On The Relativized Power of Additional Accepting Paths

Richard Beigel *
Department of Computer Science
The Johns Hopkins University
Baltimore, MD 21218

Abstract

The class UP is the class of languages in NP that are accepted by machines with at most one accepting path on each input. We define $U_{k(n)}P$ as the class of languages in NP that are accepted by machines with at most $k(n)$ accepting paths on each input of length n. Obviously

$$P \subseteq UP \subseteq U_{k(n)}P \subseteq U_{k(n)+1}P \subseteq \mathrm{FewP} \subseteq NP$$

for every polynomial $k(n) \geq 2$, where FewP is the class of languages in NP that are accepted by machines with a polynomial-bounded number of accepting paths on each input. It is an open question whether any of the containments is proper. A proper containment would imply $P \neq NP$. We consider the relativized class $U_{k(n)}P^A$, and we construct an oracle A such that

$$P^A \subset UP^A \subset U_{k(n)}P^A \subset U_{k(n)+1}P^A \subset \mathrm{FewP}^A \subset NP^A$$

for every polynomial $k(n) \geq 2$. Relative to a random oracle A, we prove that

$$P^A \subset UP^A \subset U_kP^A \subset U_{k+1}P^A \subset \mathrm{FewP}^A$$

for every constant $k \geq 2$, and we prove similar separations when $k(n)$ is not a constant. Previously, Hartmanis and Hemachandra have shown that it is not possible to separate P^A from UP^A via uniform techniques; therefore the methods we use are necessarily novel. Unlike previous random oracle results, we use nonuniform techniques and nontrivial probability theory in order to prove the separations.

Note: At the Structures '89 conference, we announced that there was an error in the random oracle proofs. Most of the results can be salvaged by applying techniques developed by Bennett and Gill [4] to prove separations with immunity. At the conference, we gave a correct proof that relative to a random oracle A there is a tally set in $UP^A - P^A$. That proof extends to $U_{k+1}P^A - U_kP^A$ for all $k \geq 2$. Once we determine the situation when $k(n)$ is not a constant, we will publish a revision.

1. Introduction

Nondeterministic models of computation with a limited number of accepting paths have been considered for several years. The class UP was defined by Valiant [12].

Definition 1 UP is the class of languages L for which there exists a nondeterministic polynomial-time Turing machine M such that M accepts L and M has at most one accepting path for each input.

Grollman and Selman showed that $P \neq UP$ if and only if one-way functions exist [8]. The class FewP was defined in [1, 6].

Definition 2 FewP is the class of languages L for which there exists a nondeterministic polynomial-time Turing machine M such that M accepts L and M has a polynomial-bounded number of accepting paths for each input.

In [7], Geske and Grollman constructed a single oracle C such that

$$P^C \subset UP^C \subset NP^C$$

and another oracle D such that

$$\mathrm{FewP}^D \subset NP^D.$$

That $P^A \neq UP^A$ for almost all oracles has been observed in various forms by Blum, Impagliazzo, and Rudich [11], in addition to ourself. In this paper, we construct a hierarchy of languages between UP^A and NP^A that is proper for at least one choice of the oracle A. Furthermore an initial segment of that hierarchy is proper for almost all oracles.

*This research was supported in part by grant number CCR-8808949 from the National Science Foundation.

Definition 3 $\mathrm{U}_{k(n)}\mathrm{P}$ is the class of languages L for which there exists a nondeterministic polynomial-time Turing machine M such that M accepts L and M has at most $k(n)$ accepting paths for each input of length n.

When $k(n)$ is a constant function, we write $\mathrm{U}_k\mathrm{P}$.

Although the oracle construction technique in Section 2. is similar to Geske and Grollman's, we use a counting technique from [3] to streamline the proof. Moreover, the random oracle techniques used in Section 3. differ from prior techniques in two novel ways. First, our techniques are not uniform. In order to separate C^A from D^A, previous techniques constructed a test language L^A such that $L^A \in C^A$ for all oracles A, but $L^A \notin D^A$ for almost all oracles A. In this paper, we construct a test language L^A that is in C^A for almost all oracles, but not all. Second, we use probability theory in a nontrivial way. For each machine M, previous techniques chose a single point x, and showed that $M^A(x) \neq L^A(x)$ for a constant fraction of all oracles. In this paper, we choose infinitely many points. We show that M^A differs from L^A on the nth point with probability proportional to $1/n$ or larger. Then we use the Borel-Cantelli lemma to show that $L^A \notin D^A$ for almost all oracles A.

As mentioned, the constructions in Section 3. are not uniform. In Section 4., we show that a uniform construction would prove $\mathrm{P} \neq \mathrm{NP}$. This builds on and unifies work in [5, 9].

2. An Oracle Separating $\mathrm{U}_{k(n)}\mathrm{P}^A$ from $\mathrm{U}_{k(n)+1}\mathrm{P}^A$

In this section we construct an oracle separating $\mathrm{U}_{k(n)}\mathrm{P}^A$ from $\mathrm{U}_{k(n)+1}\mathrm{P}^A$ for $k(n) \ll 2^{n/2}$. As a corollary, we have

$$\mathrm{P}^A \subset \mathrm{UP}^A \subset \mathrm{FewP}^A \subset \mathrm{NP}^A.$$

Because a single elegant counting argument accounts for all of the separations, this can be regarded as an extension and simplification of the results in [7].

Proposition 4 *There exists an oracle A such that $\mathrm{P}^A \subset \mathrm{UP}^A$.*

Proof: Such an oracle can be constructed via an easy initial segment argument. In fact, Baker, Gill and Solovay's oracle such that $\mathrm{NP}^A \neq \text{co-NP}^A$ does the trick [2]. ∎

We write (a.e. n)$[Q(n)]$ to denote that the predicate $Q(n)$ is satisfied by almost all n (all sufficiently large n). We say that a polynomial is *positive* if its lead coefficient is positive. We write $f(n) \ll g(n)$ if for every positive polynomial p we have (a.e. n)$[f(n) < g(n)/p(n)]$.

Proposition 5 *Let $k(n)$ be a function such that $1 \leq k(n) \ll 2^{n/2}$. Then for all positive polynomials p, for almost all n, we have*

$$k(n)\left(2^n + \binom{2^n}{k(n)+1}\right) < 2^n + 2^n\binom{2^n - p(n)}{k(n)}.$$

Proof: We analyze both sides of the claimed inequality, up to highest-order terms. Let $L(n)$ denote the left-hand side, and let $R(n)$ denote the right-hand side. Thus $L(n) = k(n)\binom{2^n}{k(n)+1}(1+O(2^{-n}))$ and $R(n) = 2^n\binom{2^n-p(n)}{k(n)}(1+O(2^{-n}))$, so

$$\begin{aligned}
R(n)/L(n) & \\
&= \tfrac{(k(n)+1)!\cdot 2^n\cdot((2^n-p(n))\cdots(2^n-p(n)-k(n)-1))}{k(n)!\cdot k(n)\cdot(2^n\cdots(2^n-k(n)))}\cdot(1\pm O(2^{-n}))\\
&= \tfrac{(k(n)+1)\cdot((2^n-p(n))\cdots(2^n-p(n)-k(n)-1))}{k(n)\cdot((2^n-1)\cdots(2^n-k(n)))}\cdot(1\pm O(2^{-n}))\\
&= \left(1+\tfrac{1}{k(n)}\right)\cdot\left(\tfrac{2^n-p(n)}{2^n-1}\right)\cdots\left(\tfrac{2^n-p(n)-k(n)-1}{2^n-k(n)}\right)\cdot(1\pm O(2^{-n}))\\
&\geq \left(1+\tfrac{1}{k(n)}\right)\cdot\left(\tfrac{2^n-p(n)-k(n)-1}{2^n-k(n)}\right)^{k(n)}\cdot(1\pm O(2^{-n}))\\
&= \left(1+\tfrac{1}{k(n)}\right)\cdot\left(1-\tfrac{p(n)-1}{2^n-k(n)}\right)^{k(n)}\cdot(1\pm O(2^{-n}))\\
&\geq \left(1+\tfrac{1}{k(n)}\right)\cdot\left(1-\tfrac{(p(n)-1)k(n)}{2^n-k(n)}\right)\cdot(1\pm O(2^{-n}))\\
&\quad \text{for almost all } n,\\
&\quad \text{because } (1-x)^y \geq 1-xy \text{ whenever } 0\leq x\leq 1\leq y\\
&> 1 \qquad \text{for almost all } n,
\end{aligned}$$

because $\frac{(p(n)-1)k(n)}{2^n-k(n)}$ is negligible compared to $\frac{1}{k(n)}$. Thus, we have shown that (a.e. n)$[R(n)/L(n) > 1]$, so (a.e. n)$[L(n) < R(n)]$. ∎

Theorem 6 *Let $k(n)$ be a function such that $1 \leq k(n) \ll 2^{n/2}$. There exists an oracle A such that*

$$\mathrm{U}_{k(n)}\mathrm{P}^A \subset \mathrm{U}_{k(n)+1}\mathrm{P}^A.$$

Proof: This construction uses the initial segment method and the technique of reversing the order of summation described at length in [3]. We write $A^{[n]}$ to denote the set of all strings of length n belonging to the set A. Define

$$L^A = \{0^n : A^{[n]} \neq \emptyset\}.$$

Let $|S|$ denote the cardinality of a set S. We will construct A so that $|A^{[n]}| \leq k(n)+1$; therefore $L^A \in \mathrm{U}_{k(n)+1}\mathrm{P}^A$ via the obvious algorithm. We will construct $A = \lim_i A_i$ via the initial segment method so as to guarantee that

$$L^A \notin \mathrm{U}_{k(n)}\mathrm{P}^A.$$

Let $p_i(n)$ denote the polynomial $n^i + i$.

Stage 0: Let $A_0 = \emptyset$. Let $n = 0$.

Stage $i > 0$: Choose $n' > p_{i-1}(n)$ such that

$$k(n')\left(2^{n'} + \binom{2^{n'}}{k(n')+1}\right) < 2^{n'} + 2^{n'}\binom{2^{n'} - p_i(n')}{k(n')}. \quad (1)$$

(This is possible by Proposition 5.) Let β denote the set of all $B \subseteq \Sigma^{n'}$ such that $|B|$ is equal to 0, 1, or $k(n') + 1$. As justified below, choose $B \in \beta$ such that one of the following two conditions holds:

(i) $|\mathrm{ACCEPT}(N_i^{A_i \cup B}, 0^{n'})| > k(n')$, or

(ii) $B = \emptyset \iff |\mathrm{ACCEPT}(N_i^{A_i \cup B}, 0^{n'})| \neq 0$.

Let $A_{i+1} = A_i \cup B$. Let $n = n'$.

Obviously the construction guarantees that $L^A \notin \mathrm{U}_{k(n)}\mathrm{P}^A$. It remains to show that the construction is possible. Suppose that stage i cannot be accomplished. We will obtain a contradiction by considering the number of accepting paths of $N_i^{A_i \cup B}$ on input $0^{n'}$, and summing over all choices of B. We write TA to denote that total. Let B denote an element of β. If $|B|$ is equal to 0 then by assumption,

$$|\mathrm{ACCEPT}(N_i^{A_i \cup B}, 0^{n'})| = 0.$$

If $|B|$ is equal to 1 or $k(n') + 1$ then by assumption,

$$|\mathrm{ACCEPT}(N_i^{A_i \cup B}, 0^{n'})| \leq k(n').$$

Therefore

$$\begin{aligned} \mathrm{TA} &= \sum_{B \in \beta} |\mathrm{ACCEPT}(N_i^{A_i \cup B}, 0^{n'})| \\ &\leq k(n') \cdot \left(\binom{2^{n'}}{1} + \binom{2^{n'}}{k(n')+1}\right). \end{aligned}$$

Let ρ denote an accepting computation of N_i on input $0^{n'}$ for some choice of $B \in \beta$ with $|B| = 1$. Because at least $2^{n'} - p_i(n')$ strings are not queried on each path, ρ must be an accepting computation for at least $\binom{2^{n'} - p_i(n')}{k(n')}$ different choices of $B \in \beta$ with $|B| = k(n') + 1$. There are exactly $2^{n'}$ choices of $B \in \beta$ with $|B| = 1$. Each such choice of B must correspond to a distinct accepting computation ρ, because each accepting computation must verify that $B \neq \emptyset$ (otherwise condition (i) is satisfied by $B = \emptyset$). Therefore there are at least $2^{n'}$ distinct paths, each of which accepts for at least $\binom{2^{n'} - p_i(n')}{k(n')} + 1$ different choices of $B \in \beta$. Thus

$$\mathrm{TA} \geq 2^{n'} \cdot \left(\binom{2^{n'} - p_i(n')}{k(n')} + 1\right).$$

However, this contradicts (1), so we conclude that the construction is possible. ∎

Corollary 7 *For any countably infinite sequence of functions such that $1 \leq k_1 < k_2 < \ldots \ll 2^{n/2}$, there is an oracle A such that*

$$\mathrm{P}^A \subset \mathrm{U}_{k_1(n)}\mathrm{P}^A \subset \mathrm{U}_{k_2(n)}\mathrm{P}^A \subset \cdots \subset \mathrm{NP}^A.$$

Proof: We can combine the initial segment arguments from the previous proof for countably many functions k. ∎

Corollary 8 *There is an oracle A such that*

$$\mathrm{U}_{k(n)}\mathrm{P}^A \subset \mathrm{U}_{k(n)+1}\mathrm{P}^A$$

for all total recursive functions $k(n) \ll 2^{n/2}$.

Proof: This is similar to the preceding proof. Note that the set of total recursive functions is countable. ∎

Results similar to the following corollary have appeared in [7].

Corollary 9 *There is an oracle A such that*

$$\mathrm{P}^A \subset \mathrm{UP}^A \subset \mathrm{FewP}^A \subset \mathrm{NP}^A.$$

3. Separations Relative to a Random Oracle

In the preceding section we constructed a single oracle A such that $\mathrm{U}_{k(n)}\mathrm{P}^A \subset \mathrm{U}_{k(n)+1}\mathrm{P}^A$. In this section we show for almost all oracles A that

$$\mathrm{P}^A \subset \mathrm{UP}^A \subset \mathrm{U}_k\mathrm{P}^A \subset \mathrm{U}_{k+1}\mathrm{P}^A \subset \mathrm{FewP}^A,$$

and that

$$\mathrm{U}_{k(n)}\mathrm{P}^A \subset \mathrm{U}_{ck(n)}\mathrm{P}^A$$

for every constant $c > 1$, whenever $k(n)$ is a time-constructible function such that $\ln n \leq k(n) \leq (k-1)n \ln 2 - \ln \ln n$.

We assume that the reader is familiar with the random oracle techniques developed by Bennett and Gill in [4]. We write $\mu(X)$ to denote the measure of a set X of oracles.

The first separation, $\mathrm{P}^A \subset \mathrm{UP}^A$, is the easiest to establish. We begin by presenting an elegant proof due to Manuel Blum, Russell Impagliazzo, and Steven Rudich. Then we will present a second proof, which we will generalize to prove the other separations.

Theorem 10 *For almost all oracles A,*

$$\mathrm{P}^A \subset \mathrm{UP}^A.$$

Proof: This proof is due to Blum, Impagliazzo, and Rudich [11]. We say that a one-one function f is *one-way*^{A} if f is computable in polynomial time relative to oracle A, the length of $f(x)$ is at least $|x|^{1/i}$ for some constant i, and f^{-1} is not computable in polynomial time relative to oracle A. We show that one-way^A functions exist relative to a random oracle A. We define a function f (polynomial-time computable relative to A) that maps a string x of length n onto a sequence of n^2+2n random bits provided by the oracle A:

$$f(x) = A(x10)A(x10^2)\cdots A(x10^{|x|^2+2|x|}).$$

The probability that two particular distinct strings x and y of length n have the same image is 2^{-n^2-2n}. Because there are 2^n strings of length n, the probability that there exist two distinct strings of length n with the same image is less than $2^{2n}2^{-n^2-2n} = 2^{-n^2}$. Therefore the probability that there exist two distinct strings with the same image is less than

$$\sum_{n\geq 1} 2^{-n^2} < 3/4.$$

Thus f is one-one with probability greater than $1/4$. A routine argument as in [4] shows that f cannot be inverted in polynomial time relative to almost all A. Thus one-way^A functions exist relative to A with probability greater than $1/4$. By the 0/1 law for tail events (see [4]), one-way^A functions exist relative to A with probability 1. In [8], it is shown that one-way functions exist iff $\mathrm{P} \neq \mathrm{UP}$. Since their proof relativizes, $\mathrm{P}^A \neq \mathrm{UP}^A$ for almost all oracles A. ∎

Now we present a second proof. Although it is more complicated than Blum, Impagliazzo, and Rudich's, the techniques will help us to separate $\mathrm{U}_{k-1}\mathrm{P}^A$ from $\mathrm{U}_k\mathrm{P}^A$ for $k \geq 2$. We begin by stating a combinatorial inequality. We will use it later in order to construct an NP^A predicate that has exactly j witnesses infinitely often for each j such that $0 \leq j \leq k-1$ with probability 1, but has k witnesses infinitely often with probability 0.

Let e denote the base of the natural logarithms. We write $f(n) \approx g(n)$ if $\lim_{n\to\infty} f(n)/g(n) = 1$.

Proposition 11 *Let E be a set consisting of $p2^n$ events each occurring independently with probability 2^{-n}. Let j be such that $j \ll 2^n$ and $j^2 \ll p2^n$. Then the probability that exactly j events occur is $\approx e^{-p}p^j/j!$.*

Proof: The probability that exactly j events occur is

$$\binom{p2^n}{j}(2^{-n})^j(1-2^{-n})^{p2^n-j}.$$

First we approximate $\binom{p2^n}{j}$. It is easy to see that

$$\frac{1}{j!}(p2^n)^j(1-\frac{j}{p2^n})^j \leq \binom{p2^n}{j} \leq \frac{1}{j!}(p2^n)^j.$$

Because j is small, we have

$$(1-\frac{j}{p2^n})^j \approx 1 - \frac{j^2}{p2^n} \approx 1.$$

Therefore

$$\binom{p2^n}{j} \approx \frac{1}{j!}(p2^n)^j.$$

Because $(1-1/x)^x$ approaches e^{-1} for large x,

$$(1-2^{-n})^{p2^n-j} \approx e^{-p+j2^{-n}} \approx e^{-p}.$$

Thus

$$\binom{p2^n}{j}(2^{-n})^j(1-2^{-n})^{p2^n-j} \approx ((p2^n)^j/j!)2^{-nj}e^{-p} = e^{-p}p^j/j!.$$

∎

We write $f(n) = \theta(g(n))$ if $f(n) = O(g(n))$ and $g(n) = O(f(n))$.

Lemma 12 *Let k be a constant, and let $p = n^{-1/k}$. Let E be a set consisting of $p2^n$ events each occurring independently with probability 2^{-n}. Then the probability that at least k events occur is $\theta(n^{-1})$, and the probability that at least $k+1$ events occur is $\theta(n^{-(k+1)/k})$.*

Proof: By the preceding lemma, the probability that exactly j events occur is $\approx e^{-p}p^j/j!$. Therefore the probability that at least j events occur is

$$\approx e^{-p}\sum_{i\geq j} p^i/i!.$$

Since $p = o(1)$, $e^{-p} = \theta(1)$. The sum is dominated by its first term, which is $\theta(p^j)$. Thus the probability that at least j events occur is $\theta(1)\cdot\theta(p^j) = \theta(p^j)$. Letting $p = n^{-1/k}$ and letting j equal successively k and $k+1$ we obtain the claimed result. ∎

We write $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$. Otherwise $f(n) = o(g(n))$.

Second Proof of Theorem 10: For each n we will do the following: We will define $2^n/n$ events each occurring independently with probability 2^{-n}. By

Lemma 12, the probability that at least one event occurs is $\theta(1/n)$, and the probability that at least two events occur is $\theta(1/n^2)$.

Because $\sum_{n\geq 1} n^{-1}$ diverges and the events are independent, the probability that at least one event occurs for infinitely many values of n is 1. Because $\sum_{n\geq 1} n^{-2}$ converges, the probability that at least two events occur for infinitely many values of n is 0 (by the Borel-Cantelli lemma).

It is helpful to consider the oracle as a sequence of bits. We define a test language RUNS^A as follows: Divide the oracle A into blocks such that the nth block has length $n\lceil 2^n/n\rceil$. Divide the nth block into $\lceil 2^n/n\rceil$ subblocks, each of length n. We say that a subblock is an event of length n in A if the subblock of A is a sequence of n 1's. Let

$$\mathrm{RUNS}^A = \{0^n : \text{there exists an event of length } n \text{ in } A\}.$$

RUNS^A belongs to NP^A via the obvious algorithm. By the preceding discussion, for almost all oracles A, there are only finitely many values of n for which there exists more than one event of length n in A. Thus, for almost all oracles A, RUNS^A is in UP^A, because we can handle the finite number of exceptions by table lookup.

(However, for almost all oracles A, there are infinitely many values of n for which there exists exactly one event of length n in A. This is the intuition for why $\mathrm{RUNS}^A \notin \mathrm{P}^A$ for almost all oracles A.) Let M^A be a P^A machine that attempts to accept RUNS^A. M^A errs if it accepts a string not belonging to RUNS^A, or if it rejects a string belonging to RUNS^A. We will show that M^A errs with probability at least $1/n$ on input 0^n, and that these errors are independent. Therefore, M^A errs at least once with probability 1. By Bennett and Gill's Lemma 1 [4], $\mathrm{RUNS}^A \notin \mathrm{P}^A$ for almost all A.

Let M^A run in time $q(n)$ on input of length n for all oracles A. Fix n and fix an input 0^n. Fix also $A^{[i]}$ for all $i \neq n$. We prove by contradiction that M^A errs on input 0^n with probability $\Omega(1/n)$; because the proof is valid regardless of the membership in A of strings whose length is not n, this error probability is independent of the errors on other inputs.

We say that a computational path *inspects* a subblock if the path queries at least one element of the subblock. Because M^A makes at most $q(n)$ queries, M^A inspects at most $q(n)$ subblocks. Since there are $\lceil 2^n/n\rceil$ subblocks, M^A inspects the ith subblock for randomly chosen i with probability at most $nq(n)/2^n$. By Lemma 12, the probability that there is more than one event is $O(1/n^2)$. Therefore the probability that M^A inspects an event is at most $nq(n)/2^n + O(1/n^2) = O(1/n^2) = o(1/n)$ for large n. Therefore, with probability $1 - o(1/n)$, M^A either (1) rejects or (2) accepts without inspecting an event. Therefore M^A either (1) rejects with probability $1 - o(1/n)$ or (2) accepts without inspecting an event with probability $\Omega(1/n)$. In the first case, M^A errs with probability $\Omega(1/n) - o(1/n) = \Omega(1/n)$ because $0^n \in \mathrm{RUNS}^A$ with probability $\Omega(1/n)$. In the second case M^A errs with probability $\Omega(1/n)\cdot\Omega(1)$ because $0^n \notin \mathrm{RUNS}^A$ with probability $\Omega(1)$. In either case M^A errs with probability $\Omega(1/n)$. ∎

Inverting a one-one function is like searching for a unique witness. Blum, Impagliazzo, and Rudich's proof attains its simplicity because it guarantees that infinitely many strings have pre-images, which is like guaranteeing that a UP test language is infinite. In our proof we could only expect to construct an infinite test language with probability one. We will generalize the second proof to separate $\mathrm{U}_{k-1}\mathrm{P}^A$ from $\mathrm{U}_k\mathrm{P}^A$.

Theorem 13 *Let $k \geq 2$. For almost all oracles A,*

$$\mathrm{U}_{k-1}\mathrm{P}^A \subset \mathrm{U}_k\mathrm{P}^A.$$

Proof: For each n we will do the following: Let $p = n^{-1/k}$. We will define $p2^n$ events each occurring independently with probability 2^{-n}. By Lemma 12, the probability that at least k events occur is $\theta(n^{-1})$, and the probability that at least $k+1$ events occur is $\theta(n^{-(k+1)/k})$.

Because $\sum_{n\geq 1} n^{-1}$ diverges and the events are independent, the probability that at least k events occur for infinitely many values of n is 1. Because $\sum_{n\geq 1} n^{-(k+1)/k}$ converges, the probability that at least $k+1$ events occur for infinitely many values of n is 0 (by the Borel-Cantelli lemma).

We define a test language RUNS^A as follows: Divide the oracle A into blocks such that the nth block has length $n\lceil p2^n\rceil$. Divide the nth block into $\lceil p2^n\rceil$ subblocks, each of length n. We say that a subblock is an event of length n in A if the subblock of A is a sequence of n 1's. Let

$$\mathrm{RUNS}^A = \{0^n : \text{there exists an event of length } n \text{ in } A\}.$$

RUNS^A is in NP^A via the obvious algorithm. By the preceding discussion, for almost all oracles A, there are only finitely many values of n for which there exist more than k events of length n in A. Thus, for almost all oracles A, RUNS^A is in $\mathrm{U}_k\mathrm{P}^A$, because we can handle the finite number of exceptions by table lookup.

(However, for almost all oracles A, there are infinitely many values of n for which there exists exactly k events of length n in A. This is the intuition for why $\mathrm{RUNS}^A \notin \mathrm{U}_{k-1}\mathrm{P}^A$ for almost all oracles A.) Let M^A be a $\mathrm{U}_{k-1}\mathrm{P}^A$ machine that attempts to accept RUNS^A. M^A errs if it accepts when it should reject, rejects when it should accept, or has more than $k-1$ accepting paths for some input. We will show that M^A errs with probability $\Omega(1/n)$ on input 0^n, and that these errors are independent. Therefore, M^A must err at least once with probability 1. By Bennett and Gill's Lemma 1 [4], $\mathrm{RUNS}^A \notin \mathrm{U}_{k-1}\mathrm{P}^A$ for almost all A.

Let M^A run in time $q(n)$ on input of length n for all oracles A. Fix n and fix an input 0^n. Fix also $A^{[i]}$ for all $i \neq n$. We assume that M^A errs with probability $o(p)$ and we will conclude that M^A errs with probability $\Omega(1/n)$. Because $p > 1/n$, this is a valid proof that M^A errs with probability $\Omega(1/n)$. Furthermore, the proof is valid regardless of membership in A of strings whose length is different from n, so this error probability is independent of the errors on other inputs.

We say that a computational path *inspects* a subblock if the path queries at least one element of the subblock.

Let $\mathcal{C}$ be the class of oracles A that contain exactly one event of length n. If M^A rejected for $\Omega(1)$ of all oracles A in $\mathcal{C}$ then M^A would err with probability $\Omega(\mu(\mathcal{C}))$. By Proposition 11, $\mu(\mathcal{C}) \approx e^{-p}p = \theta(p)$ because $p = o(1)$, so M^A would err with probability $\Omega(p)$. Therefore, M^A rejects for at most $o(1)$ of all oracles A in $\mathcal{C}$. Then M^A must have an accepting path for $1 - o(1)$ of all oracles A in $\mathcal{C}$. Let $\mathcal{C}_i$ be the class of oracles A such that the ith subblock of length n is the unique event in A. Then, for randomly chosen i, M^A must have an accepting path ρ_i for $1 - o(1)$ of all oracles $A \in \mathcal{C}_i$.

If that path ρ_i accepts without inspecting the ith subblock, then it must also accept if the ith subblock has any of the $2^n - 1$ values constituting a nonevent. Let x be the probability that ρ_i does not inspect the ith subblock. Then M^A accepts incorrectly with probability

$$(2^n - 1)x\mu(\mathcal{C}_i) = (2^n - 1)x\theta(1/(2^n)) = \theta(x),$$

so $x = o(p)$. Therefore the probability that ρ_i does not inspect the ith subblock is $o(p)$.

Let $A \in \mathcal{C}_i$. Each path makes at most $q(n)$ queries. Therefore, for randomly chosen j, the probability that any particular one of M^A's accepting paths inspects subblock j is at most $q(n)/(p2^n)$.

Choose k distinct subblocks $i_1, \ldots, i_k$ at random. For each t, the probability that M^A rejects when using an oracle $A \in C_{i_t}$ is $o(1)$; the probability that M^A fails to inspect subblock i_t is $o(p)$; and the probability that any particular one of M^A's accepting paths inspects subblock i_u where $u \neq t$ is at most $q(n)/(p2^n)$. Since expectations are additive, the probability that there exists a t such that M^A rejects when using an oracle $A \in \mathcal{C}_{i_t}$; or such that M^A does not have an accepting path ρ_{i_t} that inspects subblock i_t; or such that there exists $u \neq t$ such that ρ_{i_t} inspects subblock u is at most

$$k \cdot (o(1) + o(p)) + \binom{k}{2} \cdot q(n)/(p2^n),$$

which is $o(1)$. If ρ_{i_t} does not inspect subblock i_u, then ρ_{i_t} is an accepting path regardless of whether $A \in C_{i_u}$. Thus the result of the preceding analysis applies even if we restrict A to contain the k events $i_1, \ldots, i_k$.

Randomly choose $i_1, \ldots, i_k$, and randomly choose an oracle A such that $i_1, \ldots, i_k$ are events in A. Let ρ_{i_t} be an accepting path of M^A that inspects subblock i_t. If ρ_{i_t} inspects subblock i_t but does not inspect subblock i_u, then $\rho_{i_t} \neq \rho_{i_u}$. This happens with probability at least $1 - o(1) = \Omega(1)$. Thus, M^A has k distinct accepting paths for $\Omega(1)$ of all oracles containing k events. By Lemma 12, M^A has exactly k events with probability $\Omega(1/n)$. Therefore, M^A has k distinct accepting paths for $\Omega(1/n)$ of all oracles. ∎

By considering more than one input of length n, we extend this separation to the case when k is not a constant.

Theorem 14 *Let $k(n)$ be a function that grows faster than any constant function, but such that $k(n) \leq (n-1)\ln 2 - \ln\ln n$. For almost all oracles A,*

$$\mathrm{U}_{k(n)-1}\mathrm{P}^A \subset \mathrm{U}_{ek(n)+2\ln n}\mathrm{P}^A.$$

As in the preceding proof, an appealing strategy is to define $k(n)2^n$ events so that the probability of more than $k(n)$ events is at least $1/n$ but the probability of $ek(n)$ events is at most $1/n^2$. Unfortunately, in this case the probability of at least one event's occurring approaches 1, so for almost oracles A the language we define is co-finite (hence in P^A). Fortunately, a technique like Blum, Impagliazzo, and Rudich's will help us to overcome this pitfall.

Proof: Let the nth block of A consist of $k(n)2^n$ subblocks each having length n. Instead of looking for runs in the oracle A, we will look for arbitrary patterns. We say that a pattern x "occurs" in the oracle A if A contains a subblock equal to x. (It is not enough for x to appear in an arbitrary position in the characteristic sequence of A; in fact every string appears somewhere in A with probability 1.)

By considering arbitrary patterns instead of runs, we are able to construct a language that is co-infinite with probability 1. We write $x < m$ if x is the binary representation of a number that is less than m.

$$\mathrm{PAT}^A = \{x : x < 2e^{k(n)} \ln n \text{ and } x \text{ occurs in } A\}.$$

For every string x, the expected number of occurrences of x is $k(|x|)2^{|x|}2^{-|x|} = k(|x|)$. Because the area of the binomial distribution is centered at the expected value, the probability that x occurs at least $k(|x|)$ times is greater than 1/2.

We show that $\mathrm{PAT}^A \in \mathrm{U}_{ek(n)+2\ln n}\mathrm{P}^A$ for almost all oracles A. Let $p = k(n)$, and let x denote a string of length n. By Proposition 11, the probability that x occurs exactly j times is approximately $e^{-p}p^j/j!$. Letting $j = ep$, we see that the probability that x occurs exactly ep times is approximately

$$\begin{aligned}
e^{-p}p^{ep}/(ep)! &= \theta(e^{-p}p^{ep}e^{ep}(ep)^{-ep}/\sqrt{2\pi ep}), \\
&\quad \text{by Stirling's approximation} \\
&= \theta(e^{-p}/\sqrt{2\pi ep}) \\
&= o(e^{-p}).
\end{aligned}$$

Furthermore, for $j \geq ep$, we have

$$\frac{e^{-p}p^{j+1}/(j+1)!}{e^{-p}p^j/j!} = p/(j+1) \leq 1/e,$$

so the probability that x occurs exactly $j+1$ times is at most $1/e$ times the probability that x occurs exactly j times. Therefore the probability that x occurs exactly $ep + 2\ln n$ times is $o(e^{-p}/n^2)$, and, summing a geometric progression, we find that the probability that x occurs at least $ep + 2\ln n$ times is also $o(e^{-p}/n^2)$. Summing over strings of length n that precede $2e^p \ln n$, we find that the probability that at least one string of length n occurs at least $ep + 2\ln n$ times is $o(2e^p \ln n e^{-p}/n^2) = o(\ln n/n^2)$. Therefore the probability that such a string exists for infinitely many values of n is 0. Since we can encode the finite number of exceptions into a finite table, we have for almost all A

$$\mathrm{PAT}^A \in \mathrm{U}_{ek(n)+2\ln n}\mathrm{P}^A.$$

Let M^A be a $\mathrm{U}_{k(n)-1}\mathrm{P}^A$ machine that attempts to accept RUNS^A. M^A errs if it accepts when it should reject, rejects when it should accept, or has more than $k(n)$ accepting paths for some input. We will show that for $\Omega(1/n)$ of all oracles, M^A errs on at least one input, and that these errors are independent. Therefore, M^A errs at least once with probability 1. By Bennett and Gill's Lemma 1 [4], $\mathrm{PAT}^A \notin \mathrm{U}_{k(n)-1}\mathrm{P}^A$ for almost all A.

Let M^A run in time $q(n)$ on input of length n for all oracles A. Fix a value of n. Fix also $A^{[i]}$ for all $i \neq n$. We prove by contradiction that M^A errs with probability $\Omega(1/n)$; because the proof is valid regardless of membership in A of strings whose length is different from n, this error probability is independent of the errors on other inputs.

We say that a computational path *inspects* a subblock if the path queries at least one element of the subblock.

Let $\mathcal{C}$ be the class of all oracles A such that there exists a length-n string $x < 2e^p \ln n$ that occurs exactly once in A. Then

$$\begin{aligned}
1 - \mu(\mathcal{C}) &< (1 - e^{-p}p)^{2e^p \ln n} \\
&\leq \left(e^{-e^{-p}p}\right)^{2e^p \ln n} \\
&\quad \text{because } (\forall x)[1 + x \leq e^x] \\
&= e^{-2e^{-p}pe^p \ln n} \\
&= e^{-2p\ln n} \\
&= n^{-2p},
\end{aligned}$$

so $\mu(\mathcal{C}) = \Omega(1)$.

If $A \in \mathcal{C}$, then with probability $1 - o(1/n)$, M^A must accept every length-n string x that occurs in A, for otherwise M^A errs with probability $\Omega(1/n)$. In this case, M^A has an accepting path ρ with probability $1 - o(1/n)$. Let $\mathcal{C}_i$ be the class of all oracles A such that the ith subblock of A is the unique occurrence of some string. Let x denote the value of subblock i. If, on input x, ρ does not inspect subblock i then ρ accepts whenever subblock i takes a value different from x. Let α be the probability that, on input equal to the string contained in subblock i, ρ does not inspect subblock i. Then M^A accepts incorrectly with probability $\alpha \cdot (2^n - 1)/2^n \cdot \Omega(1)$, so $\alpha = o(1/n)$. Therefore, with probability $1 - o(1/n)$, M^A inspects and accepts every uniquely occurring string in A.

Let A contain $k(n)$ randomly selected occurrences of a length-n string x. (This happens with probability at least 1/2.) Then, as in the preceding proof, M^A has fewer than $k(n)$ accepting paths with probability

$$k(n)(o(1/n) + o(1/n)) + \binom{k(n)}{2} q(n)/2^n,$$

which is $o(1)$ because $k(n) = O(n)$. Thus M^A is not a $\mathrm{U}_{k(n)-1}\mathrm{P}^A$ machine with probability $\Omega(1)$. As before, $\mathrm{PAT}^A \notin \mathrm{U}_{k(n)-1}\mathrm{P}^A$ with probability 1. ∎

By standard padding techniques [10], this separation can be refined.

Corollary 15 *Let $k(n)$ be a time-constructible function such that* $\ln n \leq k(n) \leq (n-1)\ln 2 - \ln\ln n$. *For* *all constants $c > 1$, for almost all oracles A,*

$$\mathrm{U}_{k(n)}\mathrm{P}^A \subset \mathrm{U}_{ck(n)}\mathrm{P}^A.$$

We have seen that additional accepting paths are helpful relative to one oracle A when $k(n) \ll 2^{n/2}$, and helpful relative to almost all oracles when $k(n) \leq (n-1)\ln 2 - 2\ln\ln n$. We would be interested in knowing precisely when additional accepting paths stop helping (if ever).

4. Uniform Acceptors

Let $\mathcal{D}$ denote a class of oracle machines. We assume familiarity with relativizations of classes of languages and machines [2]. We say that $M^{()}$ is a uniform $\mathcal{D}^{()}$ machine if M^A is a $\mathcal{D}^A$ machine for every oracle A. Note that the test languages constructed in the preceding section were not accepted by uniform $\mathrm{U}_k\mathrm{P}^{()}$ machines, because we had to handle finitely many exceptions by table lookup. Now we show that it is not possible to use a uniform test language to prove the separations in the preceding section, unless we also prove $\mathrm{P} \neq \mathrm{NP}$.

In [9], it was shown that every uniform $\mathrm{UP}^{()}$ machine can be simulated by a uniform $\mathrm{P}^{()}$ machine, if $\mathrm{P} = \mathrm{NP}$. In [5, 9], it was shown that every uniform $(\mathrm{NP} \cap \text{co-NP})^{()}$ machine can be simulated by a uniform $\mathrm{P}^{()}$ machine, if $\mathrm{P} = \mathrm{NP}$. We will present a theorem that unifies those two results, and also extends them to $\mathrm{U}_k\mathrm{P}$ and p-separability. Although the techniques are easy extensions of [5, 9], we think that the results are interesting.

Lemma 16 *Assume that $\mathrm{P} = \mathrm{NP}$. Let $N^{()}$ be a nondeterministic polynomial-time oracle Turing machine that takes an ordered pair of inputs (x, y) where $|y| \leq p(|x|)$ for some polynomial p. Suppose that there exists a polynomial $k(n)$ such that*

$$(\forall A)(\forall x)[|\{y : N^A(x,y) \textit{ accepts}\}| \leq k(|x|)].$$

Then there is a deterministic polynomial-time oracle Turing machine $M^{()}$ such that for all oracles A and inputs x, $M^A(x)$ outputs a set containing $k(|x|)$ strings such that

$$(\exists y)[N^A(x,y) \textit{ accepts}]$$
$$\iff \quad (\exists y \in M^A(x))[N^A(x,y) \textit{ accepts}]$$

Proof: Let $k = k(|x|)$. If $k = 0$ then the problem is trivial. Without loss of generality, assume that no path in N makes the same query more than once. Consider $N_1^{()}$, an $\mathrm{NP}^{()}$ machine that simulates $N^{()}(x, y)$ nondeterministically for all y and all possible oracle answers. We say that two paths in $N_1^{()}$ are disjoint if they have no queries in common. Nondeterministically, we can find disjoint accepting paths in $N_1^{()}$ for k distinct values of y, if they exist. (If they do not exist, then we repeat this procedure using $k-1$ in place of k.) Since we assume that $\mathrm{P} = \mathrm{NP}$, we can perform that task deterministically [2]. Call the values $y_1, \ldots, y_k$. By making a polynomial number of queries to A, we check whether any of these k potential accepting paths is an actual accepting path using oracle A. If so, then we output the corresponding value of y and we are done. If not, then we build the oracle answers obtained into a table in $N^{()}$; we repeat the procedure above using the modified $N^{()}$. If we are not already done after $p(|x|) + 1$ iterations, then we output $\{y_1, \ldots, y_k\}$.

Now we prove the algorithm's correctness. We say that two paths *intersect* if there is a string queried on both paths. If $y \notin \{y_1, \ldots, y_k\}$, then every accepting path in $N^{()}(x, y)$ must intersect one of the accepting paths found for $y_1, \ldots, y_k$. (Otherwise there would be an oracle A and $k+1$ values of y such that $N^A(x, y)$ accepted.) Therefore, the last step of our algorithm shortens all accepting paths by at least one query for all but k values of y. After $p(|x|)$ iterations, every accepting path must make no queries for all but k values of y. An accepting path that makes no queries is an accepting path using oracle A, and will be detected. Otherwise the k remaining values of y must include any value of y such that $N^A(x, y)$ accepts. ∎

The following theorem says that if the languages accepted by two uniform $\mathrm{NP}^{()}$ machines are disjoint then they are separable by a uniform $\mathrm{P}^{()}$ machine.

Theorem 17 *Let $N_0^{()}$ and $N_1^{()}$ be uniform $\mathrm{NP}^{()}$ machines that always accept disjoint languages. Then there is a uniform $\mathrm{P}^{()}$ machine M such that for all A and all x*

$$M^A(x) \begin{cases} \textit{accepts} & \textit{if } N_0^{()}(x) \textit{ accepts} \\ \textit{rejects} & \textit{if } N_1^{()}(x) \textit{ accepts.} \end{cases}$$

Proof: Let

$$N^A(x,y) = \begin{cases} N_0^A(x) & \text{if } y = 0 \\ N_1^A(x) & \text{if } y = 1 \\ \text{reject} & \text{otherwise.} \end{cases}$$

Apply Lemma 16 with $k(n) = 1$. ∎

As a corollary we obtain Blum and Impagliazzo's and Hartmanis and Hemachandra's result that if two uniform $\mathrm{NP}^{()}$ machines accept complementary languages, then those languages are accepted by uniform $\mathrm{P}^{()}$ machines.

Corollary 18 *Let $N_0^{()}$ and $N_1^{()}$ be uniform* $\mathrm{NP}^{()}$ *machines that accept complementary languages. Then there is a uniform* $\mathrm{P}^{()}$ *machine M such that for all A and all x $M^A(x)$ accepts iff $N_0^A(x)$ accepts.*

Next we show that a uniform separation between $\mathrm{P}^{()}$ and $\mathrm{U}_k\mathrm{P}^{()}$ would prove $\mathrm{P} \neq \mathrm{NP}$.

Theorem 19 *Assume that* $\mathrm{P} = \mathrm{NP}$. *Let $U^{()}$ be a uniform* $\mathrm{U}_{k(n)}\mathrm{P}^{()}$ *machine, for some polynomial k. Then there is a uniform* $\mathrm{P}^{()}$ *machine $M^{()}$ such that for all A and all x $M^A(x)$ accepts iff $U^A(x)$ accepts.*

Proof: Let

$$N^A(x,y)\begin{cases} \text{accept} & \text{if } y \text{ is an accepting path for } U^A(x) \\ \text{reject} & \text{otherwise.} \end{cases}$$

By the Lemma 16, there is a uniform $\mathrm{P}^{()}$ machine that produces a set of $k(n)$ candidates for an accepting path y. We can check these $k(n)$ paths deterministically. ∎

We obtain Hartmanis and Hemachandra's result by letting $k(n) = 1$.

Corollary 20 *Assume that* $\mathrm{P} = \mathrm{NP}$. *Let $U^{()}$ be a uniform* $\mathrm{UP}^{()}$ *machine. Then there is a uniform* $\mathrm{P}^{()}$ *machine $M^{()}$ such that for all A and all x $M^A(x)$ accepts iff $U^A(x)$ accepts.*

Grollman and Selman have shown [8] that if all disjoint sets in NP are p-separable then $\mathrm{P} = \mathrm{UP}$. A similar argument shows that the preceding corollary also follows directly from Theorem 17.

5. Acknowledgments

I am grateful to Manuel Blum for helpful discussions, to Russell Impagliazzo for very helpful suggestions, and to Eric Allender and Lane Hemachandra for encouragement.

References

[1] E. W. Allender. The complexity of sparse sets in P. In A. L. Selman, editor, *Structure in Complexity Theory*, pages 1–11. Springer-Verlag, June 1986. Lecture Notes in Computer Science 223.

[2] T. Baker, J. Gill, and R. Solovay. Relativizations of the $P =?\ NP$ question. *SICOMP*, 4:431–442, 1975.

[3] R. Beigel. Relativized counting classes: Relations among thresholds, parity, and mods. *JCSS*, 42(1):76–96, Feb. 1991.

[4] C. H. Bennett and J. Gill. Relative to a random oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$ with probability 1. *SICOMP*, 10:96–112, Feb. 1981.

[5] M. Blum and R. Impagliazzo. Generic oracles and oracle classes. In *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 118–126, 1987.

[6] R. V. Book, T. Long, and A. Selman. Quantitative relativizations of complexity classes. *SICOMP*, 13:461–487, 1984.

[7] J. Geske and J. Grollman. Relativizations of unambiguous and random polynomial time classes. *SICOMP*, 15(2):511–519, May 1986.

[8] J. Grollman and A. Selman. Complexity measures for public-key cryptosystems. *SICOMP*, 17(2):309–335, Apr. 1988.

[9] J. Hartmanis and L. A. Hemachandra. One-way functions, robustness, and the non-isomorphism of NP-complete sets. In *Proceedings of the 2nd Annual Conference on Structure in Complexity Theory*, pages 160–174. IEEE Computer Society Press, June 1987.

[10] S. Ruby and P. C. Fischer. Translational methods and computational complexity. In *Proceedings of the Sixth Annual IEEE Symposium on Switching, Circuit Theory, and Logical Design*, pages 173–178, 1965.

[11] S. Rudich. *Limits on the Provable Consequences of One-way Functions.* PhD thesis, University of California at Berkeley, Nov. 1988.

[12] L. G. Valiant. The relative complexity of checking and evaluating. *IPL*, 5:20–23, 1976.