SIAM J. COMPUT.
Vol. 10, No. 1, February 1981

0097-5397/81/1001-0008 $01.00/0

RELATIVE TO A RANDOM ORACLE A, $\mathbf{P}^A \neq \mathbf{NP}^A \neq \text{co-}\mathbf{NP}^A$ WITH PROBABILITY 1*

CHARLES H. BENNETT† AND JOHN GILL‡

Abstract. Let A be a language chosen randomly by tossing a fair coin for each string x to determine whether x belongs to A. With probability 1, each of the relativized classes $\mathbf{LOGSPACE}^A$, $\mathbf{P}^A$, $\mathbf{NP}^A$, $\mathbf{PP}^A$, and $\mathbf{PSPACE}^A$ is properly contained in the next. Also, $\mathbf{NP}^A \neq \text{co-}\mathbf{NP}^A$ with probability 1. By contrast, with probability 1 the class $\mathbf{P}^A$ coincides with the class $\mathbf{BPP}^A$ of languages recognized by probabilistic oracle machines with error probability uniformly bounded below $\frac{1}{2}$. $\mathbf{NP}^A$ is shown, with probability 1, to contain a $\mathbf{P}^A$-immune set, i.e., a set having no infinite subset in $\mathbf{P}^A$. The relationship of $\mathbf{P}^A$-immunity to p-sparseness and $\mathbf{NP}^A$-completeness is briefly discussed: $\mathbf{P}^A$-immune sets in $\mathbf{NP}^A$ can be sparse or moderately dense, but not co-sparse. Relativization with respect to a random length-preserving permutation π, instead of a random oracle A, yields analogous results and in addition the proper containment, with probability 1, of $\mathbf{P}^\pi$ in $\mathbf{NP}^\pi \cap \text{co-}\mathbf{NP}^\pi$, which we have been unable to decide for a simple random oracle. Most of these results are shown by straightforward counting arguments, applied to oracle-dependent languages designed not to be recognizable without a large number of oracle calls. It is conjectured that all p^A-invariant statements that are true with probability 1 of subrecursive language classes uniformly relativized to a random oracle are also true in the unrelativized case.

Key words. random oracle, relativized computation, probabilistic computation, computational complexity, nondeterministic computation, polynomial immunity, polynomial isomorphism, polynomial reducibility

1. Introduction. A paper by Baker, Gill and Solovay [BGS], whose notation and definitions we adopt, has indicated the subtlety of the $\mathbf{P} = ?\mathbf{NP}$ question by exhibiting computable sets A and B such that $\mathbf{P}^A = \mathbf{NP}^A$ but $\mathbf{P}^B \neq \mathbf{NP}^B$. Here, $\mathbf{P}^X$ denotes the class of languages accepted by polynomial time bounded Turing machines able to query the set X, and $\mathbf{NP}^X$ denotes the corresponding class for nondeterministic machines.

This paper deals not with particular oracle sets but rather with statements that hold with probability 1 when the oracle is chosen randomly. The probability measure μ on the class of oracles is defined by putting each string into a random oracle with probability $\frac{1}{2}$, independent of all other strings. (Of course, such an oracle is noncomputable with probability 1). Random oracles provide easy examples of sets such as B of [BGS], and also indicate a new sense in which $\mathbf{P}^X \neq \mathbf{NP}^X$ for "most" oracles X. This is a counterpart for the nondenumerable class of all oracles of Mehlhorn's result [Me] that the subset of computable oracles X that satisfy $\mathbf{P}^X = \mathbf{NP}^X$ is effectively meager.

Any property of oracles that is insensitive to finite changes in the oracle has probability 0 or 1, by the zero-one law for tail events [Fe2]. We determine relationships that hold with probability 1 for language classes relativized to a random oracle A. Section 2 establishes the basic results $\mathbf{P}^A \neq \mathbf{NP}^A \neq \text{co-}\mathbf{NP}^A$ with probability 1, and the related results $\mathbf{LOGSPACE}^A \neq \mathbf{P}^A$ and $\mathbf{PSPACE}^A \neq \mathbf{EXPTIME}^A$ with probability 1. Section 3 relativizes the probabilistic language classes **PP** (languages recognizable in polynomial time by weak Monte Carlo tests, whose error probability may approach that of random guessing), and **BPP** (languages recognizable in polynomial time by strong Monte Carlo tests, whose error probability can be made as small as desired by iterating the test a fixed number of times). It is shown that with probability 1, the relativized class $\mathbf{PP}^A$ is properly contained in $\mathbf{PSPACE}^A$ and properly contains $\mathbf{NP}^A \cup \text{co-}\mathbf{NP}^A$. By

* Received by the editors November 6, 1979, and in final form May 20, 1980. This research was supported in part by the National Science Foundation, under grant MCS77-07555.

† IBM Watson Research Center, Yorktown Heights NY 10598.

‡ Electrical Engineering Department, Stanford University, Stanford CA 94305.

contrast, $\mathbf{P}^A$ and $\mathbf{BPP}^A$ are shown to be equal with probability 1. Section 4 shows that with probability 1, $\mathbf{NP}^A$ contains a $\mathbf{P}^A$-immune set, that is, a set having no infinite subset in $\mathbf{P}^A$. Section 5 discusses the open question of whether, relative to a random oracle, $\mathbf{P}^A$ equals $\mathbf{NP}^A \cap \text{co-}\mathbf{NP}^A$, arguing that it will be hard to decide one way or the other. On the other hand, by relativizing with respect to a random permutation π instead of a random oracle, $\mathbf{P}^\pi$ can be shown to be properly within $\mathbf{NP}^\pi \cap \text{co-}\mathbf{NP}^\pi$. Indeed, $\mathbf{NP}^\pi \cap \text{co-}\mathbf{NP}^\pi$ contains a $\mathbf{P}^\pi$-immune set with probability 1.

Most oracles used in recursive function theory and complexity theory contain built-in structure intended to help or frustrate a specific class of computations. A random oracle, on the other hand, is intuitively unbiased and unstructured; thus, it is plausible that theorems (for example, $\mathbf{P} \neq \mathbf{NP}$) that hold with probability one for computations relativized to a random oracle should also be true in the absence of an oracle. Section 6 formalizes this conjecture.

As a preview of the results to be demonstrated later, we now give heuristic arguments showing why, relative to a typical random oracle, deterministic and nondeterministic polynomial time are different ($\mathbf{P}^A \neq \mathbf{NP}^A$, Theorem 1), but deterministic and probabilistic time are the same ($\mathbf{P}^A = \mathbf{BPP}^A$, Theorem 5). Given a fixed but typical random oracle, consider the following question: do the first 2^n bits of the oracle's characteristic sequence include any run of n consecutive zeros? Such a run will be present for about half of all values of n, and if present, it could easily be detected nondeterministically by guessing the address of its beginning. On the other hand, it is fairly obvious, if not entirely straightforward to prove, that no deterministic algorithm could expect to find out whether a run exists in less than exponential time. Thus, for typical random oracles A, the language $\{0^n$: the first 2^n bits of A contain a run of n consecutive zeros$\}$ is in $\mathbf{NP}^A$–$\mathbf{P}^A$. Similarly, the language $\{0^n$: the first 2^n bits of A contain an even number of zeros$\}$ is in $\mathbf{PSPACE}^A$–$\mathbf{NP}^A$ with probability 1.

Next consider a language, such as the set of composite numbers, that is probabilistically recognizable in the sense of **BPP**. Such a language could be recognized deterministically in the presence of a random oracle by: 1) iterating the original Monte Carlo test a linearly increasing number of times as a function of input size, so that the expected cumulative number of errors, summing over all inputs, remains finite; 2) simulating this more accurate Monte Carlo algorithm deterministically by using bits from the random oracle instead of coin tosses; 3) patching the errors by a finite table. A slight refinement of this argument shows that even relativized languages of the class $\mathbf{BPP}^A$ can be recognized in deterministic polynomial time with the help of a random oracle.

Throughout this paper, the natural number x will be identified with the xth binary string in lexicographic order $(0, 1, 2, 3, \cdots \leftrightarrow \Lambda, 0, 1, 00 \cdots)$. The binary length of x, equal to the integral part of $\log_2 (x+1)$, will be denoted $|x|$. Similarly, a set or language A will be identified with its characteristic sequence, the infinite binary sequence whose xth bit, $A(x)$, is 1 iff $x \in A$. Sets of sets (e.g., language classes or events in oracle space) will be denoted by upper case Greek letters, with Ω denoting the (nondenumerable) set of all languages. The probability measure on Ω is equivalent, via the identification of languages with infinite binary sequences, to Lebesgue measure on the unit interval.

Most of the separation results in this paper are proved by exhibiting an oracle-dependent test language L^A which belongs to the one of two relativized language classes (e.g., $\mathbf{NP}^A$) for all oracles A but belongs to another narrower class (e.g., $\mathbf{P}^A$) only for a set of oracles of measure zero. Results of this sort can be proven more easily by appealing to the following lemma, which depends on certain easily satisfied conditions

on the test language L^A and the (denumerable) relativized class $\mathbf{M}^A = \{M_1^A, M_2^A, M_3^A, \cdots\}$ to which it is desired to prove that L^A does not belong for most A. For concreteness, M_i^A may be thought of as the language accepted by the ith machine of a given type (e.g., polynomial time bounded) when it is connected to oracle A. First the conditions will be described; then the lemma will be stated and proved.

Condition 1. The test language L^A, and each of the machine languages M_i^A, must depend on A via a total recursive operator from Ω to Ω. Each of these languages, in other words, must be recognizable by a Turing machine that halts for all oracles and inputs.

Condition 2. The family of machine languages should be finitely patchable with respect to the oracle: for each machine M_i and each finite bit string s there should exist another machine M_k such that $M_k^A = M_i^{s*A}$ for all oracles A. Here $s*A$ denotes the characteristic sequence obtained by substituting the finite string s for the first $|s|$ bits of A. Machine M_k may be thought of as incorporating the bit string s in its finite control, where it intercepts and answers all sufficiently small queries.

Condition 3. The family of machine languages should be finitely patchable with respect to initial portions of any uniformly A-recursive language: For any number m, any machine M_i, and any A-recursive function φ_i^A that is total and 0-1 valued for all A, there should exist another machine M_k such that for all oracles A and inputs x,

$$M_k^A(x) = \begin{cases} \varphi_i^A(x) & \text{if } x < m, \\ M_i^A(x) & \text{otherwise.} \end{cases}$$

In particular, when φ_i^A defines a test language L^A, machine M_k gives the "correct" answer $L^A(x)$ for inputs less than m and gives the same answer as machine M_i would for all other inputs.

Condition 4. The test language L^A (but not necessarily the machine languages M_i^A) must depend on the oracle in such a way that each bit of the oracle affects only finitely many bits of the language. (Condition 1, by König's lemma, implies that both L^A and M_i^A already satisfy the converse of condition 4, namely, that each bit of the language depends on only finitely many bits of the oracle.) Together, conditions 1 and 4 require that the membership of x in L^A depend only on those addresses in A lying in a finite window bounded by two monotone functions of x that tend to ∞ in the limit of large x. Oracle-dependent languages of this sort have been termed "oracle properties" by Angluin [An] and Kozen and Machtey [KM].

Conditions 1 and 4 hold by definition for all the test languages used in this paper, and conditions 1–3 can readily be seen to hold for the relevant families of oracle machines, viz. logspace bounded deterministic [LL,Si], polynomial time bounded nondeterministic [BGS], and polynomial time bounded probabilistic threshold machines ([Gi], see also § 3).

LEMMA 1. *Let L^A be a test language and $\mathbf{M}^A = \{M_1^A, M_2^A \cdots\}$ a family of machine languages satisfying conditions 1–4 above. If there exists a positive constant ε such that each machine language differs from the test language for a class of oracles of measure $> \varepsilon$, then the class of oracles for which $L^A \in \mathbf{M}^A$ has measure zero.*

Proof. The idea of the proof is to show that as a machine is fed larger and larger inputs, it keeps making fresh errors, due to bad luck at oracle addresses too large to have caused any errors earlier.

It suffices to show, for each machine M_i, that the class $\mathbf{C}_m = \{A: \forall x < m\, L^A(x) = M_i^A(x)\}$, of oracles for which it makes no error on the first m inputs, approaches measure zero in the limit $m \to \infty$. To prove this it suffices to show

that for each m there exists a larger n such that $\mu(\mathbf{C}_n) \leqq (1-\varepsilon)\mu(\mathbf{C}_m)$. Because of condition 1, membership of an oracle in the class $\mathbf{C}_m$ depends on only a finite portion of the oracle characteristic sequence; hence $\mathbf{C}_m$ may be expressed as a finite disjoint union of elementary cylinders $\mathbf{Z}_s$, where $\mathbf{Z}_s$ is the class of oracles whose characteristic sequences begin with the finite sequence s.

In view of this, the lemma would follow if one could show that ε is a lower bound not only for the overall error probability, $\lim_{n\to\infty} 1-\mu(\mathbf{C}_n)$, but also for the conditional error probability within any cylinder, $\lim_{n\to\infty} 1-\mu(\mathbf{Z}_s \cap \mathbf{C}_n)/\mu(\mathbf{Z}_s)$, even though the cylinder $\mathbf{Z}_s$ might consist entirely of oracles that cause no errors on small inputs.

To prove that this is indeed the case, note that the operation of M_i in any cylinder $\mathbf{Z}_s$ may be simulated by another, finitely patched, machine M_k, which accepts the following oracle-dependent language: **if** $[L^{s^*A}(x) \neq L^A(x)]$ **then** $L^{s^*A}(x)$ **else** $M_i^{s^*A}(x)$. Here condition 4 guarantees that $L^A(x)$ and $L^{s^*A}(x)$ differ for only finitely many x, and conditions 2 and 3 guarantee the existence of the patched machine. It is evident from the definition that the original machine's conditional error probability on cylinder $\mathbf{Z}_s$ is at least as great as the patched machine's unconditional error probability which in turn is at least ε, by the premise of the lemma. □

Remark. Kozen and Machtey [KM] derive a result analogous to Lemma 1 but for meagerness rather than measure. Under conditions 1–4, they show that set $\{A: L^A \in \mathbf{M}^A\}$ is either equal to all of oracle space or else is a meager subset of oracle space. Therefore, whenever Lemma 1 is used to prove a separation with probability 1 between two relativized complexity classes, the same separation holds for all but a meager subset of oracles. On the other hand, the possibility remains that two complexity classes may be equal with probability 1 even though they differ for all but a meager subset of oracles. This possibility is discussed further in connection with Theorem 5.

2. $\mathbf{P}^A$, $\mathbf{NP}^A$, and $\mathbf{LOGSPACE}^A$ for random oracles A. The following definition provides a function $\xi_A(x)$ that uses the oracle A to map binary strings randomly into strings of the same length.

DEFINITION. $\xi_A(x) = A(x1)A(x10)A(x100)\cdots A(x10^{|x|-1})$, where juxtaposition indicates concatenation. In other words, $\xi_A(x)$ is a $|x|$-bit string whose kth bit is 1 or 0 according to whether $x \times 10^{k-1}$ belongs to A.

Although it is easily computed by a machine with oracle A, the function ξ_A is ideally pseudorandom in that knowing its value for one argument tells nothing about its value for other arguments. (The same is true of the characteristic function $A(x)$, but in several of the proofs below it is convenient to have a function whose values are about the same size as its arguments.) The pseudorandomness of ξ_A is used to define languages depending on A that cannot be accepted without exponentially many queries of the oracle. The number of inverse images under ξ_A approaches a Poisson distribution for large n: for typical A the fraction of n-bit strings with exactly k inverse images under ξ_A approaches $e^{-k}/k!$ In particular, about $1/e$ of n-bit strings have no inverse image and $1/e$ have exactly one inverse image.

THEOREM 1. *If A is a random oracle, the* $\mathbf{P}^A \subsetneqq \mathbf{NP}^A \neq \text{co-}\mathbf{NP}^A$ *with probability* 1.

Proof. Since $\mathbf{P}^A$ is closed under complementation, Theorem 1 would follow if, for all but a class of oracles A of measure zero, one could exhibit a language in $\mathbf{NP}^A$ whose complement is not in $\mathbf{NP}^A$. Let the test language RANGE^A be defined as $\{x: \exists_y \xi_A(y) = x\}$, i.e., the range of ξ_A, and let CORANGE^A be the complement of RANGE^A. Clearly, RANGE^A belongs to $\mathbf{NP}^A$. However, CORANGE^A is not in $\mathbf{NP}^A$ because, intuitively, no nondeterministic oracle machine can verify for typical x and A that x is not in RANGE^A without evaluating $\xi_A(y)$ for every y of length $|x|$.

In order to show that with probability 1, no polynomial time bounded nondeterministic oracle machine NP_i with random oracle A accepts exactly CORANGE^A, it suffices by Lemma 1 to show that every such machine has an input on which it errs with probability at least $\frac{1}{3}$ when A is chosen randomly.

Let an arbitrary machine NP_i be chosen and consider an input of the form $x = 0^n$, where n is sufficiently large that none of the machine's nondeterministic computation paths has time to examine more than one per cent of the 2^n n-bit strings that are potential inverse images of 0^n under the ξ function. Recalling the definition of the ξ function, an n-bit string y will be said to be *examined* when the oracle is queried about any string of the form $y10^k$, for some $k < n$.

Let $\mathbf{C}_0 = \{A: \neg\exists y\, \xi_A(y) = 0^n\}$ be the class of oracles for which the input 0^n is in CORANGE^A and therefore should be accepted. This class has measure between 0.36 and 0.37 for all $n \geqq 5$, approaching $1/e = 0.3678\cdots$ for large n. Let α_0 be the conditional acceptance probability on $\mathbf{C}_0$, i.e., the fraction of oracles in $\mathbf{C}_0$ for which input 0^n actually is accepted.

Consider now another class of oracles, disjoint from $\mathbf{C}_0$ and consisting of oracles A for which 0^n has exactly one inverse image but is not its own inverse image. This class, $\mathbf{C}_1 = \{A: \xi_A(0^n) \neq 0^n \text{ and } (\exists^{\text{uniq}}\, y)\xi_A(y) = 0^n\}$, has measure exactly equal to that of $\mathbf{C}_0$ and consists entirely of oracles for which the input 0^n does not belong to CORANGE^A and therefore should be rejected. Let α_1 be the conditional acceptance probability on $\mathbf{C}_1$.

The overall error probability,

$$\varepsilon = \mu\{A: NP_i^A(0^n) \neq \text{CORANGE}^A(0^n)\},$$

is at least

$$(1-\alpha_0)\mu(\mathbf{C}_0) + \alpha_1\mu(\mathbf{C}_1) \approx (1+\alpha_1-\alpha_0)/e,$$

since every rejection in $\mathbf{C}_0$, and every acceptance in $\mathbf{C}_1$, is an error. In order to show that $\varepsilon > \frac{1}{3}$, we exhibit a probabilistic transformation of oracles, $A \to A'$, that maps $\mathbf{C}_0$ onto $\mathbf{C}_1$ in a measure-preserving manner but changes each oracle so little that most accepting computation paths under A continue to accept under A'. Therefore, $\alpha_1 \geqq \alpha_0$ and $\varepsilon \geqq 1/e$.

The transformation $A \to A'$ is best described in words. To obtain A' from A, choose randomly (by coin tossing) an n-bit string z not equal to 0^n; then delete from A all strings of the form $z \times 10^i$ for $i < n$. Recalling the definition of the ξ function, this has the effect of making $\xi_{A'}(z) = 0^n$ while preserving the equality $\xi_A(y) = \xi_{A'}(y)$ for all other arguments y. The transformation is therefore measure preserving between $\mathbf{C}_0$ and $\mathbf{C}_1$, in the sense that the expectation of any event in $\mathbf{C}_1$ is equal to the expectation that a randomly chosen point in $\mathbf{C}_0$ will map into it under the transformation. (The probabilistic transformation may be thought of more formally as a deterministic measure preserving mapping $(A, z) \to (A', \xi_A(z))$ from $\mathbf{C}_0 \times \mathrm{Y}$ onto $\mathbf{C}_1 \times \mathrm{Y}$, where Y is the probability space of n-bit strings not equal to 0^n. Hence, for any event $\mathbf{E} \subseteq \mathbf{C}_1$, $\mu(\mathbf{E}) = \mu\{\langle A, z\rangle \in \mathbf{C}_0 \times \mathrm{Y}: A' \in \mathbf{E}\}$.)

To show that $\alpha_1 \geqq \alpha_0$, choose a random oracle in $\mathbf{C}_0$ and a random n-bit string $z \neq 0^n$ and generate the transformed oracle A', a member of class $\mathbf{C}_1$. With probability α_0, there is at least one accepting path of $NP_i(0^n)$ under oracle A. Select the first accepting path. With conditional probability at least 0.99, the set of strings examined on this path does not include z, the one string with respect to which oracles A and A' differ, and so the path continues to accept under A'. Therefore the acceptance probability in $\mathbf{C}_1$ is at least 0.99 times that in $\mathbf{C}_0$ and $\varepsilon \geqq 0.36(1-\alpha_0+0.99\alpha_0) > \frac{1}{3}$.

Lemma 1 then allows us to conclude that, with probability 1, CORANGE^A is not in $\mathbf{NP}^A$. Since RANGE^A is in $\mathbf{NP}^A$, we have, with probability 1, $\mathbf{P}^A \neq \mathbf{NP}^A \neq \text{co-}\mathbf{NP}^A$. □

$\mathbf{LOGSPACE}^A$ can be defined in various ways, depending on how the query tape is handled. We follow the conventions of Ladner and Lynch [LL]: The query tape is not charged against the space bound, but to keep it from being used as a work tape, the query tape is one-way and write-only and is erased automatically following each query. (Simon [Si] treats the query tape as one of the work tapes, a two-way read/write tape that is charged against the space bound. The Ladner-Lynch definition is less restrictive and perhaps more natural, since for a random oracle $A \in \mathbf{LOGSPACE}^A$ holds with probability 1 for [LL] but not for [Si]. Theorem 2 holds for both definitions of $\mathbf{LOGSPACE}^A$.)

THEOREM 2. *If A is a random oracle, then* $\mathbf{LOGSPACE}^A \neq \mathbf{P}^A$ *with probability* 1.

Proof. The language used to prove Theorem 2 is $\text{BIGQUERY}^A = \{x : \xi_A(x) \in A\}$, which is obviously in $\mathbf{P}^A$ for every oracle A. Every oracle machine that recognizes this language must compute and store some representation of $\xi_A(x)$ on its work tape, which costs at least $|x|$ bits. Queries of the form "$x \times 10^i \in A$?" can be asked within the log space bound by simply transferring x from the input tape to the query tape, followed by the appropriate number of zeros. Such queries suffice to determine individual bits of the string $\xi_A(x)$. However, these bits cannot be accumulated on the query tape, since it is meanwhile being used for other queries, nor can they be stored on the work tape without violating the space bound. Not knowing $\xi_A(x)$, a logspace bounded machine must therefore, for every sufficiently large x, err with probability nearly $\frac{1}{2}$ in deciding whether $\xi_A(x)$ belongs to A.

More formally, let M be a logspace bounded deterministic oracle machine. A string y of length n is *queriable* by M if there is an oracle X for which y is queried by M^X on input 0^n. Initially, and just after each oracle query, the query tape is blank. Since M is logspace bounded, the total number of distinct machine states (instantaneous descriptions) with a blank query tape is at most cn^k for constants c and k depending on M but independent of n. When M is started in any one of these states, the computation proceeds deterministically, and independently of the oracle, until the next query (or until halting if no further queries were made). Therefore at most cn^k n-bit strings are queriable.

On the other hand, as the oracle A is varied, $\xi_A(0^n)$ takes on any of 2^n distinct values, all equally likely. Let $\mathbf{C} = \{A : M^A(0^n) \text{ queries } \xi_A(0^n)\}$ be the class of oracles for which $\xi_A(0^n)$ is actually queried. $\mathbf{C}$ is a subclass of $\{A : \xi_A(0^n) \text{ is queriable}\}$, and so $\mathbf{C}$ has measure at most $cn^k/2^n$, which approaches 0 for large n. Therefore $\bar{\mathbf{C}}$, the class of oracles for which $M^A(0^n)$ does *not* query $\xi_A(0^n)$, has measure 1 in the limit.

If M^A does not query $\xi_A(0^n)$, then it is obviously in a poor position to decide whether 0^n is in BIGQUERY^A, that is, whether $\xi_A(0^n)$ is in A. Consider the measure-preserving transformation of oracles that removes from A if it is present, or adds to A if it is absent, the string $\xi_A(0^n)$. This transformation maps $\bar{\mathbf{C}}$ onto itself, and for every oracle in $\bar{\mathbf{C}}$ changes the truth of $0^n \in \text{BIGQUERY}^A$ without changing the machine's answer $M^A(0^n)$. Therefore, for each machine M, the class of oracles on which $M^A(0^n)$ errs in determining whether 0^n belongs to BIGQUERY^A has measure nearly $\frac{1}{2}$ for large n. By Lemma 1, with probability 1 BIGQUERY^A is not in $\mathbf{LOGSPACE}^A$. □

COROLLARY. *If A is a random oracle, then* $\mathbf{PSPACE}^A \neq \mathbf{EXPTIME}^A$ *with probability* 1.

Proof. As above, using $\text{VERYBIGQUERY}^A = \{x : \xi_A(0^x) \in A\}$ as the test language. With probability 1, this test language is in $\mathbf{EXPTIME}^A$ but not in $\mathbf{PSPACE}^A$.

3. Probabilistic polynomial time languages. This section investigates the relativized classes of languages computable in polynomial time by probabilistic oracle machines [Gi]. Probabilistic machines are equipped with a coin toss mechanism that enables them to make fresh random choices during a computation. This randomness should be distinguished from the randomness of the oracle A, which is fixed before the computations begin. (However, some of the theorems below are proved by using the random oracle to simulate coin tosses, or vice versa.)

The language *accepted* by a probabilistic machine M with oracle A is defined as the set of inputs for which the machine halts in an accepting state with probability greater than $\frac{1}{2}$, and the characteristic function $M^A(x)$ takes on the value 1 or 0 according to this majority result (if the acceptance probability is exactly $\frac{1}{2}$, $M^A(x)=0$). The *error probability* of M^A on input x is defined as the fraction of coin toss sequences leading to nonacceptance if $M^A(x)=1$, or to acceptance if $M^A(x)=0$. A probabilistic oracle machine M is polynomial time bounded if there exists a polynomial p such that, for all oracles A and inputs x, all computation paths halt within $p(|x|)$ steps.

Several classes of probabilistic polynomial time languages can be defined, depending on the allowed error probability.

DEFINITION. Let A be any oracle set.

1) $\mathbf{PP}^A$ is the class of languages accepted by polynomial time bounded probabilistic oracle machines with oracle A. Simon [Si] has shown that the same class results if the definition is strengthened to include only languages recognizable by machines with error probability less than $\frac{1}{2}$ on all inputs nonmembers as well as members.

2) $\mathbf{BPP}^A$ is the class of languages accepted by polynomial time bounded probabilistic oracle machines with error probability uniformly bounded below $\frac{1}{2}$. A language L is in $\mathbf{BPP}^A$ iff there is a polynomial time bounded probabilistic oracle machine M and a constant $\varepsilon<\frac{1}{2}$ such that $L=M^A$ and the error probability of M^A is less than ε for all inputs, members as well as nonmembers.

The difference between **BPP** and **PP** is that for languages in **BPP** the error probability can be made uniformly as small as desired by repeating the probabilistic computation a uniform number of times, whereas this is not generally possible for a language in **PP**. In particular, if a language L is recognizable with error probability uniformly below $\varepsilon<\frac{1}{2}$, then performing the computation m times and taking the majority decision (m odd) suffices to reduce the error probability uniformly below

$$\sum_{k=0}^{(m-1)/2}\binom{m}{k}\varepsilon^{m-k}(1-\varepsilon)^k,$$

which approaches zero exponentially with increasing m (this follows from the fact that for large m, the binomial distribution approximates a normal distribution of standard deviation $\sqrt{m\varepsilon(1-\varepsilon)}$ and mean $(1-\varepsilon)m$; and the fact that the area under the tail of the normal curve, from $-\infty$ to a point x standard deviations below the mean, is bounded above by $\text{const}\times\exp(-x^2/2)$ [Fe]). Thus, $\mathbf{BPP}^A$ may be defined without loss of generality as the set of languages accepted by polynomial time bounded probabilistic oracle machines M^A with error probability uniformly below, say, $\frac{1}{4}$.

A well-known subclass of **BPP** is the class called **R** [AM], [Ra] or **VPP** [Gi] consisting of languages, such as the composite numbers, that are probabilistically recognizable in polynomial time by one-sided Monte Carlo tests that never accept a nonmember of the language. **BPP** includes such languages and their complements, as well as languages (no natural examples are known) for which only two-sided Monte Carlo tests exist.

Another subclass of **BPP**, known as **ZPP**[Gi], may be defined as $\mathbf{R} \cap \text{co-}\mathbf{R}$, or equivalently as the class of languages recognizable by probabilistic machines with zero error probability and polynomial bounded *average* run time.

It is easily shown [Gi] that $\mathbf{P} \subseteq \mathbf{ZPP} \subseteq \mathbf{BPP} \subseteq \mathbf{PP} \subseteq \mathbf{PSPACE}$ and, perhaps more surprisingly, that $\mathbf{NP} \subseteq \mathbf{PP}$. From the definitions, it is obvious that **PP**, **BPP**, and **ZPP** are closed under complementation. All these relations continue to hold when the classes are relativized to an arbitrary oracle. In this section, we show that, relative to a random oracle A, the classes $(\mathbf{NP}^A \cup \text{co-}\mathbf{NP}^A) \subsetneq \mathbf{PP}^A \subsetneq \mathbf{PSPACE}^A$ are distinct with probability 1, whereas $\mathbf{BPP}^A = \mathbf{P}^A$ with probability 1.

THEOREM 3. *If A is a random oracle, then* $\mathbf{PP}^A \subsetneq \mathbf{PSPACE}^A$ *with probability* 1.

Proof. Let $\text{ODD}^A = \{x$: an odd number of strings of length $|x|$ are in $A\}$. ODD^A is computable in linear space with oracle A, and so ODD^A is in $\mathbf{PSPACE}^A$. On the other hand, it is intuitively clear that a probabilistic algorithm to decide whether x is in ODD^A without querying all strings of length $|x|$ must, for typical x and A, have an error probability of exactly $\frac{1}{2}$.

For any polynomial time bounded probabilistic oracle machine M, let $\varepsilon(x, A)$ be the error probability of M with oracle A and input x. The computation path of M^A on input x is determined by the random Bernoulli sequence B of coin tosses. Therefore, the error probability can be written as $\varepsilon(x, A) = \mu\{B: M^{AB}(x) \neq \text{ODD}^A(x)\}$, where $M^{AB}(x)$ is the output of $M^A(x)$ with coin toss sequence B and μ is Lebesgue measure on the set of infinite coin toss sequences.

Choose an input x so large that no computation path of $M^A(x)$ has time to query all strings of length $|x|$. Let $\mathbf{C}^+ = \{A: \varepsilon(x, A) < \frac{1}{2}\}$ and $\mathbf{C}^- = \{A: \varepsilon(x, A) > \frac{1}{2}\}$. We shall show that $\mu(\mathbf{C}^+) = \mu(\mathbf{C}^-)$.

We define a measure-preserving transformation $(A, B) \to (A', B)$, in the product space $\Omega_A \times \Omega_B$ of oracles A with Bernoulli sequences B, which maps $\mathbf{C}^+ \times \Omega_B$ onto $\mathbf{C}^- \times \Omega_B$ and vice-versa. The transformation consists of adding to A if it is absent, or removing from A if it is present, the first string of length $|x|$ not queried in the computation path $M^{AB}(x)$. The transformation thus always changes the value of $\text{ODD}^A(x)$ while never changing the machine's answer $M^{AB}(x)$. Hence, it maps $\mathbf{C}^- \times \Omega_B$ onto $\mathbf{C}^+ \times \Omega_B$ and vice versa. Therefore, $\mu(\mathbf{C}^+) = \mu(\mathbf{C}^-)$.

Since $\mathbf{C}^- \subseteq \bar{\mathbf{C}}^+$, we conclude that $\mu(\mathbf{C}^+) \leq \frac{1}{2}$. For all oracles not in $\mathbf{C}^+$, the machine M^A does not correctly decide whether x is in ODD^A. Therefore, by Lemma 1, with probability 1, ODD^A is not in $\mathbf{PP}^A$. □

THEOREM 4. *If A is a random oracle, then* $\mathbf{NP}^A \cup \text{co-}\mathbf{NP}^A \subsetneq \mathbf{PP}^A$ *with probability* 1.

Proof. By Theorem 1, RANGE^A is in $\mathbf{NP}^A$–co-$\mathbf{NP}^A$ and CORANGE^A is in co-$\mathbf{NP}^A$–$\mathbf{NP}^A$ with probability 1. Therefore, with probability 1, the combined language RANGE^A **join** $\text{CORANGE}^A = \{0x: x \in \text{RANGE}^A\} \cup \{1x: x \in \text{CORANGE}^A\}$ is in neither $\mathbf{NP}^A$ nor co-$\mathbf{NP}^A$ but is in $\mathbf{PP}^A$ because both $\mathbf{NP}^A$ and co-$\mathbf{NP}^A$ are subclasses of $\mathbf{PP}^A$. □

Remark. This same example establishes that with probability 1 $\mathbf{NP}^A \cup \text{co-}\mathbf{NP}^A$ is properly contained in the class $\Delta_2^{\mathbf{P},A}$ of languages recognizable in polynomial time relative to an oracle in $\mathbf{NP}^A$. $\Delta_2^{\mathbf{P},A}$ is a member of the relativized Meyer-Stockmeyer **P**-hierarchy [MS], [BGS] and [BS], a polynomially time bounded analogue of the Kleene arithmetical hierarchy [Ro]. Like $\mathbf{PP}^A$, it includes $\mathbf{NP}^A$ and is closed under complementation. Whether relativization by a random oracle separates classes higher than $\Delta_2^{\mathbf{P},A}$ in the hierarchy is currently unknown, as is the relationship between $\Delta_2^{\mathbf{P},A}$ and $\mathbf{PP}^A$.

THEOREM 5. *If A is a random oracle, then $\mathbf{P}^A = \mathbf{ZPP}^A = \mathbf{R}^A = \mathbf{BPP}^A$ with probability* 1.

Proof. It is sufficient to show that for every $\delta > 0$, the class of oracles A for which $\mathbf{P}^A \neq \mathbf{BPP}^A$ has measure less than δ. As noted earlier, $\mathbf{BPP}^A$ may be defined without loss of generality as the class of languages recognizable by probabilistic polynomial time bounded oracle machines M_i^A with error probability uniformly bounded below $\frac{1}{4}$ for all inputs.

For any polynomial time bounded probabilistic oracle machine M_i we can effectively construct another, $M_{f(i)}$, that recognizes the same language as does M_i but with smaller error probability; in fact, there is a recursive function f such that if the error probability $\varepsilon_i(x, A)$ of M_i^A on input x is less than $\frac{1}{4}$, then $\varepsilon_{f(i)}(x, A)$ decreases exponentially with i and $|x|$, being bounded above by $\delta \cdot 2^{-(i+2|x|+2)}$. The machine $M_{f(i)}$ takes the majority vote of $c(i+2|x|+2)$ independent computations of $M_i^A(x)$; the constant c depends on δ, but not on i, x or A.

Next, we construct a *deterministic* polynomial time bounded oracle machine $M_{g(i)}$ that operates as follows. With input x, it first computes $p_{f(i)}(|x|)$, a polynomial upper bound on the length of queries that can be made by $M_{f(i)}$. Such a bound always exists because of the polynomial time bound on $M_{f(i)}$. Then, $M_{g(i)}^A$ simulates the probabilistic oracle machine computation $M_{f(i)}^A(x)$. Each time the simulated computation $M_{f(i)}^A(x)$ requires a coin toss, $M_{g(i)}^A(x)$ obtains a bit by querying the oracle A about the least string of length greater than $p_{f(i)}(|x|)$ that has not yet been queried.

Let $\mathbf{E}_{ix}$ be the class of oracles A for which $M_i^A(x)$ has error probability less than $\frac{1}{4}$ but $M_{g(i)}^A(x)$ does not agree with the majority answer of $M_i^A(x)$. Since the queries made by $M_{g(i)}^A(x)$ in simulating coin tosses are larger than any queries actually made by any simulated probabilistic computation $M_{f(i)}^A(x)$, the measure of $\mathbf{E}_{ix}$ does not exceed the error probability of $M_{f(i)}^A(x)$. That is,

$$\mu(\mathbf{E}_{ix}) \leqq \max\{\varepsilon_{f(i)}(x, A): \varepsilon_i(x, A) < \tfrac{1}{4}\} < \delta \cdot 2^{-(i+2|x|+2)}.$$

Taking the union over i and x, we obtain

$$\mu\left(\bigcup_{ix} \mathbf{E}_{ix}\right) \leqq \sum_{ix} \mu(\mathbf{E}_{ix}) < \delta.$$

(The convergence of this sum does not require that the events $\mathbf{E}_{ix}$ be independent, merely that they individually be of small measure; in general the $\mathbf{E}_{ix}$ will be strongly correlated, because the construction allows the same oracle bit to simulate a coin toss for many different machines and inputs.) To conclude the proof, we observe that $\mathbf{P}^A = \mathbf{BPP}^A$ for every oracle A not in $\bigcup_{ix} \mathbf{E}_{ix}$, since for every such oracle, if language L is accepted by M_i^A with error probability uniformly less than $\frac{1}{4}$, then L is recognized by the deterministic oracle machine $M_{g(i)}$. □

Remark. For each δ in the above proof, the set of oracles $\mathbf{N}_\delta = \bigcap_{ix} \bar{\mathbf{E}}_{ix}$ is a nowhere-dense set in the sense of Mehlhorn [Me], and the union over δ of these sets is a meager set of measure 1 on which $\mathbf{P}^A \mathbf{BPP}^A$. This raises the interesting possibility that the set of *all* oracles for which $\mathbf{P}^A = \mathbf{BPP}^A$ may be sparse in one sense (Baire category theory), but co-sparse in another, more intuitive sense (measure).

COROLLARY. *If L is a non-oracle-dependent language which belongs to $\mathbf{P}^A$ with probability* 1 *for random A, then L belongs to the unrelativized class* **BPP**. *Conversely, every language in* **BPP** *is in $\mathbf{P}^A$ with probability* 1.

Proof. The first part follows from the ability of a probabilistic algorithm without oracle to simulate, by coin tossing, the answers a random oracle would give to a

deterministic algorithm. The converse is a special case of the theorem just proved: **BPP** is always a subclass of $\mathbf{BPP}^A$, which in turn is equal to $\mathbf{P}^A$, with probability one.

Remark. The second part of this corollary, that any language in **BPP** is in $\mathbf{P}^A$ with probability 1, generalizes to **BPP** Adleman's result [Ad] that any language in **R** has polynomial size circuits. A language L is in **R** iff every member of L is "witnessed" by at least half the strings of appropriate (polynomial $p(n)$) size and no nonmember is witnessed by any. Adleman showed that under these conditions, there exists for each n a specific set of $\leqq n$ witnesses sufficient to witness all members of L smaller than n bits. A fixed table of $np(n)$ bits is thus enough to simulate the approximately $2^n np(n)$ bits of witnesses that would be consulted if witnesses were generated probabilistically on each input.

In the proof of Theorem 5, if the language L accepted by probabilistic machine M_i is oracle-independent, belonging to **BPP** rather than merely to $\mathbf{BPP}^A$, then the bound $p_{f(i)}(|x|)$ on the size of queries by $M_{f(i)}$ can be taken to be zero. This means that the deterministic machine $M_{g(i)}$ uses the same initial bits of the random oracle, $A(1)$, $A(2)$, $A(3)$, $\cdots$ over and over again, to simulate the (in general different) coin toss sequences that the machines M_i and $M_{f(i)}$ would generate on different inputs. If the number of coin tosses made by the original probabilistic machine M_i is bounded by a polynomial $q(n)$ in the input length, then the number made by the more accurate machine $M_{f(i)}$ is bounded by a larger polynomial $cnq(n)$, and the number of random oracle bits needed by the deterministic machine to evaluate $L(x)$ accurately for all inputs of length $\leqq n$ is also bounded by $cnq(n)$. Thus, a fixed table of $cnq(n)$ random bits suffices to compute, without error, a finite set whose probabilistic computation, with errors, would use approximately $2^n q(n)$ coin toss bits.

4. $\mathbf{P}^A$-immunity. Classes such as **P** and **NP** refer to worst case performance. However, for RANGE^A and the other oracle-dependent languages discussed here, *most* members are as difficult to recognize as the worst case. A particularly strong form of this property is called **P**-immunity: a set is **P**-immune if it has no infinite subset that is in **P**. For typical oracles A, RANGE^A is not itself $\mathbf{P}^A$-immune, because, for example, it contains the $\mathbf{P}^A$-recognizable infinite subset $\{x: \xi_A(x) = x\}$. However, Theorem 6, proved later in this section, gives a set in $\mathbf{NP}^A$ that is $\mathbf{P}^A$-immune and $\mathbf{P}^A$-co-immune (i.e., its complement $\mathbf{P}^A$-immune) with probability 1. It is of course not known whether **NP** contains a **P**-immune set in the absence of an oracle, for that would imply $\mathbf{P} \neq \mathbf{NP}$, nor is it known whether all oracles X that make $\mathbf{P}^X \neq \mathbf{NP}^X$ also imply that $\mathbf{NP}^X$ contains a $\mathbf{P}^X$-immune set.

Another interesting question is whether there is an oracle X for which a set can be at once $\mathbf{P}^X$-immune and $\mathbf{NP}^X$-complete. (In order to define $\mathbf{NP}^X$-completeness, one must of course specify a reducibility relation. In § 6 it will be argued that, in order to be a fully relativized concept, $\mathbf{NP}^X$-completeness ought to be defined in terms of a relativized reducibility such as **P**, X-Turing reducibility, in which U is reducible to V iff $U \in \mathbf{P}^{X \text{ join } V}$, rather than the more customary **P**-Turing reducibility.) When X is the empty set, or a random oracle, immunity and completeness appear to be incompatible. Standard **NP**-complete sets such as $\text{SAT} = \{f$: the propositional formula f is satisfiable$\}$ contain infinite easy subsets, and so are not **P**-immune. Moreover, Berman and Hartmanis [BH] have shown that all known **NP**-complete sets are p-isomorphic, and conjecture that all **NP**-complete sets are.

This conjecture would imply that no **NP**-complete set is **P**-immune, since p-isomorphism preserves **P**-immunity. [*Proof.* Let sets U and V be p-isomorphic. Then, by definition of p-isomorphism there is a 1:1 onto function f with both f and f^{-1}

computable in polynomial time such that $x \in U$ iff $f(x) \in V$. Let U be not **P**-immune, and let $E \in \mathbf{P}$ be an infinite easy subset of U. Then, $f(E)$ is an infinite easy subset of V, making V not **P**-immune. $f(E)$ is infinite because f is 1:1, and $f(E) \in \mathbf{P}$ because $E \in \mathbf{P}$ and f^{-1} is computable in polynomial time. This argument also applies in a relativized form: for any A, if U and V are p-isomorphic, or even if they are only p^A-isomorphic, i.e., interconvertible by a permutation A-computable in polynomial time, then U is $\mathbf{P}^A$-immune iff V is $\mathbf{P}^A$-immune.]

The Berman–Hartmanis conjecture implies that complete sets cannot be p-sparse (a p-sparse set being one whose number of members of length $\leqq n$ is bounded by a polynomial in n). Immune sets, on the other hand, may be p-sparse or not; for typical random oracles A the $\mathbf{P}^A$-immune set of Theorem 6 below is moderately dense, but its intersection with a p-sparse set such as 0^* is p-sparse, still $\mathbf{P}^A$-immune, and still in $\mathbf{NP}^A$. (Recently Mahaney [Ma] has shown that, unless $\mathbf{P} = \mathbf{NP}$, no **NP**-complete set can be p-sparse).

Although $\mathbf{P}^A$-immune sets can be moderately dense, with probability 1 no $\mathbf{P}^A$-immune set in $\mathbf{NP}^A$ can be so dense that its complement is p-sparse. [*Proof.* Let A be a typical random oracle, and let S be a co-p-sparse set accepted by the nondeterministic machine NP_i^A. Since S is co-p-sparse, there exist probabilistic polynomial time algorithms (e.g., on input x, accept with probability $2^{-|x|}$) that, with probability arbitrarily close to unity, when applied to the inputs $0, 1, 2, \cdots$ in sequence, accept infinitely many members of S but no nonmembers. Each such probabilistic algorithm can be simulated by a deterministic polynomial time algorithm that queries A about strings too long to have been queried by NP_i^A on the same input. Thus, there is, for typical A, a deterministic algorithm to accept an infinite subset of S, rendering S not $\mathbf{P}^A$-immune.]

Although $\mathbf{P}^A$-immune sets in $\mathbf{NP}^A$ cannot be co-p-sparse, those not in $\mathbf{NP}^A$ can be. For example, the set $\{x: \forall j \leqq |x| \varphi_j^A(0) \neq x\}$ is co-p-sparse yet has no infinite A-r.e. subset. Hence, it is certainly $\mathbf{P}^A$-immune.

We now show that with probability 1, $\mathbf{NP}^A$ contains a $\mathbf{P}^A$-immune set.

THEOREM 6. *If A is a random oracle, the set* $\mathrm{RANGE3}^A = \{x: \exists y\, \xi_A(y) = xxx\}$ *and its complement are $\mathbf{P}^A$-immune with probability* 1. *Here, xxx denotes x thrice concatenated.*

Proof. $\mathrm{RANGE3}^A$ is infinite and co-infinite, and indeed about as dense as RANGE^A, having on the average $2^n(1-e^{-1})$ members each of length n. It is obviously in $\mathbf{NP}^A$. However, it is $\mathbf{P}^A$-immune because, intuitively, the expected cumulative number of successful guesses, on input x, of a string y that would map into xxx, approaches a finite limit as $x \to \infty$. Note that $\mathrm{RANGE3}^A$ is not $\mathbf{NP}^A$-complete, because it contains answers to only a few of the questions needed to recognize, say, RANGE^A in polynomial time.

To prove that $\mathrm{RANGE3}^A$ is $\mathbf{P}^A$-immune, it suffices to prove for each deterministic polynomial-time algorithm M that **C**, the class of oracles A for which that algorithm accepts an infinite subset of $\mathrm{RANGE3}^A$, is of measure zero.

Let M be applied to all inputs, $\Lambda, 0, 1, 00, \cdots$ in sequence and consider the finite set of oracle strings first examined in the course of the computation on input w:

$$\mathrm{EXAM}(A, w) = \{y: M^A(w) \text{ examines } y\} - \{y: \exists v < w M^A(v) \text{ examines } y\}.$$

Recall that a string y is said to be examined when any of the oracle strings affecting the value of $\xi_A(y)$ is queried. In general, we have regarded the oracle as having been chosen probabilistically in the beginning, after which computations proceed deterministically relative to it; however, when considering a fixed sequence of computations, it is

permissible to regard $\xi_A(y)$ as being decided probabilistically for each argument y at the time that argument is first examined. Subsequent evaluations of $\xi_A(y)$ must of course return the same value.

In order to be useful evidence in favor of accepting a member of RANGE3^A, an examined string y must have $\xi_A(y) = xxx$, for some x, and must have been examined sufficiently early, $\exists_{w \leq x} y \in \text{EXAM}(A, w)$, to influence the acceptance of x. The set of strings for which this is so may be defined:

$$\text{EVIDENCE}(A) = \bigcup_w \{y: y \in \text{EXAM}(A, w) \text{ and } \exists_{x \geq w} \xi_A(y) = xxx\}.$$

It is not difficult to see that, with probability 1, EVIDENCE(A) contains only finitely many members. To prove this, note that the polynomial bound on M implies that, for all but finitely many w, EXAM(A, w) contains fewer than $2^{|w|/2}$ members. Furthermore, since at the time each y in EXAM(A, w) is first examined, it has by definition not been examined before, the event $\{A: \exists x \geq w \xi_A(y) = xxx\}$ is independent of all previously examined parts of the oracle, and has probability $2^{-2|w|}$ or less, because of the preponderance of $3n$-bit strings not of the form xxx. Summing $2^{|w|/2} \cdot 2^{-2|w|}$ over all w, one obtains a finite expected number of strings in EVIDENCE(A), and, by the Borel–Cantelli lemma, this implies that $\mu\{A: \text{EVIDENCE}(A) \text{ is infinite}\} = 0$.

We now define $x_k(A)$ as the kth input string accepted without evidence under oracle A:

$$x_k(A) = \min \{x: x > x_{k-1}(A) \text{ and } x \in M^A \text{ and } \forall_{y \in \text{EVIDENCE}(A)} \xi_A(y) \neq xxx\}.$$

$x_k(A)$ may not always be defined (e.g., when M^A, the language accepted by M with oracle A, is finite, or when, with probability zero, EVIDENCE(A) is infinite); however, when infinitely many inputs are accepted, then (with conditional probability 1) all but finitely many of them are accepted without evidence.

The class $\mathbf{C} = \{A: M^A \text{ is an infinite subset of } \text{RANGE3}^A\}$, which we seek to show has measure zero, has the same measure as $\mathbf{D} = \mathbf{C} \cap \{A: \text{EVIDENCE}(A) \text{ is finite}\}$. $\mathbf{D}$, in turn, can be viewed as the limit of the nested sequence of classes $\mathbf{D}_1 \supset \mathbf{D}_2 \supset \mathbf{D}_3 \cdots$, where

$$\mathbf{D}_k = \{A: x_k(A) \text{ exists and } \forall_{i<k} x_i(A) \in \text{RANGE3}^A\}.$$

$\mathbf{D}$ can have nonzero measure only if the ratio $\mu(\mathbf{D}_k)/\mu(\mathbf{D}_{k-1})$ approaches unity as $k \to \infty$. However, it is easy to see that this ratio has a lim sup not exceeding $1 - e^{-1} \approx 0.632$. This is the limiting probability that, at the stage when input $x = x_k(A)$ is accepted without evidence, xxx, having no inverse image among the strings examined so far, does have an inverse image among the nearly $2^{3|x|}$ strings of length $3|x|$ not examined so far. Therefore, $\mu(\mathbf{D}) = \mu(\mathbf{C}) = 0$, and RANGE3^A is $\mathbf{P}^A$-immune with probability 1.

The proof that RANGE3^A is $\mathbf{P}^A$-co-immune with probability 1 proceeds similarly. Here it is even clearer that if infinitely many members of the complement of RANGE3^A are accepted, all but finitely many of them must be accepted without adequate evidence (no polynomial number of instances of y such that $\xi_A(y) \neq xxx$ can increase above $1/e \approx 0.368$, the asymptotic fraction of oracles for which $\forall_y \xi_A(y) \neq xxx$). □

Remark. The set $\text{RANGE2}^A = \{x: \exists_y \xi_A(y) = xx\}$ may also be $\mathbf{P}^A$-immune, inasmuch as the obvious strategy for recognizing members of it yields only finitely many. RANGE2^A and RANGE^A are $\mathbf{P}^A$-coimmune with probability 1.

5. Relativization of the P = ? NP ∩ co-NP question. It is unclear whether, relative to a random oracle A, $\mathbf{P}^A$ is properly contained in the intersection of $\mathbf{NP}^A$ and co-$\mathbf{NP}^A$.

If $\mathbf{P}^A = \mathbf{NP}^A \cap \text{co-}\mathbf{NP}^A$, then $\mathbf{P}^A$ includes such non-oracle-dependent, seemingly-difficult problems as factorization, known to be in $\mathbf{NP} \cap \text{co-}\mathbf{NP}$. By the corollary to Theorem 5, this would imply that such problems are solvable probabilistically in the sense of **BPP**, making them computationally tractable in a practical sense, contrary to appearances.

On the other hand, we have not been able to find an oracle-dependent language in $\mathbf{NP}^A - \mathbf{P}^A$ whose complement is also in $\mathbf{NP}^A - \mathbf{P}^A$. The attempt to construct an oracle-dependent language analogous, say, to FACTPROJ $= \{\langle x, y\rangle : x \geqq \text{prime-factorization-of}(y)\}$, which encodes factorization, is frustrated by the existence of multiple inverse images under the ξ function, in contrast with the uniqueness of factorization. Thus, FACTPROJ is in both **NP** and co-**NP**, but the obvious A-dependent analogue, $\text{XIPROJ}^A = \{\langle x, y\rangle : \exists_z x \geqq z \text{ and } \xi_A(z) = y\}$, like RANGE^A, is in $\mathbf{NP}^A$ but not co-$\mathbf{NP}^A$.

If we replace the random function $\xi_A(x)$ by a function $\pi(x)$ which randomly maps strings of each length onto one another in a 1 : 1 fashion (i.e., a permutation), then it is easy to show that, with probability 1, $\mathbf{P}^\pi$ is properly contained in $\mathbf{NP}^\pi \cap \text{co-}\mathbf{NP}^\pi$. The probability measure is the product measure over n of an assignment of equal weight $1/(2^n!)$ to each permutation of n-bit strings. This separation can be demonstrated using the oracle-dependent language $\text{PIPROJ}^\pi = \{\langle x, y\rangle : x \geqq \pi^{-1}(y)\}$, or, more simply, $\text{HALFRANGE}^\pi = \{x : \exists y\, \pi(0y) = x\}$. Both PIPROJ^π and HALFRANGE^π are in $(\mathbf{NP}^\pi \cap \text{co-}\mathbf{NP}^\pi) - \mathbf{P}^\pi$.

By a proof like that of Theorem 6, the oracle-dependent set $\text{HALFRANGE3}^\pi = \{x : \exists y\, \pi(0y) = xxx\}$, which belongs to $\mathbf{NP}^\pi \cap \text{co-}\mathbf{NP}^\pi$, can be shown to be $\mathbf{P}^\pi$-immune and $\mathbf{P}^\pi$-coimmune with probability 1.

All the theorems given earlier for complexity classes relativized to a random oracle A hold for the analogous complexity classes relativized to a random 1 : 1 function π. The π analogues of all but Theorem 3 are proved using the many-to-one random function $\xi_\pi(x) = [\text{the first } |x| \text{ bits of } \pi(xx)]$, which has nearly the same statistics as ξ_A. The π analogue of Theorem 3 can be proved using the language $\text{ODDPERM}^\pi = \{x : \pi$ performs an odd permutation on strings of length $|x|\}$. For any string length n, odd and even permutations are equiprobable, and they remain conditionally equiprobable as long as two or more arguments of the permutation remain unexamined. On the other hand, by exhaustively tracing all the permutation's cycles, its parity can be determined within a polynomial space bound. A random permutation can thus apparently substitute for a random oracle.

On the other hand, we can think of no way to use a random oracle A to construct a rapidly-evaluable random 1 : 1 function π, analogous to the construction of ξ_A from A; for this reason, the π function is less intuitively appealing, seeming to have more built-in structure, than the many-to-one ξ function.

Oracles with even more complicated kinds of randomness can be imagined, and indeed are apparently necessary to yield an easy proof, in the relativized setting, of certain putative properties of the natural number system, viz., the ability to support classical and public-key cryptography [DH]. A secure public-key cryptosystem, for example, exists with probability 1 relative to the oracle A **join** B, where A is a random oracle of the usual sort and B contains pairs of mutually-inverse random permutations indexed by A; e.g., for each n-bit string x, if u and v denote respectively the first and last halves of the $6n$-bit string $\xi_A(xxxxxx)$, then the functions $\xi_B(uy) = uz$ and $\xi_B(vz) = vy$ define mutually inverse random permutations between n-bit strings y and z. Each user of such a system picks an x randomly and secretly, finds u and v from it using A, and publishes u but not v. Other users then use B in conjunction with the public key u to encrypt messages $(y \to z)$ that only the original user, with private key v,

can economically decrypt $(z \to y)$ (using keys of length $3n$ rather than n insures that, despite the many-to-one nature of ξ_A, all but finitely many of the keys will be unique). The oracle A **join** B is a random analogue of the more complicated but recursive cryptographic oracles of Brassard [Br]. As Brassard points out, it is difficult to find an intuitively satisfactory asymptotic definition of cryptographic security. The relativized cryptosystem described above is secure in the ordinary, non-asymptotic sense that for typical message sizes (say $n = 100$), standard cyptanalytic tasks such as chosen plaintext attack could not be performed rapidly and reliably by a probabilistic query machine with a small number of internal states.

6. Discussion, random oracle hypothesis. Without oracles, the hierarchy of complexity classes includes the following known relations:

$$\mathbf{LOGSPACE} \subseteq \mathbf{P} \subseteq \mathbf{ZPP} \subseteq \left\{ \begin{array}{c} \mathbf{R} \subseteq \mathbf{NP} \\ (\mathbf{R} \cup \text{co-}\mathbf{R}) \subseteq \mathbf{BPP} \\ \text{co-}\mathbf{R} \subseteq \text{co-}\mathbf{NP} \end{array} \right\} \subseteq \mathbf{PP} \subseteq \mathbf{PSPACE}.$$

None of the inclusions is known to be proper, except that $\mathbf{LOGSPACE} \subsetneq \mathbf{PSPACE}$. Relativization with respect to a random oracle A yields the following greatly sharpened relations, with probability 1:

$$\mathbf{LOGSPACE}^A \subsetneq \left\{ \begin{array}{c} \mathbf{P}^A = \mathbf{ZPP}^A \\ = \\ \mathbf{R}^A = \mathbf{BPP}^A \end{array} \right\} \subsetneq \left\{ \begin{array}{c} \mathbf{NP}^A \\ \neq \\ \text{co-}\mathbf{NP}^A \end{array} \right\} \subsetneq \mathbf{PP}^A \subsetneq \mathbf{PSPACE}^A.$$

Relativization with respect to a random permutation function π, instead of the random oracle A, yields all these results and, in addition, $\mathbf{P}^\pi \subsetneq \mathbf{NP}^\pi \cap \text{co-}\mathbf{NP}^\pi$ with probability 1, which we have been unable to decide for a simple random oracle.

In view of the large number of classes that are separated by random oracle relativization, one might suppose that if there exists any oracle at all relative to which two classes are distinct, they they will be distinct relative to a random oracle. That this is not the case was shown by Hunt's [Hu] construction of an oracle X for which $\mathbf{P}^X \subsetneq \mathbf{ZPP}^X$, even though, by Theorem 5, these classes coincide with probability 1 relative to a random oracle. On the other hand, separations and identities that hold with probability 1 relative to a random oracle can generally also be demonstrated relative to particular recursive oracles.

Most of the random oracle results are obtained by using the oracle's randomness to force language recognition to depend on oracle queries, thereby in effect substituting number and size of queries for the more conventional (but theoretically intractable) dynamic computation resources of time and space. Thus, there is no immediate prospect of proving similarly sharp results in the absence of an oracle. On the other hand, random oracles by their very structurelessness appear more benign and less likely to distort the relations among complexity classes than the oracles traditionally used in complexity theory and recursive function theory, which are usually designed expressly to help or frustrate some class of computations. This suggests that statements that hold with probability 1 for languages relativized to a random oracle A are also true in the unrelativized case $A = \varnothing$.

To formalize this conjecture, the universe of appropriate statements needs to be defined. In particular, one wishes to include statements such as $\mathbf{P}^A \neq \mathbf{NP}^A$, $\mathbf{P}^A = \mathbf{BPP}^A$, and $\exists S$ ($S \in \mathbf{NP}^A$ and S is $\mathbf{P}^A$-immune), while excluding such incompletely relativized statements as $\mathbf{P} = \mathbf{P}^A$, or "$A$ is recursive."

Since the languages of interest in relativized complexity theory are uniformly A-recursive (i.e., recognizable by Turing machines that halt for all oracles and inputs), they may be referred to by the Gödel numbers of their characteristic function, and a k-adic relativized relation among such languages may be represented by a denumerable set of k-tuples of Gödel numbers of languages obeying the relation.

DEFINITION: A natural number i is a *uniform index* if the function φ_i^A is total and zero-one valued for all oracles A.

In this definition, φ_i^A, as usual, denotes the function computed by the ith Turing machine with oracle A. Without loss of generality, these Turing machines may be taken to be deterministic machines with no time or space bound, since nondeterminism, probabilism, and uniform (i.e., oracle-independent) time or space bounds can be incorporated implicitly by appropriate choice of the index i. A relativized language class such as $\mathbf{NP}^A$ (or equivalently the monadic relation $L \in \mathbf{NP}^A$) may now be formally defined as an indexed collection of A-parameterized languages invariant under appropriate group operations.

DEFINITION. Let I be a set of uniform indices. The A-parameterized class of languages $\mathbf{C}_I^A = \{\{x: \varphi_i^A(x) = 1\}: i \in I\}$ indexed by members of I is an *acceptable relativized class* iff:

1) for every oracle A, the class $\mathbf{C}_I^A$ is invariant under p^A-isomorphism [BH]; i.e., if f is a 1 : 1 onto function such that both f and f^{-1} are computable in polynomial time with oracle A, and if L is any language, then $L \in \mathbf{C}_I^A$ iff $f(L) \in \mathbf{C}_I^A$;

2) the class $\mathbf{C}_I^A$ is invariant under polynomial time Turing equivalences [La], [LLS] of the oracle set; i.e., if $B \in \mathbf{P}^A$ and $A \in \mathbf{P}^B$, then $\mathbf{C}_I^A = \mathbf{C}_I^B$.

Using this definition, it is not difficult to find index sets I for the language classes $\mathbf{P}^A$, $\mathbf{BPP}^A$, $\mathbf{NP}^A$, co-$\mathbf{NP}^A$, $\mathbf{PP}^A$, and $\mathbf{PSPACE}^A$. Other p^A-invariant classes generable in this manner are the class of finite languages and the class of languages with exactly k members, $k = 0, 1, 2$, etc. It is not clear, however, that more complicated classes such as $\{S: S \text{ is } \mathbf{NP}^A\text{-complete}\}$ and $\{S: S \in \mathbf{NP}^A \text{ and } S \text{ is } \mathbf{P}^A\text{-immune}\}$ can be generated by a single set of indices. To handle such cases, higher order relativized relations appear necessary.

DEFINITION: Let J be a set of ordered k-tuples of uniform indices. The A-parameterized class $\mathbf{R}_J^A$ of k-tuples of languages indexed by the members of J is an *acceptable relativized relation* iff:

1) for every oracle $\mathbf{A}$, the relation $\mathbf{R}_J^A$ is invariant under p^A-isomorphism: i.e., if f is a p^A-isomorphism, and $\langle L, M, \cdots, Q\rangle$ is a k-tuple of languages, then $\langle L, M, \cdots, Q\rangle \in \mathbf{R}_J^A$ iff $\langle f(L), f(M), \cdots, f(Q)\rangle \in \mathbf{R}_J^A$;

2) $\mathbf{R}_J^A$ is invariant under polynomial time Turing equivalences of the oracle set.

Among the important dyadic relations are language equality and complementation ($L = M$ and $L = \bar{M}$) and reducibilities such as the relativized Turing reducibility $\leq^A$, whose index set is the union over k of pairs $\langle i, j\rangle$ such that for all A, $\varphi_i^A = \varphi_k^{L(i,A) \text{ join } A}$, where $L(j, A)$ denotes the language whose characteristic function is φ_j^A. An important refinement of $\leq^A$, obtained by restricting the index k to polynomial time bounded machines, is the reducibility $\leq^{\mathbf{P},A}$, which holds between two languages L and M iff they are uniformly A-recursive and L is uniformly recognizable in polynomial time with oracle M **join** A. Notice that simple Turing reducibility (or its polynomial refinement), in which the oracle for the reduction is M rather than M **join** A, is not an acceptable relativized relation, because it is not invariant under p^A-isomorphism for typical A. In the present context of full relativization, $\mathbf{NP}^A$-completeness should be taken to mean completeness with respect to an invariant reducibility such as $\leqq^{\mathbf{P},A}$.

Intersection and union of languages may be expressed by acceptable triadic relations, e.g.,

$$\mathbf{R}_{\cap}^{A} = \{\langle L, M, N\rangle: L, M, \text{ and } N \text{ are uniformly } A\text{-recursive and } L = M \cap N\}.$$

The subset relation $L \subseteq M$, used for example in defining $\mathbf{P}^A$-immunity, may be expressed by quantifying the above triadic relation as $\exists_N L = M \cap N$, where the bound variable N, like the free variables L and M, range over uniformly A-recursive languages.

With the notion of acceptable relativized classes and relations thus delimited, it is easy to define a broad class of statements to which the random oracle hypothesis may reasonably be expected to apply.

DEFINITION. The A-parameterized statement S^A is an *acceptable relativized statement* if it is definable in quantificational logic using

bound variables denoting uniformly A-recursive languages;
acceptable relativized relations on these variables;
the logical operators AND, OR and NOT.

The oracle set A and the relations' index sets I, J, etc., appear only as parameters, and cannot be acted on by any of the quantifiers or relations. Note that acceptable relativized statements, by virtue of their invariance under **P**-Turing equivalence of the oracle set, can only have probability 0 or 1. The random oracle hypothesis may now be stated:

Random Oracle Hypothesis. Let S^A be any acceptable relativized statement. The corresponding unrelativized statement $S^{\varnothing}$ is true if and only if S^A is true with probability 1 when A is chosen randomly.

In particular, since $\mathbf{NP}^A \neq \mathbf{P}^A$ and $\mathbf{P}^A = \mathbf{BPP}^A$ are acceptable statements that are true with probability 1, the random oracle hypothesis would imply $\mathbf{P} = \mathbf{BPP} \neq \mathbf{NP}$. We believe that this hypothesis, or a similar but stronger one, captures a basic intuition of the pseudorandomness of nature from which many apparently true complexity results follow. The random oracle hypothesis could be strengthened by attempting to include non A-recursive languages, by relaxing the invariances required of acceptable relations (e.g., invariance under logspace^A-isomorphism rather than p^A-isomorphism), and by asserting further that results true relative to a random *permutation* are true absolutely.

The random oracle hypothesis does not deny all differences between no oracle and a random oracle: clearly, machines equipped with a random oracle can recognize nonrecursive sets, while unaided machines cannot. Similarly, there exist sets which are immune absolutely, but, with probability 1, not immune relative to a random oracle [Ba]. However, all known differences of this sort concern partially relativized properties; in a fully relativized setting the differences disappear, since (for example) the nonrecursive sets recognized by random oracle machines are all A-recursive.

In view of the great amount of effort expended in unsuccessful attempts to prove apparently true statements such as $\mathbf{P} \neq \mathbf{NP}$, and $\mathbf{NP} \neq \mathbf{PSPACE}$, it is possible that these statements may be independent of other commonly accepted axioms of arithmetic and set theory. The random oracle hypothesis is thus a plausible candidate for a new axiom.

The random oracle hypothesis would be proved if an easily computable substitute for ξ (or A or π) could be found, e.g., a function ψ that requires little time and space to evaluate, but is pseudorandom in the sense that the inevitable correlations among $\psi(x)$ for different x cannot be exploited without large amounts of time and space. The search for this kind of pseudrorandomness is related to the search (also quite unsuccessful so

far) for a provably almost-everywhere moderately-hard-to-compute function [Rb], [GB].

It is not hard to invent polynomially computable functions that *appear* pseudorandom; the difficulty arises in proving them so. For example, the function $\psi(x) =$ [the third $|x|$ bits of $\cos(x)$] appears to have the statistical properties of the ξ function, and finding inverse images appears to require an exponential search, but no one knows how to prove this.

[*Remark.* In defining ψ it is necessary to skip an increasing number of early bits of $\cos(x)$ because these early bits are more often 1 than zero, owing to the cosine's turning points at ± 1. The bias in the kth bit is of order $2^{-k/2}$; thus, the sequence of $(2|x|+1)$st bits of $\cos(x)$, for $x = 1, 2, 3, \cdots$ should have a bias decreasing as x^{-1}, rendering it statistically indistinguishable from a random Bernoulli sequence. Other more complicated deviations from pseudorandomness, e.g., those arising from the nonuniform distribution of the difference between $\cos(x)$ and $\cos(x+1)$, would presumably be obliterated in the same way.]

By giving up the requirement that a function or set be easy to compute, one gains the ability to prove that specific sets are pseudorandom, i.e., that they have the properties of a generic random oracle with respect to bounded computation. A natural but nonrecursive example would be an *algorithmically random* set such as the bit sequence of Chaitin's real number ω [Ch], which expresses the halting probability of a universal Turing machine with random input. The set $\{x :$ the xth digit of ω is a 1$\}$ is in class Δ_2 of the arithmetical hierarchy, but passes all computable tests of randomness, and could be substituted for the generic A in all the above theorems. Meyer and McCreight [MM], using a priority construction, have exibited a recursive pseudorandom set, recognizable in quadruple exponential space but appearing random with respect to all test sets recognizable in double exponential space. Similar constructions should yield a proof of a weak analogue of the random oracle hypothesis, viz., that if an acceptable relativized statement S^A is true with probability 1 for random A, then it is also true for some recursive A. The converse, of course, does not hold, since many relativized statements [BGS] are known to be true for some recursive oracles but false for others.

Acknowledgments. The authors thank Larry Carter, Mark Wegman, George Markowsky, Dexter Kozen and Gilles Brassard for numerous helpful discussions, Gregory Chaitin for helping to formulate the method of using a random oracle to simulate probabilistic computation, and Robert Solovay for criticizing a preliminary version of the proof of Theorem 1.

REFERENCES

[Ad] L. ADLEMAN, *Two theorems on random polynomial time*, Proceedings of the 19th IEEE Symposium on the Foundations of Computer Science, Ann Arbor, MI, 1978, pp. 75–83.

[AM] L. ADLEMAN AND K. MANDERS, *Reducibility, Randomness, and Intractability*, Proceedings of the 9th ACM Symposium on the Theory of Computing, 1977, pp. 151–153.

[An] D. ANGLUIN, *On counting problems and the polynomial time hierarchy*, Theoret. Comput. Sci., to appear.

[Ba] YA. M. BARZDIN', *On computability by probabilistic machines*, Dokl. Akad. Nauk SSSR, 189 (1969), pp. 699–702, = Soviet Math. Dokl., 10 (1969), pp. 1464–1467.

[Br] G. BRASSARD, *Relativized cryptography*, Proceedings of the 20th IEEE Symposium on the Foundations of Computer Science, San Juan, Puerto Rico, 1979, pp. 383–391.

[BGS] T. BAKER, J. GILL AND R. SOLOVAY, *Relativizations of the $P = ?\ NP$ question*, this Journal, 4 (1975), pp. 431–442.

[BH] L. BERMAN AND J. HARTMANIS, *On isomorphisms and densities of NP and other complete sets*, this Journal, 6 (1977), pp. 305–322.

[BS] T. BAKER AND A. SELMAN, *A second step toward the polynomial hierarchy*, Proceedings of the 17th IEEE Symposium on the Foundations of Computer Science, 1976, pp. 71–75.

[Ch] G. CHAITIN, *A theory of program size formally identical to information theory*, J. Assoc. Comput. Mach., 22 (1975), pp. 329–340.

[DH] W. DIFFIE AND M. HELLMAN, *New directions in cryptography*, IEEE Trans. Inform. Theory IT-22 (1976), pp. 644–654.

[Fe] W. FELLER, *An Introduction to Probability Theory and its Applications*, John Wiley, New York, 1957, Chapter 7.

[Fe2] W. FELLER, *An Introduction to Probability Theory and its Applications, volume II*, John Wiley, New York, 1971, Chapter 4.

[GB] J. GILL AND M. BLUM, *On almost everywhere complex recursive functions*, J. Assoc. Comput. Mach., 21 (1974), pp. 425–435.

[Gi] J. GILL, *Computational complexity of probabilistic Turing machines*, this Journal, 6 (1977), pp. 675–695.

[Hu] J. W. HUNT, *Topics in probabilistic complexity*, Ph.D. dissertation, Department of Electrical Engineering, Stanford University, Stanford CA, 1978, p. 58.

[KM] D. KOZEN AND M. MACHTEY, *On relative diagonals*, J. Comput. System Sci., to appear.

[La] R. E. LADNER, *On the structure of polynomial time reducibility*, J. Assoc. Comput. Mach., 22 (1975), pp. 151–171.

[LL] R. E. LADNER AND N. A. LYNCH, *Relativization of questions about log space computability*, Math. Systems Theory, 10 (1976), pp. 19–32.

[LLS] R. E. LADNER, N. A. LYNCH AND A. L. SELMAN *A comparison of polynomial time reducibilities*, Theoret. Comput. Sci., 1 (1975), pp. 103–123.

[dLMSS] K. DE LEEUW, E. F. MOORE, C. E. SHANNON AND N. SHAPIRO, *Computability by probabilistic machines*, in Automata Studies, An. Math. Studies No. 34, Princeton University Press, Princeton, NJ, 1956, pp. 182–212.

[Ma] STEPHEN R. MAHANEY, *Sparse complete sets for NP: solution of a conjecture by Berman and Hartmanis*, Proceedings of the 21st IEEE Symposium on the Foundations of Computer Science, Syracuse, New York, 1980, to appear.

[Me] K. MEHLHORN, *On the size of sets of computable functions*, Proceedings of the 14th IEEE Symposium on Switching and Automata Theory, Iowa City, IO, 1973, pp. 190–196.

[MM] A. R. MEYER AND E. M. MCCREIGHT, *Computability complex and pseudorandom zero-one valued functions*, in Theory of Machines and Computations, Z. Kohavi and Azaria Paz, eds., Academic Press, New York, 1971, pp. 19–42.

[MS] A. MEYER AND L. STOCKMEYER, *The equivalence problem of regular expressions with squaring requires exponential time*, Proceedings of the 13th IEEE Symposium on Switching and Automata Theory, 1972, pp. 125–129.

[Ra] C. RACKOFF, *Relativized questions involving probabilistic algorithms*, Proceedings of 10th ACM Symposium on Theory of Computing, San Diego, CA, 1978, pp. 338–342.

[Rb] M. O. RABIN, *Degree of Difficulty of Computing a Function and a Partial Ordering of Recursive Sets*, Tech. Rep. 2, Hebrew Univ., Jerusalem, Israel, 1960.

[Ro] H. ROGERS, JR. *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, New York, 1967.

[Si] J. SIMON, *On Some Central Problems in Computational Complexity*, Tech. Rep TR 75-224, Dept. of Computer Science, Cornell University, Ithaca, NY, 1975.