Theoretical Computer Science 12 (1980) 161–173
© North-Holland Publishing Company

ON COUNTING PROBLEMS AND THE POLYNOMIAL-TIME HIERARCHY*

Dana ANGLUIN**
Department of Computer Science, Edinburgh University, Edinburgh, EH9 3JZ, United Kingdom

Communicated by R. Book
Received April 1978
Revised October 1979

Abstract. We consider the relation between the relativized polynomial time hierarchy and relativizations of Gill's class PP of sets recognizable in polynomial time by probabilistic Turing machines and of Valiant's class D#P of sets polynomial time Turing reducible to functions that give the number of accepting computations of nondeterministic polynomial-time bounded Turing machines. The main result is that there exists an oracle set A such that $\mathrm{PP}^A - (\Pi_2^{P,A} \cup \Sigma_2^{P,A}) \neq \emptyset$, with the corollary that also $\mathrm{D\#P}^A - (\Pi_2^{P,A} \cup \Sigma_2^{P,A}) \neq \emptyset$. The proof is an application of Baker and Selman's technique for showing that $\Sigma_2^{P,A} \subsetneq \Sigma_3^{P,A}$ for some oracle set A.

1. Introduction

Several investigators have considered the idea of looking at the number (or fraction) of accepting computations among all the computations of a nondeterministic machine on a given input, and posed questions about the relations between notions of computation based on this idea and the more familiar notions of deterministic and nondeterministic computation, specifically relations to the classes P and NP of sets recognizable deterministically and nondeterministically in polynomial time. We briefly describe three such approaches.

Gill [4] defines computation on a 'probabilistic Turing machine' by saying that an input is accepted by a machine if and only if the probability of an accepting computation of the machine on the input exceeds $\frac{1}{2}$ (where each binary nondeterministic choice is viewed as a Bernoulli trial with $p = \frac{1}{2}$). He denotes PP the class of sets recognized by polynomial-time bounded probabilistic Turing machines. It is not difficult to see that NP ⊆ PP; however, the question of whether P = ?PP is open.

Miller [6] and Adleman and Manders [1] define an apparently more restricted and 'practical' notion of random recognition, by analogy to probabilistic algorithms to test primality [8]. In this definition, a machine M recognizes a set S provided for all

* This research was supported by the Science Research Council of the United Kingdom and also by the National Science Foundation under grant MCS77-11350.

** Author's current address: Department of Computer Science, Yale University, New Haven, CT 06520, U.S.A.

$x \notin S$ there is *no* accepting computation of M on input x, and for all $x \in S$, the probability of an accepting computation of M on input x exceeds $\frac{1}{2}$. (Note that there are probabilistic Turing machines that do not recognize, in this sense, any set.) If we denote by RP (Adleman and Manders use simply R) the class of sets recognizable in this sense by polynomial-time bounded nondeterministic Turing machines, then it is not difficult to see that $\mathrm{RP} \subseteq \mathrm{NP}$; however, the question of whether $\mathrm{P} = ?\mathrm{RP}$ is open.

Valiant [10, 11] considers the class of functions f such that for some polynomial-time bounded nondeterministic Turing machine M and for all inputs x, $f(x)$ is precisely the number of distinct accepting computations of M on input x. This class he denotes by #P. A typical member of #P is the function that gives for each labelled graph the number of distinct Hamiltonian paths in the graph. Valiant considers a related class of sets which he denotes by D#P consisting of those sets that can be recognized by deterministic polynomial-time bounded Turing machines with oracles for members of #P. It is not difficult to see that $\mathrm{PP} \subseteq \mathrm{D\#P}$; however, the question of whether $\mathrm{P} = ?\mathrm{D\#P}$ is open.

To summarize the known relations among these classes,

$$\mathrm{P} \subseteq \mathrm{RP} \subseteq \mathrm{NP} \subseteq \mathrm{PP} \subseteq \mathrm{D\#P} \subseteq \mathrm{PSPACE}$$

and it is an open question whether any of these inclusions is proper. If we consider these classes relativized to oracle sets, then a bit more can be said about the inclusions for them. It is not difficult to verify that for every oracle set A,

$$\mathrm{P}^A \subseteq \mathrm{RP}^A \subseteq \mathrm{NP}^A \subseteq \mathrm{PP}^A \subseteq \mathrm{D\#P}^A \subseteq \mathrm{PSPACE}^A.$$

Furthermore, for example, there exist oracle sets A and B such that $\mathrm{P}^A = \mathrm{NP}^A$ and $\mathrm{P}^B \subsetneq \mathrm{NP}^B$. This and other results concerning relations among the relativized classes $\mathrm{P}^{(\,)}$, $\mathrm{NP}^{(\,)}$, co-$\mathrm{NP}^{(\,)}$ and $\mathrm{PSPACE}^{(\,)}$ may be found in [2]. Rackoff [7] has shown that there exist oracle sets A and B such that $\mathrm{P}^A = \mathrm{RP}^A \subsetneq \mathrm{NP}^A$ and $\mathrm{P}^B \subsetneq \mathrm{RP}^B = \mathrm{NP}^B$. Baker and Selman [3] have considered classes higher in the polynomial time hierarchy [9, 12] and shown that there exists an oracle set A such that $\Sigma_2^{P,A} \subsetneq \Sigma_3^{P,A}$.

In this paper we show that there exists an oracle set A such that $\mathrm{PP}^A - (\Sigma_2^{P,A} \cup \Pi_2^{P,A}) \neq \emptyset$ and consequently $\mathrm{D\#P}^A - (\Sigma_2^{P,A} \cup \Pi_2^{P,A}) \neq \emptyset$. The proof proceeds by giving some sufficient conditions for the application of Baker and Selman's ingenious argument showing that there exists an oracle set A such that $\Sigma_2^{P,A} \subsetneq \Sigma_3^{P,A}$. It is hoped that these conditions will be of use in similar situations. In addition, we formulate a notion of recognizing properties of the oracle set that seems to underlie this and other relativization arguments. We also give a particular nonuniform analogue of the relativized polynomial-time hierarchy that appears to formalize a fairly common tool of thought in this area, and also seems to be a natural setting for the combinatorial portion of Baker and Selman's argument (Lemma 4.4).

Concerning the relation between counting problems and the polynomial-time hierarchy, it should be noted that Mathon [5] has shown that the problem of determining the number of isomorphisms between two labelled graphs is polynomial-time Turing reducible to a set in NP, by using the group structure of the set of isomorphisms. This result has the corollary that if counting graph isomorphisms were complete in #P (with respect to polynomial-time Turing reducibility), then we would have D#P $\subseteq \Sigma_2^P \cup \Pi_2^P$, in contrast to the relativized case. However, the question of whether counting graph isomorphisms is complete in #P is open. Also no polynomial-time reduction is known of a problem proved to be complete in #P, (e.g., counting Hamiltonian circuits or counting perfect matchings in a graph [10, 11]) to a set at any level in the polynomial-time hierarchy.

2. Definitions

In this section we describe our basic model of computation and give definitions of the relativized classes of interest to us in the remainder of this paper. Other definitions are introduced as required in subsequent sections.

The *alphabet* is $\Sigma = \{0, 1\}$. The set of all words over Σ is denoted Σ^*. The length of a string $y \in \Sigma^*$ is denoted $|y|$. The set of all strings over Σ of length n is denoted Σ^n. The cardinality of a set S is denoted $|S|$. If $S \subseteq \Sigma^*$, then $\Sigma^* - S$ is denoted $\bar{S}$.

The model of computation we use is an oracle Turing machine acceptor. This is a Turing machine with several one-way infinite tapes: a read-only *input tape*, a finite number of read–write *work tapes*, a write-only *oracle query tape*, and a read-only *oracle answer tape*. Each tape has a single head, initially positioned at the first square of the tape. Initially, all tapes except the input tape are blank, and the input tape contains the input string followed by blanks. There is a distinguished *initial state* of the machine, and some states of the machine are *accept* states, some others are *reject* states. Whenever the machine enters an accept or reject state, the computation is terminated and called *accepting* or *rejecting* accordingly. In addition, there are two other distinguished states, the *query state* and the *answer state*. Computations of the machine are defined with respect to an *oracle function*, which is any total function from Σ^* to Σ^*. Steps of the machine involving neither the query nor the answer state are defined as usual. When the machine enters the query state, if the initial nonblank portion of the query tape is the string $x \in \Sigma^*$ and the oracle function is f, then at the next step, the query tape is erased to blanks and its head is positioned to the first square, the answer tape contains the string $f(x)$ followed by blanks and its head is positioned to the first square, and the machine is in the answer state. No other tape contents or head positions are altered in this step

An oracle Turing machine acceptor may be either *deterministic* or *nondeterministic*. We shall want to use subsets of Σ^* as oracles; these are assumed to be presented by their characteristic functions. The name of such a machine will typically be written $M^{()}$ to indicate that it takes an oracle. Machine $M^{()}$ with oracle f (a function) or A (a set) will be denoted M^f or M^A.

Such a machine $M^{()}$ will be said to *run in time* $t(n)$ if and only if for every oracle function $f: \Sigma^* \to \Sigma^*$ and for every nonnegative integer n and every string $x \in \Sigma^n$, every computation of M^f on input x terminates after at most $t(n)$ steps in an accept or reject state.

In particular, we consider the sequence of polynomials $p_1, p_2, p_3, \ldots$ where $p_i(x) = x^i + i$ for all i, x, and a standard effective enumeration of deterministic oracle Turing machine acceptors $M_1^{()}, M_2^{()}, M_3^{()}, \ldots$ and of nondeterministic oracle Turing machine acceptors $N_1^{()}, N_2^{()}, N_3^{()}, \ldots$ such that for each i, $M_i^{()}$ and $N_i^{()}$ run in time p_i.

For each $A \subseteq \Sigma^*$, the class P^A is the collection of all sets $S \subseteq \Sigma^*$ such that for some i, for all $x \in \Sigma^*$, $x \in S$ if and only if the unique computation of M_i^A on input x halts in an accept state.

For each $A \subseteq \Sigma^*$, the class NP^A is the collection of all sets $S \subseteq \Sigma^*$ such that for some i, for all $x \in \Sigma^*$, $x \in S$ if and only if some computation of N_i^A on input x halts in an accepting state.

Given $A \subseteq \Sigma^*$ and $x \in \Sigma^*$, we define the *probability* associated with a particular computation of N_i^A on input x as 2^{-k}, where k is the number of nondeterministic binary choices occurring in this particular computation. For each $A \subseteq \Sigma^*$, the class PP^A is the collection of all sets $S \subseteq \Sigma^*$ such that for some i, for all $x \in \Sigma^*$, $x \in S$ if and only if the sum of the probabilities associated with accepting computations of N_i^A on input x exceeds $\frac{1}{2}$. This definition differs slightly from Gill's in [4], but as he points out there, the same class is defined. In particular, we note that PP^A is closed under complementation.

For each $A \subseteq \Sigma^*$, $\#\mathrm{P}^A$ is the class of functions $f: \Sigma^* \to \Sigma^*$ such that for some i and all $x \in \Sigma^*$, $f(x)$ is the binary representation of the number of distinct accepting computations of N_i^A on input x.

For each $A \subseteq \Sigma^*$, $\mathrm{D}\#\mathrm{P}^A$ is the class of sets $S \subseteq \Sigma^*$ such that for some i and for some $f \in \#\mathrm{P}^A$, and for all $x \in \Sigma^*$, $x \in S$ if and only if the unique computation of M_i^f on input x halts in an accepting state.

Finally, to define $\Sigma_2^{P,A}$ and $\Pi_2^{P,A}$, for each $A \subseteq \Sigma^*$ we say $\Sigma_2^{P,A}$ is the class of sets $S \subseteq \Sigma^*$ such that $S \in \mathrm{NP}^B$ for some $B \in \mathrm{NP}^A$. $\Pi_2^{P,A}$ is the set of all sets $S \subseteq \Sigma^*$ such that $\bar{S} \in \Sigma_2^{P,A}$. We shall rely on a characterization of $\Pi_2^{P,A}$ given in [3] as follows. Let U_i^A denote the set of all those $x \in \Sigma^*$ such that for all $y \in \Sigma^*$ with $|y| \leq p_i(|x|)$ there exists a $z \in \Sigma^*$ with $|z| \leq p_i(|x|)$ such that M_i^A accepts $\langle x, y, z\rangle$ where $\langle a, b, c\rangle$ is some standard linear-time computable encoding of a triple of strings into a single string. Then also $\Pi_2^{P,A} = \{U_1^A, U_2^A, U_3^A, \ldots\}$.

Each of these definitions can be viewed as specifying a function mapping oracle sets $A \subseteq \Sigma^*$ into classes or sets; we shall denote these functions by $\mathrm{P}^{()}, \mathrm{NP}^{()}, \mathrm{PP}^{()}, \#\mathrm{P}^{()}, \mathrm{D}\#\mathrm{P}^{()}, \Sigma_2^{P,()}, \Pi_2^{P,()}$, and $U_i^{()}$ respectively, and speak of them somewhat loosely as 'relativized classes' or 'relativized sets'. This should, however, lead to no ambiguity. We omit a definition of RP or RP^A, since this is not required in what follows; see [1, 7] for these definitions.

3. Oracle properties

In several of the proofs concerning relativized complexity classes, it seems natural to view the oracle machines as attempting to recognize certain properties of the oracle set, where the particular input string serves only to select a certain portion of the oracle set. More concretely, a typical recognition problem is: "accept the input x if and only if the subset of strings of length $|x|$ of the oracle set has property Q." For example, Q is nonemptiness in Baker, Gill, and Solovay's construction of a set A such that $P^A \subsetneq NP^A$. We formalize this notion below.

A *ranked oracle property* Q is an indexed collection of classes $Q = \{Q_0, Q_1, Q_2, \ldots\}$ such that each Q_n is a subset of $\mathcal{P}(\Sigma^n)$. Thus Q_n specifies which sets of strings of length n have the 'desired property'. For brevity, we shall omit the word 'ranked' in the remainder of this paper. An oracle property Q is *recursive* if and only if the map $n \mapsto Q_n$ can be computed by a Turing machine, using some straightforward representation of the finite collection Q_n of finite sets.

If Q is any oracle property, we define a related map $\mathcal{R}_Q$ from oracle sets to recognition problems as follows. For each set $A \subseteq \Sigma^*$,

$$\mathcal{R}_Q(A) = \{x \in \Sigma^*: A \cap \Sigma^{|x|} \in Q_{|x|}\}.$$

If $\mathscr{C}^{()} = \{S_1^{()}, S_2^{()}, S_3^{()}, \ldots\}$ is a relativized class of sets, (e.g., $\mathscr{C}^{()}$ is one of $NP^{()}$, $\Pi_2^{P,()}$, etc.), then we say $S_i^{()}$ *represents* oracle property Q if and only if for every set $A \subseteq \Sigma^*$, $S_i^A = \mathcal{R}_Q(A)$.

The connection between relativized classes of sets $\mathscr{C}^{()}$ and oracle properties Q is in some cases given by a kind of 'uniformity' lemma that states that either for some set A, $\mathcal{R}_Q(A) \notin \mathscr{C}^A$ or for some positive integer i, $S_i^{()}$ represents Q. Clearly, if no $S_i^{()}$ represents Q, then for each i there exists A such that $S_i^A \neq \mathcal{R}_Q(A)$. What such a lemma asserts is that then there will exist a *single* set A such that for all i, $S_i^A \neq \mathcal{R}_Q(A)$, an interchange of quantifiers. We state such a lemma formally for $\Pi_2^{P,()}$.

Lemma 3.1. *Let $\Pi_2^{P,()} = \{U_1^{()}, U_2^{()}, U_3^{()}, \ldots\}$ be as in Section 2. Then for every (recursive) oracle property Q either there exists a (recursive) oracle set A such that $\mathcal{R}_Q(A) \notin \Pi_2^{P,A}$ or for some i, $U_i^{()}$ represents Q.*

This lemma is used implicitly by Baker and Selman in the reduction of Theorem 2.3 to Lemma 2.4 in [3]. We sketch the proof of this Lemma, which relies on closure of $\Pi_2^{P,A}$ under finite variation and on the finiteness of the number of oracle queries affecting the acceptance of any given input.

Proof of Lemma 3.1. Let Q be any oracle property. Suppose that $U_i^{()}$ does not represent Q for any i. We define by stages an oracle set A such that $\mathcal{R}_Q(A) \notin \Pi_2^{P,A}$. After stage i we shall have determined the set A_i of all numbers of A of length at most n_i in such a way that no matter how the definition of A is completed, $U_1^A, U_2^A, \ldots, U_i^A$ are all distinct from $\mathcal{R}_Q(A)$.

As a basis, we take $n_0 = 0$ and $A_0 = \emptyset$. At each stage, it will be possible to extend the definition of A as specified unless it happens at some stage i that for every possible extension of the definition of A, $U_i^A = \mathcal{R}_Q(A)$. If this is the case, then for every set B of strings of length exceeding n_{i-i}, $U_i^C = \mathcal{R}_Q(C)$, where $C = A_{i-1} \cup B$.

Then we may 'patch up' $U_i^{()}$ to get some $U_j^{()}$ that represents Q as follows. We begin by defining a machine $M^{()}$.

Let D be any oracle set and $\langle x, y, z\rangle$ be any input string. If $|x| \leq n_{i-1}$, then query D about every string of length $|x|$ and accept $\langle x, y, z\rangle$ if and only if the set of members of D of length $|x|$ is a member of $Q_{|x|}$. (This part may be done by table lookup.) If $|x| > n_{i-1}$, then let y' and z' be obtained from y and z by deleting any symbols after the initial $p_i(|x|)$ symbols and simulate M_i^C on input $\langle x, y', z'\rangle$, where $C = A_i \cup B$ and B is the set of strings of D of length exceeding n_{i-1}, and accept $\langle x, y, z\rangle$ if and only if M_i^C accepts $\langle x, y', z'\rangle$.

There exists some $j \geq i$ that is an index for the defined machine $M^{()}$ in the enumeration $M_1^{()}, M_2^{()}, \ldots$ of deterministic polynomial-time bounded oracle Turing machines. It is then straightforward but tedious to verify that $U_j^D = \mathcal{R}_Q(D)$ for every oracle set D, i.e., $U_j^{()}$ represents Q. This contradicts our hypothesis that no $U_i^{()}$ represents Q.

Thus, at any stage i there exists a set B of strings of length exceeding n_{i-1} such that $U_i^C \neq \mathcal{R}_Q(C)$, where $C = A_{i-1} \cup B$. Choose some such B and some string x_i such that $x_i \in U_i^C$ if and only if $x_i \notin \mathcal{R}_Q(C)$. Consider the set of queries in the computation of M_i^C on input $\langle x_i, y, z\rangle$ as y and z vary over all strings of length at most $p_i(|x_i|)$. There are finitely many such queries in each of finitely many computations, so we may choose some $n_i \geq \max\{n_{i-1}, |x_i|\}$ such that no string of length exceeding n_i is queried in these computations. Define A_i to be the set of elements of $C = A_{i-1} \cup B$ of length not exceeding n_i, and observe that for any extension of the definition of A,

$$x_i \in U_i^A \Leftrightarrow x_i \in U_i^C \Leftrightarrow x_i \notin \mathcal{R}_Q(C) \Leftrightarrow x_i \notin \mathcal{R}_Q(A)$$

because A and C agree on strings of length not exceeding n_i. Define $A = \bigcup_{i=0}^{\infty} A_i$.

It is clear that $\mathcal{R}_Q(A) \neq U_i^A$ for all i, i.e., $\mathcal{R}_Q(A) \notin \Pi_2^{P,A}$.

For the version of the lemma in which both Q and A are recursive, we need to verify that the definition of A may be made recursive by trying all finite extensions of A_{i-1} at stage i until an appropriate x_i is found.

4. Decision trees

In this section we define a 'model of computation' that gives a finite, non-uniform analogue of sets in the relativized polynomial time hierarchy. We give a formulation of the ingenious argument of Baker and Selman in [3] for this model, which is then used to obtain our main result in the next section.

A *uniform decision tree of rank n* is an ordered pair $T = (t, l)$, where t is a rooted ordered binary tree and l is a function on the nodes of t that maps every internal node of t to an element of Σ^n and every leaf of t to an element of $\{0, 1\}$. For such a T, if $S \subseteq \Sigma^*$, then there is a unique path $\pi(S) = (u_0, u_1, \ldots, u_k)$ from the root u_0 of t to a leaf u_k of t such that for each i, $0 \leq i < k$, if $l(u_i) \in S$, then u_{i+1} is the left son of u_i in t, otherwise u_{i+1} is the right son of u_i. We shall say that *T accepts S* if and only if

$l(u_k) = 1$, otherwise we say T rejects S. Define the collection of subsets of Σ^n accepted by T:

$$\mathscr{A}_n(T) = \{S \subseteq \Sigma^n : T \text{ accepts } S\}.$$

The *height* of T is simply the height of t as a binary tree, i.e., the number of edges in the longest path from root to leaf. For each $m, n \geq 0$ let

$$\mathscr{T}^n_m = \{T : T \text{ is a uniform decision tree of rank } n \text{ and height } \leq m\}.$$

The corresponding classes of sets are

$$\mathscr{D}^n_m = \{\mathscr{A}_n(T) : T \in \mathscr{T}^n_m\}.$$

Example 4.1. A member of $\mathscr{T}^3_3$ is exhibited in Fig. 1 that accepts all 96 subsets of $\{0, 1\}^3$ that contain 001, 010, and 100 or that contain 111 but not 001.

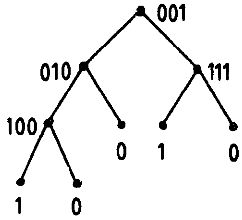

Fig. 1. An element of $\mathscr{T}^3_3$.

A decision tree is intended to represent the possible computation paths of an oracle machine, each query in a path being represented by an internal node, which gives rise to two continuations of the path depending on the answer to the query. We require a notion of bounded union and intersection to correspond to bounded quantification. If $\mathscr{C}$ is any class of sets and k is any nonnegative integer, let

$$\bigvee_k \mathscr{C} = \{S : S \text{ is a union of } \leq 2^k \text{ elements of } \mathscr{C}\},$$

$$\bigwedge_k \mathscr{C} = \{S : S \text{ is an intersection of} \leq 2^k \text{ elements of } \mathscr{C}\}.$$

The unbounded cases are simply:

$$\bigvee \mathscr{C} = \bigcup_{k=1}^{\infty} (\bigvee_k \mathscr{C}), \quad \text{and} \quad \bigwedge \mathscr{C} = \bigcup_{k=1}^{\infty} (\bigwedge_k \mathscr{C}).$$

Example 4.2. Let Q be the oracle property of nonemptiness, i.e., $Q_n = \mathscr{P}(\Sigma^n) - \{\emptyset\}$ for each n. Fix $n \geq 0$. For each $w \in \Sigma^n$ let $T_w \in \mathscr{T}^n_1$ be the tree that tests whether w is in the set and accepts if so. Thus $\mathscr{A}_n(T_w) = \{S \subseteq \Sigma^n : w \in S\}$. Then $Q_n = \bigcup_{|w|=n} \mathscr{A}_n(T_w)$ so $Q_n \in \bigvee_n \mathscr{D}^n_1$. This may be pictured as a sort of 'game tree' as in Fig. 2.

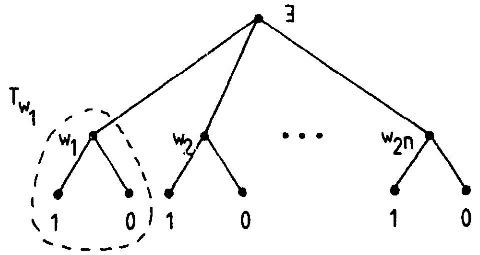

Fig. 2. Representing nonemptiness.

We now state and prove the connection we need between representing oracle properties by elements of $\Pi_2^{P,()}$ and by intersections of unions of sets recognized by decision trees. There is an obvious generalization to $\Pi_k^{P,()}$ or $\Sigma_k^{P,()}$ and k alternations of $\bigwedge_{p(n)}$ and $\bigvee_{p(n)}$.

Lemma 4.3. *Let* $\Pi_2^{P,()} = \{U_1^{()}, U_2^{()}, U_3^{()}, \ldots\}$ *be as in Section* 2. *Let* Q *be any oracle property and suppose* $j \geq 1$ *is such that* $U_j^{()}$ *represents* Q. *Then there exists a polynomial* q *such that for all* n, $Q_n \in \bigwedge_{q(n)} \bigvee_{q(n)} \mathscr{D}_{q(n)}^n$.

Proof. From the definition of $U_j^{()}, M_j^{()}$ is a deterministic oracle Turing machine acceptor that runs in time p_j and for every $A \subseteq \Sigma^*$, U_j^A is the set of all $x \in \Sigma^*$ such that for all strings y of length at most $p_j(|x|)$ there exists a string z of length at most $p_j(|x|)$ such that M_j^A accepts the input $\langle x, y, z\rangle$. Choose q to be a polynomial larger than p_j such that for any $A \subseteq \Sigma^*$ and any triple $\langle x, y, z\rangle$ such that $|y| \leq p_j(|x|)$ and $|z| \leq p_j(|x|)$, M_j^A runs for at most $q(|x|)$ steps on input $\langle x, y, z\rangle$.

Fix $n \geq 0$. For each pair (y, z) such that $|y|, |z| \leq p_j(n)$, consider the computation of $M_j^{()}$ on input $\langle 0^n, y, z\rangle$. (Different oracles A may result in different computation paths of M_j^A on input $\langle 0^n, y, z\rangle$.) Construct a decision-tree $T(y, z)$ representing the possible paths in this computation by simulating the computation until it makes a query or halts in an accepting or rejecting state. If it halts, the corresponding path in the decision tree is terminated at an appropriately labelled leaf (1 for accept, 0 for reject). If it makes a query of a string of length other than n, the simulation is continued as though the reply were negative. If the computation queries a string it has previously queried along this path, the simulation is continued in accordance with the earlier reply. If the computation queries a string u of length n that was not previously queried along this path, a new internal node is added to the path in the decision tree, with the label u. The simulation is then split into two continuations; one assuming the reply was 'yes', for the left branch of the path, the other assuming the reply was 'no'.

It should be clear that the decision tree $T(y, z)$ accepts the set $S \subseteq \Sigma^n$ if and only if M_j^S accepts $\langle 0^n, y, z\rangle$. Hence, by the hypothesis that $U_j^{()}$ represents Q, Q_n is precisely the set of $S \subseteq \Sigma^n$ such that for all y such that $|y| \leq p_i(n)$ there exists z such that $|z| \leqslant p_i(n)$ and M_j^S accepts $\langle 0^n, y, z\rangle$. Thus

$$Q_n = \bigcap_{|y| \leqslant p_i(n)} \bigcup_{|z| \leqslant p_i(n)} \mathscr{A}_n(T(y, z))$$

and since $2^{q(n)}$ bounds the number of terms in the union and intersection, $Q_n \in \bigwedge_{q(n)} \bigvee_{q(n)} \mathscr{D}_{q(n)}^n$.

The above lemma allows us to apply lower bounds obtained for the nonuniform model to the uniform model. We now isolate the combinatorial content of Baker and Selman's argument in a key lemma concerning the power of $\bigvee \mathscr{D}_m^n$ to represent sets.

Lemma 4.4. *Suppose $R \in \bigvee \mathscr{D}_m^n$ for some $m, n \geqslant 1$. Suppose that there exists a nonempty class $K \subseteq \mathscr{P}(\Sigma^n)$ such that*

(i) *no element of K is an element of R, and*

(ii) *every proper subset of an element of K is an element of R.*

Then there exists a set $S_0 \in K$ such that $m^{|S_0|} \geqslant |K|$.

(Thus, if K contains only 'small' sets and contains 'many' of them, this gives a lower bound on m, the height of the decision tree being used to represent R in $\bigvee \mathscr{D}_m^n$.)

Proof of Lemma 4.4. The result clearly holds when $|K| = 1$, so assume $|K| \geqslant 2$. Let J be an index set such that $R = \bigcup_{j \in J} \mathscr{A}_n(T_j)$ where each $T_j \in \mathscr{T}_m^n$. Since $|K| \geqslant 2$, there exists some nonempty set $S \in K$, so since $\emptyset$ is a proper subset of S, by (ii), $\emptyset \in R$, so $\emptyset \in \mathscr{A}_n(T_j)$ for some $j \in J$. Consider the path $\pi(\emptyset)$ accepting $\emptyset$ in T_j; this path can query at most m strings. Since for each $S \in K$, $S \notin R$, so $\pi(\emptyset)$ must query some string $w \in S$, otherwise S will be erroneously accepted by T_j. Thus for each $S \in K$, $\pi(\emptyset)$ queries some string in S, and $\pi(\emptyset)$ queries at most m distinct strings. Hence there exists a string w_1 such that $\pi(\emptyset)$ queries w_1 and at least $|K|/m$ elements of K contain w_1. Let $Q_1 = \{w_1\}$, $K_1 = \{S \in K: w_1 \in S\}$, so $|K_1| \geqslant |K|/m$. Inductively, assume we have constructed a set of strings Q_i of cardinality i, a subset K_i of K such that $K_i = \{S \in K: Q_i \subseteq S\}$, and $|K_i| \geqslant |K|/m^i$. (This is verified with $i = 1$ as basis.) Then either

(i) $|K_i| = 1$, so $1 \geqslant |K|/m^i$, so we take S_0 to be the unique element of K_i and since $Q_i \subseteq S_0$, $|S_0| \geqslant i$, so $m^{|S_0|} \geqslant |K|$, or

(ii) $|K_i| > 1$, so Q_i must be a *proper* subset of some $S \in K_i$, so $Q_i \in R$ by (ii). Then for some $k \in J$, T_k must accept Q_i. If $\pi(Q_i)$ is the path in T_k accepting Q_i, then there are at most m queries made along the path. T_k must reject every $S \in K_i$, so for each $S \in K_i$ there must be some string $w \in S - Q_i$ that $\pi(Q_i)$ queries, otherwise S would be erroneously accepted. Thus there is some string $w_{i+1} \notin Q_i$ queried by $\pi(Q_i)$ such that at least $|K_i|/m$ elements of K_i contain w_{i+1}. Choosing $Q_{i+1} = Q_i \bigcup \{w_{i+1}\}$ and $K_{i+1} = \{S \in K_i: w_{i+1} \in S\}$ will confirm the induction hypothesis for $i+1$, since by hypothesis $|K_i| \geqslant |K|/m^i$.

The form in which Baker and Selman used this argument was actually a little different, involving supersets and complements of critical elements. We state this form as a lemma; its proof is entirely analogous to that of Lemma 4.4.

Lemma 4.5. *Suppose $R \in \bigvee \mathscr{D}^n_m$ for some $m, n \geq 1$. Suppose K is a non-empty class of sets from $\mathscr{P}(\Sigma^n)$ such that:*

(i) *no element of K is an element of R, and*

(ii) *every proper superset of an element of K is an element of R. Then there exists some $S_0 \in K$ such that*

$$m^{|\Sigma^n - S_0|} \geq |K|.$$

As a simple consequence of Lemma 4.4, we may say something about the representation of sets in $\bigwedge_k \bigvee \mathscr{D}^n_m$.

Lemma 4.6. *Suppose $R \in \bigwedge_k \bigvee \mathscr{D}^n_m$ for some $k, m, n \geq 1$. Suppose K is a nonempty class of sets from $\mathscr{P}(\Sigma^n)$ such that:*

(i) *$K \cap R = \emptyset$, and*

(ii) *every proper subset of an element of K is an element of R. Then there exists some $S_0 \in K$ such that $m^{|S_0|} \geq |K|/2^k$.*

Proof. Fix

$$R = \bigcap_{i=1}^{2^k} \bigcup_{j \in J} \mathscr{A}_n(T_{ij}),$$

where each $T_{ij} \in \mathscr{D}^n_m$. Let R_i denote $\bigcup_{j \in J} \mathscr{A}_n(T_{ij})$ for each i, $1 \leq i \leq 2^k$. For each $S \in K$, $S \notin R$ so for some i, $S \notin R_i$. Thus there exists some i_0 such that if $K_0 = \{S \in K : S \notin R_{i_0}\}$, then $|K_0| \geq |K|/2^k$. Apply Lemma 4.4 with R_{i_0} and this K_0 to conclude that there is some $S_0 \in K_0 \subseteq K$ such that $m^{|S_0|} \geq |K_0| \geq |K|/2^k$.

5. Main result

Define the specific oracle property Q by $Q_n = \{S \subseteq \Sigma^n : |S| < 2^{\lfloor n/2 \rfloor}\}$ for all $n \geq 0$.

Lemma 5.1. *For all $A \subseteq \Sigma^*$, $\mathscr{R}_Q(A) \in \mathrm{PP}^A$.*

Proof. By a standard trick we construct a nondeterministic machine with over half its computations accepting if and only if the number of strings of length n in the oracle set is less than $2^{\lfloor n/2 \rfloor}$. On input x of length n, make an initial nondeterministic choice of alternative (i) or (ii) below:

(i) generate a string of n bits nondeterministically and accept if and only if it is among the first $2^{\lfloor n/2 \rfloor}$ such strings in lexicographic order;

(ii) generate a string y of length n and query the oracle set, accepting if and only if the reply is negative.

For a given oracle set A, if $t = |A \cap \Sigma^n|$, the probability of accepting an input is $> \frac{1}{2}$ if and only if $2^{\lfloor n/2 \rfloor} + (2^n - t) > 2^n$, i.e., $t < 2^{\lfloor n/2 \rfloor}$.

Lemma 5.2. *There exists a recursive oracle set* $A \subseteq \Sigma^*$ *such that* $\mathscr{R}_Q(A) \notin \Pi_2^{P,A}$.

Proof. Suppose to the contrary that for every recursive $A \subseteq \Sigma^*, \mathscr{R}_Q(A) \in \Pi_2^{P,A}$. Then, since Q is a recursive oracle property, by the uniformity lemma (Lemma 3.1) there exists an $i \geq 1$ such that $U_i^{()}$ represents Q. By the lemma on representation (Lemma 4.3) there exists a polynomial q such that for all $n \geq 0, Q_n \in \bigwedge_{q(n)} \bigvee_{q(n)} \mathscr{D}_{q(n)}^n$. For each n, let $C_n = \{S \subseteq \Sigma^n : |S| = 2^{\lfloor n/2 \rfloor}\}$, and apply Lemma 4.6 with Q_n for R and C_n for K to conclude that there exists some $S_0 \in C_n$ such that $(q(n))^{|S_0|} \geq |C_n|/2^{q(n)}$. But we know $|S_0| = 2^{\lfloor n/2 \rfloor}$ and

$$|C_n| = \binom{2^n}{2^{\lfloor n/2 \rfloor}}.$$

An elementary argument shows that

$$\binom{2^{2m}}{2^m} > 2^{m2^m}$$

for all $m \geq 1$, so for all $m \geq 1$,

$$(q(2m))^{2^m} > 2^{m2^m - q(2m)},$$

a contradiction.

Corollary 5.3. *There exists a recursive oracle set A such that* $\mathrm{PP}^A - \Pi_2^{P,A} \neq \emptyset$.

To finish off the proof of the main result, we need the following.

Lemma 5.4. *For any oracle set* $A \subseteq \Sigma^*$ *if* $\mathrm{PP}^A - \Pi_2^{P,A} \neq \emptyset$, *then* $\mathrm{PP}^A - (\Sigma_2^{P,A} \cup \Pi_2^{P,A}) \neq \emptyset$.

Proof. Suppose $S \in \mathrm{PP}^A - \Pi_2^{P,A}$. Define $T = \{1w : w \in S\} \cup \{0w : w \notin S\}$. We assert $T \in \mathrm{PP}^A$. Note that PP^A is closed under complement, so there exist polynomial-time probabilistic oracle Turing machines $N_i^{()}$ and $N_j^{()}$ such that N_i^A recognizes S and N_j^A recognizes $\bar{S}$. To recognize T with oracle A, on input $x = aw$, where $a \in \{0, 1\}$, select the computation of N_i^A on w if $a = 1$, and select the computation of N_j^A on w if $a = 0$; reject x if it is the null string.

To see that $T \notin (\Pi_2^{P,A} \cup \Sigma_2^{P,A})$ first suppose that $T \in \Pi_2^{P,A}$. Let $M_i^{()}$ be a polynomial-time bounded deterministic oracle Turing machine acceptor such that T is the set of all $x \in \Sigma^*$ such that for all $y \in \Sigma^*, |y| \leq p_i(|x|)$, there exists a $z \in \Sigma^*$,

$|z|\leqslant p_i(|x|)$ such that M_i^A accepts $\langle x, y, z\rangle$. We define a new machine $M^{(\ )}$ which on input $\langle x, y, z\rangle$ changes the input to $\langle 1x, y, z\rangle$ and calls M_i. If $p(n)$ is the polynomial $p_i(n+1)$, then S will be precisely the set of strings x such that for all $y\in\Sigma^*$, $|y|\leqslant p(|x|)$, there exists $z\in\Sigma^*$, $|z|\leqslant p(|x|)$ such that M^A accepts $\langle x, y, z\rangle$, so $S\in\Pi_2^{P,A}$, a contradiction. Similarly, if $T\in\Sigma_2^{P,A}$, we may use a machine that prefixes a zero to x to see that $\bar{S}\in\Sigma_2^{P,A}$, so $S\in\Pi_2^{P,A}$, a contradiction.

Thus $T\in \mathrm{PP}^A-(\Sigma_2^{P,A}\cup\Pi_2^{P,A})$.

Combining Corollary 5.3 and Lemma 5.4, we have

Theorem 5.5. *There exists a recursive oracle set A such that* $\mathrm{PP}^A-(\Pi_2^{P,A}\cup\Sigma_2^{P,A})\neq\emptyset$, *and consequently also* $\mathrm{D}\#\mathrm{P}^A-(\Pi_2^{P,A}\cup\Sigma_2^{P,A})\neq\emptyset$.

6. Remarks

One interesting open question is whether there exists an oracle set A such that $\mathrm{PP}^A\subsetneq\mathrm{PSPACE}^A$. Existing techniques, including Rackoff's argument [7] to construct an oracle B such that $\mathrm{P}^B=\mathrm{RP}^B\subsetneq\mathrm{NP}^B$, do not seem to suffice in this case.

Acknowledgment

The careful and constructive criticism of the referees was extremely helpful in revising the original version of this paper.

References

[1] L. Adleman and K. Manders, Reducibility, randomness, and intractability, *Proc. 9th Annual ACM Symposium on Theory of Computing*, Boulder, CO (1977) 151–163.

[2] T. Baker, J. Gill and R. Solovay, Relativizations of the P = ?NP question, *SIAM J. Comput.* **4** (1975) 431–442.

[3] T. Baker and A. Selman, A second step toward the polynomial hierarchy, *Theoret. Comput. Sci.* **8** (1979) 177–187.

[4] J. Gill, Computational complexity of probabilistic Turing machines, *SIAM J. Comput.* **6** (1977) 675–695.

[5] R. Máthon, A note on the graph isomorphism counting problem, *Information Processing Lett.* **8** (1979) 131–132.

[6] G.L. Miller, Riemann's hypothesis and tests for primality, Ph.D. dissertation, Mathematics Department, University of California at Berkeley (1975).

[7] C. Rackoff, Relativized questions involving probabilistic algorithms, *Proc. 10th Annual ACM Symposium on Theory of Computing*, San Diego, CA (1978) 338–342.

[8] R. Solovay and V. Strassen, A fast Monte-Carlo test for primality, *Siam J. Comput.* **6** (1977) 84–85; also Erratum, **7** (1978) 118.

[9] L.J. Stockmeyer, The polynomial-time hierarchy, *Theoret. Comput. Sci.* **3** (1977) 1–22.

[10] L.G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* **8** (1979) 189–201.

[11] L.G. Valiant, The complexity of reliability and enumeration problems, *SIAM J. Comput.* **8** (1979) 410–421.

[12] C. Wrathall, Complete sets and the polynomial hierarchy, *Theoret. Comput. Sci.* **3** (1977) 23–33.