The upper $\leq_{\text{T}}^p$-cone of A is $\mathbf{PF}$-meager if and only if $A \notin \mathbf{P}$. The same holds for $\leq_m^p$-cones.*

Corollary 4.5 *The complete $\leq_{\text{T}}^p$-degree for $\mathbf{EXP}$ is $\mathbf{PF}$-meager.*

4.1 PF-Generic Sets

Lemma 4.6 *Let Δ be a countable class closed under pftt-reductions. A set $G \in 2^\omega$ is Δ-generic if and only if G meets every strategy h^{LC} for $h \in \Delta$.*

Proof: Suppose $h \in \Delta$ and h^{LC} is a strategy avoided by G. Then letting $f(\langle x, y\rangle) \stackrel{\text{df}}{=} h(y)$, we see that $f \in \Delta$ and G avoids f_i^{LC} for all $i \in \omega$. Thus, $\{G\}$ is Δ-meager as witnessed by f, so G is not Δ-generic. Conversely, if $\{G\}$ is Δ-meager as witnessed by $f \in \Delta$, then by definition there is an i such that G avoids the strategy f_i^{LC}, and furthermore, $f_i \in \Delta$. □

We may take Lemma 4.6 as an alternative definition of Δ-genericity. It was shown in [Fen91], using the older definition of globally computed strategies, that resource-bounded generic sets exist in subrecursive complexity classes. More precisely, it was shown there that if a function class Δ' contains a universal function for Δ, then Δ-generic sets exist in $R(\Delta')$. This is not the case under the present definitions, however, owing to the enormous power of locally computed strategies and the fact that they are not enumerable in any uniform recursive way. In fact, we have the following:

Proposition 4.7 *If G is $\mathbf{PF}$-generic, then G is immune (hence not recursively enumerable).*

In the next section, we weaken and "uniformize" our notions of Δ-genericity so that the statement above regarding Δ-generic sets existing in $R(\Delta')$ will again be true.

Proof of Proposition 4.7: Suppose G is **PF**-generic. G must be an infinite set, since it meets every strategy s_k of the form, "given σ, extend with k ones." Suppose there is an infinite r.e. subset $A \subseteq G$. Then there is a function $f \in \mathbf{PF}$ with range$(f) = A$. We use f to define a strategy that G avoids. Intuitively, given σ we build an extension τ by continually extending σ with zeros until we notice that $A(n) = 1$ for some n such that $\tau(n) = 0$. At this point we know that $\tau \not\prec G$, since $G(n) = 1$. More formally, define

$$h(\sigma, 0^n) \stackrel{\mathrm{df}}{=} \begin{cases} \sigma(n) & \text{if } n < |\sigma|, \text{ else} \\ 0 & \text{if } \{f(0), \ldots, f(n)\} \cap \\ & \quad \{|\sigma|, |\sigma|+1, \ldots, n-1\} = \emptyset, \\ \perp & \text{otherwise.} \end{cases}$$

Clearly, $h \in \mathbf{PF}$. Moreover, h^{LC} defines a strategy: since G is infinite, there must exist a least n_0 for which an element appears in $\{f(0), \ldots, f(n_0)\} \cap \{|\sigma|, \ldots, n_0 - 1\}$, and since this set can only increase in size with n, we have $h(\sigma, 0^n) = \perp$ for all $n \geq n_0$. Finally, $h^{\mathrm{LC}}(\sigma) \not\prec G$ for all σ, so G avoids h^{LC}, contradicting the fact that G is **PF**-generic. $\square$

Remark: Since $\omega - G$ is also **PF**-generic, G is bi-immune and thus cannot be co-r.e., either. Note also that these results hold for G Δ-generic for any class Δ that we are considering. On the other hand, for all $\Delta \subseteq$ rec, since all Δ-generic sets are weakly 1-generic (equivalently, rec-generic), there are Δ-generic sets recursive in any nonrecursive r.e. set (a result due to R. A. Shore; see [Kur83]).

Actually, an easy modification of the above proof shows that G must be hyperimmune. This fact has curious implications. Kurtz [Kur83] showed that the rec-generic (unbounded Turing) degrees are exactly the hyperimmune degrees. It follows that the degrees of rec-generic sets and **PF**-generic sets are the same. Indeed, for $\Delta \subseteq$ rec, the degrees of Δ-generic sets are independent of Δ; they are exactly the hyperimmune degrees. This shows, among other things, that Kurtz's argument that all rec-generic sets are hyperimmune does not require difficult local computations. Finally, although the degrees are always the same, the actual classes of Δ-generic sets do in general depend upon the choice of Δ. If Δ' contains a universal function for Δ, then one can construct a Δ-generic set that is not Δ'-generic. This construction *does* require difficult local computations.

Our final result of this section draws a connection between measure and category—actually rec-measure/randomness and **PF**-category/genericity. It's proof is basically an effective version of the classical theorem stating that for any continuous $f\colon 2^\omega \to 2^\omega$, there is a comeager subset of 2^ω whose image under f has measure 0. Note that the reduction in the theorem is any (recursive) truth-table reduction.

Theorem 4.8 *Let f be any tt-reduction. There exists a* **PF***-comeager set X such that*

$$\mu_{\mathrm{rec}}(\{B \mid (\exists A \in X) B \leq_{tt} A \text{ via } f\} = 0.$$

Corollary 4.9 *If R is* rec*-random and G is* **PF***-generic, then $R \not\leq_{tt} G$.*

Proof of Theorem 4.8: For definitions and discussion of resource-bounded measure via martingales, we refer the reader to Lutz [Lut92]. Fix a tt-reduction f. For any A, let $f(A)$ be the unique set tt-reducible to A via f. We will describe a finite-extension strategy g for Player II in a Banach-Mazur game, and a recursive Martingale d such that g ensures that d always succeeds on $f(R)$, where R is the result of any play of the game. The strategy g will be locally computable in **PF**.

Terminology: if σ and τ are strings, say that τ *commits* σ if, for all $x \in$ domain(σ), $\sigma(x)$ is the value computed by f from some sequence extending τ, and the use of f's computation is entirely within domain(τ). That is, τ is enough of the oracle to compute σ via f. We say that A *commits* σ if σ is a prefix of $f(A)$. The *weight* of a string τ is the position of the rightmost 1 in τ (reading from left to right, the leftmost bit being at position 1). If τ is all zeros, then τ has weight 0. If an infinite sequence B has finitely many ones, then its weight is defined similarly.

Given a string τ, $g(\tau)$ will be equal to τ extended with a large number of zeros, the exact number to be determined later. The idea of the construction of g and d is that d is looking to "bet" on a string committed by a finite set (i.e., an infinite sequence with finitely many ones). The job of g is to throw in enough zeros into the sequence so that it looks finite "enough" so that d's bets pay off.

Set $d(\lambda) = 1$. For any string σ, the value of $d(\sigma\hat{\ }0)$ will always be either $\frac{1}{2}d(\sigma)$, $d(\sigma)$, or $\frac{3}{2}d(\sigma)$, and so the same goes for $d(\sigma\hat{\ }1)$. The martingale $d(\sigma)$ is looking to bet on the extension of σ committed by a string τ with lowest weight.

For $\sigma \in 2^{<\omega}$ and $b \in \Sigma$, the (recursive) algorithm for $d(\sigma\hat{\ }b)$ is as follows:

1. See if there is a string τ committing $\sigma\hat{\ }0$ with weight not exceeding $|\sigma|$. If there is, find one of lowest possible weight w_0. If not, set $w_0 = |\sigma| + 1$.

2. See if there is a string τ committing $\sigma\hat{\ }1$ with weight not exceeding $|\sigma|$. If there is, find one of lowest possible weight w_1. If not, set $w_1 = |\sigma| + 1$.

3. If $w_0 = w_1$, then set $d(\sigma\hat{\ }b) = d(s)$ (no bet). Else, if $w_0 < w_1$ then *bet* on $\sigma\hat{\ }0$ by setting $d(\sigma\hat{\ }b) = (\frac{3}{2} - b)d(\sigma)$. Else, bet on $\sigma\hat{\ }1$ by setting $d(\sigma\hat{\ }b) = (\frac{1}{2} + b)d(\sigma)$.

The strategy g is locally computed by a function $h \in \mathbf{PF}$. On input τ and 0^n with $n \geq |\tau|$, h runs for n steps. It starts computing $d(v_0), d(v_1), d(v_2), \ldots$, where $v_0, v_1, v_2, \ldots$ are the longest strings committed by $f(\tau), f(\tau\hat{\ }0), f(\tau\hat{\ }00), \ldots$, respectively. If, within n steps, an i is found such that $d(v_i) > |\tau|$, then h outputs $\perp$; otherwise, h outputs 0.

Intuitively, τ represents the state of a Banach-Mazur game just after Player I has moved. The strategy $g(\tau)$ extends with zeros, stopping when it finds an i such that $d(v_i) > |\tau|$. The extension is thus $\tau\hat{\ }0^t$ for some t.

Clearly, if $g(\tau)$ always extends finitely, then it is a valid strategy, and d always succeeds on the sequence produced by the game. Thus all that remains to be shown is that $g(\tau)$ extends finitely. Let $A \stackrel{\text{df}}{=} \tau\hat{\ }0^\infty$ and let B be the infinite sequence extending all the v_n, i.e., $B = f(A)$.

Claim 1 *There is a finite prefix $v \prec B$ such that, for any prefix $\beta \prec B$ extending v, $d(\beta)$ bets on a prefix of B.*

[The finiteness of $g(\tau)$ follows from the claim: since v is extended by some v_j, d will bet correctly from v_j onwards and its assets will increase geometrically, so $g(\tau)$ "waits" until it sees that d's assets exceed $|\tau|$ on some prefix $v_k \prec B$, then it extends further with additional zeros, if needed, to commit v_k.]

We now prove the claim. Let w be the weight of A (same as the weight of τ). Let v initially be some arbitrary prefix of B. The martingale $d(v)$ bets on at most one of two possible extensions of v. The only way that $d(v)$ would not bet on the prefix of B would be if the other extension were committed by some string η with weight $\leq w$. In this case, we update v by properly extending it to be any longer prefix of B. If $d(v)$ still fails to bet on a prefix of B, then this failure is witnessed by *another* string of weight not exceeding w, *incompatible* with η. If this happens, we extend v further, and so on. The number of times that $d(v)$ can fail to bet correctly in this way is therefore bounded by the maximum number of pairwise incompatible strings of weight $\leq w$. Since this number is finite, there is a v beyond which d always bets on a prefix of B.

This ends the proof of Theorem 4.8. □

Remark: There is an alternate way of defining resource-bounded genericity that does not use our notion of resource-bounded category. This type of genericity, defined and studied in [ASFH87, ASNT94], appears to have properties markedly different from our notion above. In particular, it can be shown [Fen95] that there are rec-generic sets (even $aw2$-generic sets [Fen94]) that are not n^c-generic, for any $c \geq 1$, and vice versa.

Other researchers have also investigated resource-bounded category and genericity in slightly different settings [Böt94, Fos90, Fos93, Yam93].

5 Length-Bounded Δ-Category

We can bring resource-bounded generic sets back down into subrecursive classes by taking a somewhat weaker definition of category. In the definitions below, when we say that a function f is time-constructible we mean that $f(x)$ can be computed in time polynomial in $f(x)$. It is well-known that every recursive function is majorized by a time-constructible function. We will always assume below that $(\forall n) f(n) \geq n$ for such functions f.

Definition 5.1 *Let f be a time-constructible function. A strategy s is* f-bounded *if*

$$|s(\sigma)| \leq f(|\sigma|)$$

for all but finitely many $\sigma \in 2^{<\omega}$.

Definition 5.2 *Let f be any time-constructible function. A set $X \subseteq 2^\omega$ is* $\Delta/_f$-meager *if X is Δ-meager witnessed by a function $h \in \Delta$ such that h_i^{LC} is f-bounded for all $i \in \omega$. A set $G \in 2^\omega$ is* $\Delta/_f$-generic *if $\{G\}$ is not $\Delta/_f$-meager.*

Definition 5.2 gives us a uniform scheme for defining an infinite variety of interesting and different notions of resource-bounded category—one for each choice of f. In particular, Lutz's original definition fits quite nicely into this scheme. The proofs of Theorems 3.3, 3.4, and 3.7 can be modified easily to yield similar results about $\Delta/_f$-meager and $\Delta/_f$-generic sets.

The following proposition, whose proof we omit, asserts the robustness of $\Delta/_f$-category by showing that it can be defined in different ways.

Proposition 5.3 *Let f be a time-constructible function. The following are equivalent:*

1. *X is $\Delta/_f$-meager.*
2. *X is Δ-meager witnessed by an $h \in \Delta$ such that $|h_i^{\mathrm{LC}}(\sigma)| \leq f(|\sigma|)$ for all but finitely many pairs $\langle i, \sigma \rangle$.*
3. *Player II has an f-bounded winning strategy in the game $G[X; \omega^\omega, \Delta]$.*
4. *Player II has a winning strategy s in the game $G[X; \omega^\omega, \Delta]$, such that $|s(\sigma)| \leq f(|\sigma|)$ for all $\sigma \in 2^{<\omega}$.*

It should be noted that if $\Delta \subseteq$ rec, then X is Δ-meager if and only if X is $\Delta/_f$-meager for *some* time-constructible f. Thus we lose nothing if we forget about Δ-category altogether and only study $\Delta/_f$-category. The length bounds we impose do make a difference, however, regarding Δ-genericity. They give us resource-bounded generic sets in subrecursive complexity classes, in contrast to Proposition 4.7. One can show, for instance, $\mathbf{PF}/_f$-generics exist in **EXP** for *any* time-constructible f. See Theorem 6.2 of [Lut92] for a similar statement about '**PF**-random' sets.

It is easily seen that $\mathbf{PF}/_{\mathrm{poly}}$-category (i.e., polynomial-length extensions) is identical to Lutz's original definition of **PF**-category [Lut87, Lut90].

The table of Figure 5 describes some known facts regarding length-bounded **PF**-category, for various length bounds f (the columns). For each property, "meager" or "generic" always means $\mathbf{PF}/_f$-meager or $\mathbf{PF}/_f$-generic, for the various f. A set A is *weakly complete* for **E** if $A \in \mathbf{E}$ and its lower $\leq_m^p$-cone under is not **PF**-meager (or does not have **PF**-measure 0, depending on the context). Lutz [Lut94] constructed an incomplete, weakly complete set for **E** in the context of **PF**-measure. Ambos-Spies, Terwijn, and Zheng [ASTZ] improved Lutz's result by showing that the weakly complete sets have measure 1 in **E**. Joseph, Pruim, and Young [JPY94] constructed an incomplete, weakly complete set for **E** in the context of $\mathbf{PF}/_{\mathrm{poly}}$-category. Inversely, Corollary 4.4 above says that there are no incomplete, weakly complete sets for **E** in the context of (unbounded) **PF**-category.

5.1 Games with One-Bit Strategies

There is another game characterization of the $\Delta/_f$-meager sets, due to Pruim [Pru95], in terms of what we call the *f-bounded 1-bit game*. The characterization holds in the case where Δ has the following additional closure

f	poly	$2^{\text{poly log}}$	$2^{2^{\text{poly log log}}}$	unbounded
EXP-complete $\leq_T^p$-degree is meager	no	no	yes	yes
$\mathbf{NP} \neq \mathbf{EXP} \Longrightarrow$ (**NP** is meager)	? (oracle dep.)	? (oracle dep.)	? (oracle dep.)	yes
generics in **EXP**	yes	yes	yes	no
incomplete, weakly complete sets for **E**	yes [JPY94]	?	?	no
E-weakly-complete is comeager in **E**	?	?	?	no
$\{A \mid \mathbf{P}^A = \mathbf{NP}^A\}$ is meager	no [Fen91]	no	yes	yes

Figure 1: Some known facts about $\mathbf{PF}/_f$-category.

property: for every characteristic (0/1-valued) function $f \in \Delta$, the function $f \circ g$ is also in Δ, where g is defined inductively as follows:

$$\begin{aligned} g(x, \lambda) &= x \\ g(x, 0^{i+1}) &= g(x, 0^i)\hat{\ } f(g(x, 0^i)). \end{aligned}$$

Notice that all of the Δ we are considering are "closed under iterations" in this sense.

In the f-bounded one-bit game $G_f[X; \omega^\omega, \Delta]$, each player's turn consists of a sequence of consecutive moves. Each move is determined by following a strategy as before, but since we are allowed multiple moves per turn, we may restrict each move to extending the input by one bit. Thus, strategies for one-bit games are strategies which are only allowed to extend their input by one bit. Player I is allowed any strategy and may make any number of moves per turn, as long as he eventually ends his turn. Player II must use a strategy from Δ and always makes exactly $f(n)$ moves per turn, where n is the length of the string describing the game at the end of Player I's previous turn.

There are two main differences between the games $G_f[X; \omega^\omega, \Delta]$ and $G[X; \omega^\omega, \Delta/_f]$. In the game $G_f[X; \omega^\omega, \Delta]$, Player II has easy access to all of the previous bits played during the current turn, but does not know when the turn began. In $G[X; \omega^\omega, \Delta/_f]$, it is the other way around; Player II now knows when the turn began, but must calculate the ith bit of his move based solely on the status of the game at the start of the turn. Pruim points out, however, that the apparent disadvantages of each game over the other are easily overcome. Player II in $G_f[X; \omega^\omega, \Delta]$ can look back at the portion of the game played so far (including the moves already made on the current turn) and find an appropriate portion on which to imitate the strategy from the other game, and do this in such a way that neighboring bits are decided in a consistent manner. Conversely, if Δ is closed under iterations, then Player II in $G[X; \omega^\omega, \Delta/_f]$ can simply simulate i moves of Player II in $G_f[X; \omega^\omega, \Delta]$ and output the appropriate bit.

6 Open Problems

We would like the resolve the remaining question marks in Figure 5. We would especially like to see in which contexts the results about weakly complete sets break down.

Acknowledgments

I would like to thank Jack Lutz, Elvira Mayordomo, Randy Pruim, and Klaus Ambos-Spies for several helpful discussions.

References

[ASFH87] K. Ambos-Spies, H. Fleischhack, and H. Huwig. Diagonalizations over polynomial time computable sets. *Theoretical Computer Science*, 51:177–204, 1987.

[ASNT94] K. Ambos-Spies, H.-C. Neis, and S. A. Terwijn. Genericity and measure for exponential time. In *Proceedings of MFCS '94*, Lecture Notes in Computer Science, pages 221–232. Springer-Verlag, 1994. Extended abstract.

[ASTZ] K. Ambos-Spies, S. A. Terwijn, and X. Zheng. Resource-bounded randomness and weakly complete problems. Manuscript.

[Böt94] M. Böttcher. The lookahead function as a parameter for resource-bounded generics. Manuscript, 1994.

[Fen91] S. A. Fenner. Notions of resource-bounded category and genericity. In *Proceedings of the 6th Annual IEEE Structure in Complexity Theory Conference*, pages 196–212, 1991. Journal version in preparation.

[Fen94] S. Fenner. Almost weakly 2-generic sets. *Journal of Symbolic Logic*, 59(3):868–887, September 1994. An earlier version appeared in Proceedings of the Sixth Annual IEEE Structure in Complexity Theory Conference.

[Fen95] S. Fenner, 1995. Unpublished.

[Fos90] J. A. Foster. *Forcing and Genericity on the Polynomial Hierarchy.* PhD thesis, Illinois Institute of Technology, 1990.

[Fos93] J. A. Foster, 1993. Private communication.

[Joc80] C. G. Jockusch. Degrees of generic sets. In F. R. Drake and S. S. Wainer, editors, *Recursion Theory: Its Generalizations and Applications*, pages 110–139. Cambridge University Press, 1980.

[JPY94] D. Joseph, R. Pruim, and P. Young. Weakly hard problems in resource-bounded category. Manuscript, August 1994.

[Kur83] S. A. Kurtz. Notions of weak genericity. *Journal of Symbolic Logic*, 48(3):764–770, September 1983.

[Lut87] J. H. Lutz. Resource-bounded baire category and small circuits in exponential space. In *Proceedings of the 2nd Annual IEEE Structure in Complexity Theory Conference*, pages 81–91, 1987.

[Lut90] J. H. Lutz. Category and measure in complexity classes. *SIAM J. Comp.*, 19(6):1100–1131, December 1990.

[Lut92] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44:220–258, 1992. An earlier version appeared in Proceedings of the Fourth Annual IEEE Structure in Complexity Theory Conference.

[Lut94] J. H. Lutz. Weakly hard problems. In *Proceedings of the 9th IEEE Structure in Complexity Theory Conference*, pages 146–161, 1994.

[Oxt80] J. C. Oxtoby. *Measure and Category.* Springer-Verlag, 1980.

[Pru95] R. Pruim, 1995. Private communication.

[Yam93] T. Yamakami, 1993. Unpublished.