Relativized Polynomial Time Hierarchies Having Exactly K Levels*

(Preliminary Version)

Ker-I Ko
Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794

1. Introduction

One of the main goals in complexity theory is to develop proof techniques to separate complexity classes. While it is well recognized that most separation results are beyond today's proof techniques, interesting progress has been made recently on separation results for relativized complexity classes. Baker, Gill and Solovay [2] showed that the relativized $P=?NP$ question may be answered both ways depending on the oracles; i.e., there exist sets X and Y such that $P(X) = NP(X)$ and $P(Y) \neq NP(Y)$. Baker and Selman [3] extended it to the second level of the polynomial time hierarchy showing that there exists a set Z such that $\Sigma_3^P(Z) \neq \Sigma_2^P(Z)$. The proof technique of Baker and Selman's result is a complicated counting argument which, however, does not seem powerful enough to be applicable to separating the third level of the relativized polynomial time hierarchy.

More recently, Furst, Saxe and Sipser [4] and Sipser [9] proposed the idea of applying probabilistic arguments to this problem. They reduced the problem of separating the relativized polynomial time hierarchy to the problem of proving lower bounds on the size of small depth circuits. The major breakthrough in this direction is due to Yao [12] who, based on Furst, Saxe and Sipser's idea, showed an exponential lower bound on the size of small depth parity circuits and hence exhibited an oracle A which separates the class $PSPACE(A)$ from $PH(A)$. Hastad [5, 6] simplified Yao's proof and gave a proof for the claim made in [12] that there exists an oracle B such that for all $k > 0$, $PH(B) \neq \Sigma_k^P(B)$. We summarize the known results about the relativized polynomial time hierarchies as follows:

(1) $\forall A\ \forall k > 0\ [\Sigma_k^P(A) = \Pi_k^P(A) \Rightarrow PH(A) = \Sigma_k^P(A)]$ (Stockmeyer [10]).

(2) $\exists B\ PSPACE(B) = PH(B) = P(B)$ (Baker, Gill and Solovay [2]).

(3) $\exists C\ \forall k > 0\ PSPACE(C) \neq PH(C) \neq \Sigma_k^P(C)$ (Yao [12] and Hastad [5, 6]).

(4) $\forall k = 1, 2, \exists D_k\ PH(D_k) = \Sigma_k^P(D_k) \neq \Sigma_{k-1}^P(D_k)$ (Baker, Gill and Solovay [2], Heller [7]).

From the above results, the relativized polynomial time hierarchies may have quite different structures depending on the oracles. However, these results have not exhausted all possible structures of the relativized polynomial time hierarchies. For example, the following question remains open: does there exist a set D_k for each $k \geq 3$ such that (4) above holds? Furthermore, if such sets D_k exist, can we construct them to also separate $PSPACE(D_k)$ from $PH(D_k)$? In this paper, we show that the probabilistic arguments developed by Yao [12] and Hastad [5, 6] are powerful enough to construct oracles D_k with the above required properties. More precisely, we prove the following results.

(5) $\forall k \geq 1\ \exists E_k\ PSPACE(E_k) = PH(E_k) = \Sigma_k^P(E_k) \neq \Sigma_{k-1}^P(E_k)$.

(6) $\exists F_0\ PSPACE(F_0) \neq PH(F_0) = P(F_0)$.

(7) $\forall k \geq 1\ \exists F_k\ PSPACE(F_k) \neq PH(F_k) = \Sigma_k^P(F_k) \neq \Sigma_{k-1}^P(F_k)$.

The proof techniques for these results are the combination of the encoding scheme of Baker, Gill and Solovay [2] and the probabilistic arguments of Yao [12] and Hastad [5]. The main complication comes from the possible interference between the two constructions, which can be handled by using slightly different formulations of Yao and Hastad's basic lemmas.

2. Preliminaries

In this paper, all sets A are sets of strings over the alphabet $\Sigma = \{0, 1\}$. For each string x, let $|x|$ denote

*Research supported in part by NSF Grant CCR-8696135.

its *length*. Let Σ^n be the set of all strings of length n. We assume that there is a one-to-one *pairing function* $\langle,\ldots,\rangle$ that encodes an arbitrary number of strings $x_1,\ldots,x_n$ into a single string $\langle x_1,\ldots,x_n\rangle$. We assume that $|\langle x_1,\ldots,x_n\rangle| \geq \sum_{i=1}^{n} |x_i|$. For each set A, let χ_A be its characteristic function.

We assume that the reader is familiar with oracle TMs and related complexity classes. In particular, we will work on the relativized complexity classes $P(A)$, $NP(A)$, $\Sigma_k^P(A)$, $\Pi_k^P(A)$, and $PSPACE(A)$.

The relativized polynomial time hierarchy $PH(A) = \cup_{k=0}^{\infty}\Sigma_k^P(A)$ can be characterized by alternating quantifiers. Let $R(A;x)$ be a predicate over a set variable A and a string variable x. We say that $R(A;x)$ is a P^1-*predicate* if R is computable in polynomial time by a deterministic oracle machine which uses set A as the oracle and takes string x as the input. (The superscript 1 indicates that the predicate is on a type-1 object.) Let $k \geq 1$. We say $\sigma(A;x)$ is a $\Sigma_k^{P,1}$-*predicate* if there exist a P^1-predicate $R(A;x,y_1,\ldots,y_k)$ over a set variable and $k+1$ string variables, and a polynomial q, such that for all sets A and all x with $|x| = n$, $\sigma(A;x)$ is true iff

$$(\exists y_1,|y_1| \leq q(n))(\forall y_2,|y_2| \leq q(n))\cdots$$
$$(Q_k y_k,|y_k| \leq q(n))R(A;x,y_1,\ldots,y_k),$$

where $Q_k = \exists$ if k is odd, and $Q_k = \forall$ if k is even. It is well known that a set B is in $\Sigma_k^P(A)$ iff there exists a $\Sigma_k^{P,1}$-predicate σ such that for all x, $[x \in B \Leftrightarrow \sigma(A;x)]$ [9].

For any complexity class $\mathcal{C}$, a set $A \in \mathcal{C}$ is $\leq_m^P$-*complete* for $\mathcal{C}$ if for every set $B \in \mathcal{C}$, there exists a polynomial time computable function f such that for all x, $x \in B$ iff $f(x) \in A$. We will use some specific complete sets for these classes. First, we assume a fixed enumeration $\{M_i\}$ for all polynomial time oracle TMs, a fixed enumeration $\{N_i\}$ of all polynomial time nondeterministic oracle TMs, and, for each $k \geq 1$, a fixed enumeration $\{\sigma_i^k\}$ of all $\Sigma_k^{P,1}$-predicates. We assume that the ith machine M_i or N_i has its runtime bounded by the ith polynomial p_i, where $p_i(n) = n^i + i$. Also assume that the ith $\Sigma_k^{P,1}$-predicate $\sigma_i^k(A;x) \equiv (\exists y_1,|y_1| \leq q(n))(\forall y_2,|y_2| \leq q(n))\cdots(Q_k y_k,|y_k| \leq q(n))R(A;x,y_1,\ldots,y_k)$ has the property that both the length-bounding polynomial q and the runtime of the deterministic oracle TM that computes the predicate R are bounded by the ith polynomial $\{p_i\}$.

Define, for each set A, the set $K(A)$ to be $\{\langle i,z,1^j\rangle |$ the nondeterministic oracle TM N_i accepts z in j moves when A is used as the oracle$\}$. Then, it is obvious that $K(A)$ is complete for $NP(A)$. Furthermore, for any string x, the question of whether $x \in K(A)$ depends only on the set $\{y \in A \mid |y| < |x|\}$, because $x = \langle i,z,1^j\rangle$ implies $j < |x|$. (In other words, if B agrees with A on strings of length $< |x|$, then $x \in K(A)$ iff $x \in K(B)$.) We can extend this to $\Sigma_k^P(A)$-complete sets for $k > 1$. Let $K^1(A) = K(A)$ and $K^k(A) = K(K^{k-1}(A))$ for $k > 1$. Then, For each $k \geq 1$ and each set A, $K^k(A)$ is complete for $\Sigma_k^{P,A}$, and the question of whether $x \in K^k(A)$ depends only on the set $\{y \in A \mid |y| < |x|\}$ (this can be proved by induction). Define, for each set A, the set $Q(A)$ to be $\{\langle i,x,1^j\rangle |$ the ith oracle machine M_i accepts x using at most j cells when A is used as the oracle$\}$. Then $Q(A)$ is complete for the class $PSPACE(A)$ and the question of whether $x \in Q(A)$ depends only on the set $\{y \in A \mid |y| < |x|\}$.

We will deal with *circuits* of unbounded fanin which have only AND and OR gates and which have variables or its negations as inputs. A circuit is formally defined as a tree. Each interior node of the tree is attached with an AND gate or an OR gate, and has an unlimited number of child nodes. It is usually assumed that the gates alternate so that all children of an OR (or AND) gate are AND (or OR, respectively) gates. Each leaf is attached with a constant 0, a constant 1, a variable x, or a negated variable $\bar{x}$. Each circuit computes a boolean function on its variables. The *depth* of a circuit is the length of the longest path in the tree. The *size* of a circuit is the number of gates (or, the number of interior nodes) in the tree. The *fanin* of a gate is the number of children of the node. The *bottom fanin* of a circuit is the maximum fanin of a gate of the lowest level in the tree. Each circuit C has a dual circuit C' which has the same tree structure as C with each AND (OR) gate of C changed to an OR (AND, respectively) gate, and each variable x changed to its negation. The dual circuit C' computes the negation of the function computed by C.

Let V be the set of variables occurred in a circuit C. Then a *restriction* ρ of C is a mapping from V to $\{0,1,*\}$. For each restriction ρ of C, $C\lceil_\rho$ denotes the circuit C' obtained from C by replacing each variable x with $\rho(x) = 0$ by 0 and each y with $\rho(y) = 1$ by 1. Assume that ρ' is a restriction of $C\lceil_\rho$. We write $C\lceil_{\rho\rho'}$ to denote $(C\lceil_\rho)\lceil_{\rho'}$. We also write $\rho\rho'$ to denote the combined restriction on C with values $\rho\rho'(x) = \rho(x)$ if $\rho(x) \neq *$ and with values $\rho\rho'(x) = \rho'(x)$ if $\rho(x) = *$. If a restriction ρ of C maps no variable to $*$, then we say ρ is an *assignment* of C. Let ρ be a restriction of C, we say that ρ *completely determines* C if $C\lceil_\rho$ computes a constant function 0 or 1. An assignment ρ of C always completely determines the circuit C.

There are some specific circuits which are useful in our proofs. One of them is the circuits defining the functions f_k^m defined in [5]. Our definition of function f_k^m is a little different from that defined in [5]. Let C_k^m be a depth-k circuit having the following properties:

(a) the top gate of C_k^m is an OR gate with fanin $\sqrt{m}$,

(b) the fanin of all bottom gates of C_k^m is $\sqrt{m}$,

(c) the fanin of all other gates is m, and

(d) there are m^{k-1} variables each occurring exactly once in a leaf in the positive form.

Let the function computed by C_k^m be f_k^m.

The following relation between the relativized polynomial time hierarchy and small depth circuits is due to Furst, Saxe and Sipser [3].

Lemma 2.1 [3]. Let $k \geq 1$ and $q(n)$ and $r(n)$ be two polynomial functions. Let $\sigma(A;x) = (\exists y_1, |y_1| \leq q(n))(\forall y_2, |y_2| \leq q(n)) \cdots (Q_k y_k, |y_k| \leq q(n))\ R(A;x,y_1,\ldots,y_k)$ be a $\Sigma_k^{P,1}$-predicate, where $n = |x|$ and $R(A;x,y_1,\ldots,y_k)$ is computable in time $r(n)$ by a deterministic oracle TM M using oracle A. Then, for each string x, there exists a circuit C having the following properties:

(a) the depth of C is $k+1$,

(b) the fanin of each gate in C is $\leq 2^{q(n)+r(n)}$,

(c) the bottom fanin of C is $\leq r(n)$,

(d) the variables of C are strings which are queried by M on input $(x,y_1,\ldots,y_k)$ over all possible $y_1,\ldots,y_k$ of length $\leq q(n)$ and all possible oracles A, and

(e) for each set A, if we use $\chi_A(z)$ as the input value for each variable z in C then C outputs 1 iff $\sigma(A;x)$ is true.

Another interesting relation between circuits and sets in relativized polynomial time hierarchies is about complete sets $K^k(A)$.

Lemma 2.2. Let $k \geq 1$. For every x of the form $\langle i,y,1^j\rangle$, there is a circuit C such that

(a) the depth of C is $\leq 2k$,

(b) the fanin of each gate in C is $\leq 2^{|x|}$,

(c) the bottom fanin of C is $\leq |x|$,

(d) the variables of C are strings of length $\leq |x|$, and

(e) for each set A, if we use $\chi_A(z)$ as the input value for each variable z in C then C outputs 1 iff $x \in K^k(A)$.

Proof. We prove the lemma by induction on k. First, let $k=1$. Then, $x = \langle i,y,1^j\rangle \in K(A)$ iff the machine N_i accepts y in $\leq j$ moves using A as an oracle. That is, $x \in K(A)$ iff $(\exists z, |z| \leq j)R(A;y,z)$ for some P^1-predicate R which is computable in time $\leq j$. Thus the initial case of $k=1$ follows from Lemma 2.1.

Assume that $k > 1$. Let $x = \langle i,y,1^j\rangle$ be given. Then, $x \in K^k(A)$ iff the machine N_i accepts y in $\leq j$ moves using $K^{k-1}(A)$ as an oracle. By Lemma 2.1, there is a depth-2 circuit C_1 such that its top gate has fanin $\leq 2^j \leq 2^{|x|}$, its bottom fanin is $\leq j \leq |x|$, its variables are strings of length $\leq |x|$, and for each set A, if we use $\chi_{K^{k-1}(A)}(z)$ as the input value for each variable z in C_1 then C_1 outputs 1 iff $x \in K^k(A)$.

Now, by the inductive hypothesis, for each variable z in C_1 of the form $\langle i_1,u,1^{j_1}\rangle$, there is a circuit C_z of depth $2(k-1)$ such that

(a) the fanin of each gate of C_z is $\leq 2^{|z|}$,

(b) the bottom fanin of C_z is $\leq |z|$,

(c) the variables in C_z are of length $\leq |z|$, and

(d) for any set A, if we use $\chi_A(w)$ as the input value for each variable w in C_z, then C_z outputs 1 iff $z \in K^{k-1}(A)$.

For each variable z in C_1 not of such a form, let C_z be the constant 0 (because $z \notin K^{k-1}(A)$). Replace each variable z in C_1 by the circuit C_z. We obtain the desired circuit C. □

3. Relativized Hierarchy Having k Levels

In this section, we prove that for each $k \geq 1$, there exists an oracle A such that $\Sigma_k^P(A) = \Pi_k^P(A) \neq \Sigma_{k-1}^P(A)$. We need a lower bound result on small depth circuits. Hastad [5] has proved that there exists a function f_k^n computable by a polynomial-size depth-k circuit but not by any depth-k circuit having subexponential size and small bottom fanin. The following lemma is a stronger form of this result. It states that no depth-k circuit with small bottom fanin can compute any of an exponential number of f_k^n functions. The main idea of the proof is the same as that of Hastad's proof. We only give a sketch here, and leave the complete proof to the full paper.

Recall that C_k^n is a circuit defining the function f_k^n. Let $\mathrm{CIR}(k,t)$ be the class of depth-k circuits which have size $\leq 2^t$ and bottom fanin $\leq t$.

Lemma 3.1. For every $k \geq 2$ there exists a constant n_k such that the following holds for all $n > n_k$. Let $t = n^{\log n}$, $m < 2^t$, and $C_0, C_1, \ldots, C_m$ be $m+1$ circuits each defining a $f_k^{2^n}$ function, with their variables pairwisely disjoint. Let C be a circuit in $\mathrm{CIR}(k,t)$. Then, there exists a restriction ρ on C such that ρ completely determines C but it does not completely determine any C_i, $0 \leq i \leq m$.

Sketch of Proof. In Hastad's proof of the exponential lower bound for depth-k, small bottom-fanin circuits for the f_k^n function, the following result was developed.

Lemma 3.1.1. Let C' be a circuit computing a $f_k^{2^n}$ function and let q be a real number. Then there exists a probability distribution R of restrictions on variables in C' (called $R_{q,B}^+$ or $R_{q,B}^-$ in [5]) such that for a random restriction ρ from R,

(i) for any circuit $G \in CIR(1,t)$ with a top AND gate, the probability that $G\lceil_\rho$ is not equivalent to a circuit $H \in CIR(1,t)$ with a top OR gate is $\leq \delta^t$, where $\delta < 6qt$, and

(ii) the probability that every subcircuit D of the lowest two levels of C' has the property that $D\lceil_\rho$ contains at least $2^{n/2}$ many undetermined children is $\geq 2/3$.

Moreover, the above lemma holds even if C' has a larger fanin, as long as the fanin is smaller than 2^t. Note that if we let C' be AND of all circuits

$C_0, C_1, \cdots, C_m$, then we may apply the above lemma to C' and obtain a restriction such that (a) by Lemma 3.1.1(i), $C\lceil_\rho \in CIR(k-1,t)$ and (b) by Lemma 3.1.1(ii), each $C_i\lceil_\rho$ computes a $f_{k-1}^{2^n}$ function. Thus, an induction proof can be used as in Hastad's original proof. □

Theorem 3.2. For each $k \geq 1$, there exists a set A such that $\Sigma_k^P(A) = \Pi_k^P(A) \neq \Sigma_{k-1}^P(A)$.

Proof. The case $k = 1$ has been proven by Baker, Gill and Solovay [2]. We assume that $k \geq 2$. (Actually, the case $k = 2$ has been proven by Baker and Selman [3] and Heller [7].) Let

$$L_k(A) = \{1^n \mid (\exists v_1, |v_1| = n)(\forall v_2, |v_2| = n) \cdots (Q_k v_k, |v_k| = n)\ 1^n v_1 v_2 \ldots v_k \in A\}.$$

Note that $L_k(A)$ is in $\Sigma_k^P(A)$.

The construction of the oracle A will be done by stage. At each stage $\alpha = (k+1)n + 1$, we will satisfy the requirement

$R_{0,n}$: for all strings u of length n, $u \notin K^k(A) \Leftrightarrow (\exists v_1, |v_1| = n)(\forall v_2, |v_2| = n) \cdots (Q_k v_k, |v_k| = n)\ 0uv_1v_2 \ldots v_k \in A$.

At stage $\alpha = (k+1)n$, we will try to find a suitable i such that the following requirement is satisfied with $n_i = n$.

$R_{1,i}$: there exists an n_i such that $1^{n_i} \in L_k(A)$ iff $\sigma_i^{k-1}(A; 1^{n_i})$ is false.

The requirement $R_0 = \bigwedge_{n=1}^{\infty} R_{0,n}$ states that $K^k(A)$ is in $\Pi_k^P(A)$, and hence $\Sigma_k^P(A) = \Pi_k^P(A)$. The requirement $R_1 = \bigwedge_{i=1}^{\infty} R_{1,i}$ states that $L_k(A)$ is not in $\Sigma_{k-1}^P(A)$. Therefore a set A satisfying all requirements has the property $\Sigma_k^P(A) = \Pi_k^P(A) \neq \Sigma_{k-1}^P(A)$.

The main difficulty of the construction is that when we try to satisfy requirement $R_{1,i}$ in stage $\alpha = (k+1)n$, we may have to simulate some oracle machine M which may query about strings of length longer than α. We cannot arbitrarily assign answers to the queries made by M because such an assignment may conflict with requirement $R_{0,m}$ for some $m > n$. What we need is an assignment of answers to queries which does not conflict with future constructions at stage $\alpha' > \alpha$. The existence of such an assignment will be proved by using Lemma 3.1.

In each stage α, we will define a set $A(\alpha)$ and a set $\overline{A(\alpha)}$ and an integer β_α. Sets $A(\alpha)$ and $\overline{A(\alpha)}$ are defined so that $A(\alpha)$ is always an extension of $A(\alpha-1)$, $\overline{A(\alpha)}$ is always an extension of $\overline{A(\alpha-1)}$, and $A(\alpha) \cap \overline{A(\alpha)} = \emptyset$. Also, in stage α we don't add any string of length $< \alpha$ to $A(\alpha)$ or $\overline{A(\alpha)}$. β_α is defined to be the length of the longest string which is added to $A(m)$ or $\overline{A(m)}$ in stages $m \leq \alpha$. When $\alpha > \beta_{\alpha-1}$, the construction in stage α can be done without interfering with the constructions made in earlier stages.

Prior to stage 1, assume that $A(0) = \overline{A(0)} = \emptyset$, and let $\beta_0 = 1$. Let all integers i be uncanceled.

Stage α, where α does not have the form $\alpha = (k+1)n + 1$ or $\alpha = (k+1)n$. Do nothing. Let $A(\alpha) := A(\alpha-1)$, $\overline{A(\alpha)} := \overline{A(\alpha-1)}$ and $\beta_\alpha := \beta_{\alpha-1}$.

Stage $\alpha = (k+1)n$. Let i be the least integer that is not yet canceled. If $\alpha \leq \beta_{\alpha-1}$ or $n \leq n_k$ (n_k is the constant defined in Lemma 3.1) or $kp_i(n) + 1 \geq t = n^{\log n}$, then do nothing. Let $A(\alpha) := A(\alpha-1)$, $\overline{A(\alpha)} := \overline{A(\alpha-1)}$ and $\beta_\alpha := \beta_{\alpha-1}$.

If $\alpha > \beta_{\alpha-1}$ and $n > n_k$ and $kp_i(n) + 1 < t = n^{\log n}$, then consider the following circuits:

(1) For each u of length $n \leq |u| \leq p_i(n)$, the circuit C_u of depth k is defined as follows:

(a) the top gate of C_u is an OR gate,

(b) the fanin of each gate of C_u is $2^{|u|}$,

(c) the variables of C_u are exactly those in $0u\Sigma^{k|u|}$, each occurring positively in exactly one leaf of C_u in the increasing order (under the lexicographic order).

There are totally $\leq 2^{p_i(n)} < 2^t$ many such circuits C_u. Note that each circuit C_u has the property that for all sets A, if we use $\chi_A(y)$ as the input value for each variable y then C_u outputs 1 iff $(\exists v_1, |v_1| = |u|)(\forall v_2, |v_2| = |u|) \cdots (Q_k v_k, |v_k| = |u|)\ 0uv_1v_2 \ldots v_k \in A$. Also note that each circuit C_u contains a subcircuit computing a function $f_k^{2^n}$.

(2) The circuit C_0 has the same tree structure as the circuit C_u defined above, with $|u| = n$, except that the variables of C_0 are those in $1^n\Sigma^{kn}$. Note that if we use $\chi_A(y)$ as the input value for each variable y then C_0 outputs 1 iff $1^n \in L_k(A)$.

(3) The circuit C is the circuit associated with the $\Sigma_{k-1}^{P,1}$-predicate $\sigma_i^{k-1}(A; 1^n)$ as defined in Lemma 2.1, with the restriction that each variable y of length $< \alpha = (k+1)n$ is replaced by the constant value $\chi_{A(\alpha-1)}(y)$. In particular, C has depth k, has size $\leq 2^{kp_i(n)+1}$, and has the bottom fanin $\leq p_i(n)$. The variables in C are strings of length $\leq p_i(n)$. For each set A which agrees with $A(\alpha-1)$ on strings of length $< \alpha$, if we use $\chi_A(y)$ as the input value for each variable y then C outputs 1 iff $\sigma_i^{k-1}(A; 1^n)$ is true.

It is easy to see that each circuit C_u or C_0 contains a subcircuit C_u' or C_0', respectively, which defines a $f_k^{2^n}$ function. Choose a restriction ρ such that for all u, $C_u' = C_u\lceil_\rho$ computes exactly a $f_k^{2^n}$ function and $C_0' = C_0\lceil_\rho$ computes exactly a $f_k^{2^n}$ function. We observe that by the choice of n such that $kp_i(n) + 1 < t = n^{\log n}$, $C \in \mathrm{CIR}(k,t)$. So, $C' = C\lceil_\rho$ is also in $\mathrm{CIR}(k,t)$. Furthermore, the number of circuits C_u' is $< 2^t$. Thus, we can apply Lemma 3.1 to the circuits C_0' and C_u', $n \leq |u| \leq p_i(n)$, and the circuit C' to obtain a restriction ρ' of C' such that ρ' completely determines the circuit C' but not circuit C_0' nor any circuit C_u', $n \leq |u| \leq p_i(n)$. Finally, we find an assignment ρ''

of variables of $C_0'\lceil_{\rho'}$ such that $C_0'\lceil_{\rho'\rho''}$ computes a constant function 1 iff $C'\lceil_{\rho'}$ computes a constant function 0. Note that ρ'' only assigns values to strings in $1^n\Sigma^{kn}$, and so none of $C_u'\lceil_{\rho'}$ is completely determined by ρ''.

Define $A(\alpha) := A(\alpha-1) \cup \{w \mid \rho\rho'\rho''(w) = 1\}$ and $\overline{A(\alpha)} := \overline{A(\alpha-1)} \cup \{w \mid \rho\rho'\rho''(w) = 0\}$. Let $\beta_\alpha = \max\{\alpha, p_i(n)+1\}$ and cancel i. This completes stage $\alpha = (k+1)n$.

Stage $\alpha = (k+1)n+1$. For each u of length n, we determine whether $u \in K^k(A(\alpha-1))$. Then, we find a subset $B \subseteq \{w \in 0u\Sigma^{kn} \mid w \notin A(\alpha-1) \cup \overline{A(\alpha-1)}\}$ such that $u \notin K^k(A(\alpha-1)) \Leftrightarrow (\exists v_1, |v_1| = n)(\forall v_2, |v_2| = n)\cdots(Q_k v_k, |v_k| = n)\ 0uv_1v_2\ldots v_k \in A(\alpha-1) \cup B$. (We will show that such a set B always exists.) Let $A(\alpha) = A(\alpha-1) \cup B$ and $\overline{A(\alpha)} = \overline{A(\alpha-1)}$. Let $\beta_\alpha = \max\{\alpha, \beta_{\alpha-1}\}$. Stage $\alpha = (k+1)n+1$ is complete when we finish the above construction for each u of length n.

Let $A = \cup_{\alpha=1}^{\infty} A(\alpha)$. First, we claim that in each stage $\alpha = (k+1)n+1$, for each u of length n, the set B can be found.

Proof of Claim. If none of strings in $0u\Sigma^{kn}$ has been assigned to $A(\alpha-1)$ or $\overline{A(\alpha-1)}$, then certainly such a set B exists. Assume that some strings in $0u\Sigma^{kn}$ have been assigned to $A(\alpha-1)$ or $\overline{A(\alpha-1)}$. Then, by the choice of β_α, there is at most one stage $\alpha' = (k+1)m < \alpha$ in which these assignments are made.

In that stage, the assignments are made such that the corresponding circuit C_u, after applying the restriction $\rho\rho'\rho''$, is not completely determined. Note that the circuit C_u and the predicate $\tau(A;u) \equiv (\exists v_1, |v_1| = n)(\forall v_2, |v_2| = n)\cdots(Q_k v_k, |v_k| = n)\ 0uv_1v_2\ldots v_k \in A$ has the relation that when assigning value $\chi_A(y)$ to each variable y, circuit C_u outputs 1 iff the predicate $\tau(A;u)$ is true. The fact that the restriction $\rho\rho'\rho''$ does not completely determine C_u implies that there exist assignments ρ_0 and ρ_1 such that $C_u\lceil_{\rho\rho'\rho''\rho_0}$ outputs 0 and $C_u\lceil_{\rho\rho'\rho''\rho_1}$ outputs 1. Let $B_0 = \{w \in 0u\Sigma^{kn} \mid \rho\rho'\rho''(w) = *, \rho_0(w) = 1\}$ and $B_1 = \{w \in 0u\Sigma^{kn} \mid \rho\rho'\rho''(w) = *, \rho_1(w) = 1\}$. Then, B_0 and B_1 are disjoint from $A(\alpha-1)$ and $\overline{A(\alpha-1)}$ and $\tau(A(\alpha-1) \cup B_0; 1^n) = 0$ and $\tau(A(\alpha-1) \cup B_1; 1^n) = 1$. This proves the claim. □

Next we observe that after stage α, we never add any string of length $\leq \alpha$ to $A(\alpha)$ or $\overline{A(\alpha)}$. From this observation and the fact that the question of $x \in K^k(A)$ does not depend on the strings of length $\geq |x|$, we see that each stage $(k+1)n+1$ satisfies requirement $R_{0,n}$.

Finally, for each i, we observe that eventually we will cancel it in some stage $\alpha = (k+1)n$, since the inequality $kp_i(n)+1 < t = n^{\log n}$ holds for almost all n. In that stage, we add strings to $A(\alpha)$ or $\overline{A(\alpha)}$ (by $\rho\rho'\rho''$) so that when we use $\chi_A(y)$ as the input value for each variable y, the circuits C and C_0 are completely determined and circuit C outputs 1 iff circuit C_0 outputs 0. By the relation between circuit C and predicate $\sigma_i^{k-1}(A;1^n)$ and the relation between circuit C_0 and the predicate $1^n \in L_k(A)$, we know that $\sigma_i^{k-1}(A;1^n)$ is true iff $1^n \notin L_k(A)$. This shows that the requirement $R_{1,i}$ is satisfied by A and $n_i = n$. This completes the proof of Theorem 3.2. □

The above proof can easily be modified to construct an oracle A such that $PSPACE(A) = \Sigma_k^P(A) \neq \Sigma_{k-1}^P(A)$. All we need to do is to replace the set $K^k(A)$ by the set $Q(A)$ which is $\leq_m^P$-complete for $PSPACE(A)$.

Corollary 3.3. For each $k \geq 1$, there exists a set A such that $PSPACE(A) = \Sigma_k^P(A) \neq \Sigma_{k-1}^P(A)$.

4. Relativized Hierarchies and PSPACE

In this section, we show that for each $k \geq 1$, there exists an oracle A such that $\Sigma_k^P(A) = \Pi_k^P(A) \neq \Sigma_{k-1}^P(A)$ and also $PSPACE(A) \neq PH(A)$. The proof also uses the lower bound results on small depth circuits developed by Yao [12] and Hastad [5]. The following lemma is from [5]. We say a circuit C computes the parity of n inputs if C has n variables and for all inputs to those variables, C outputs 1 iff the number of 1's in the input is odd.

Lemma 4.1 [5]. There exist an integer n_0' and a real number $\epsilon > 0$ such that for any $k > 0$ and any $n > (n_0')^k$, no depth-k circuit C of $\leq 2^{\epsilon n^{1/(k-1)}}$ gates can compute the parity of n inputs.

Corollary 4.2. For any constant c, there is an n_c' such that for all $n > n_c'$, no depth-k, $k = c\log\log n$, circuit C of $\leq 2^{\epsilon n^{1/(k-1)}}$ gates can compute the parity of n inputs.

Proof. Let n_c' be the smallest integer m such that $m > (n_0')^{c\log\log m}$, where n_0' is the absolute constant of Lemma 4.1. □

We first consider the simplest case that the relativized polynomial time hierarchy collapses to the class P.

Theorem 4.3. There exists a set A such that $PSPACE(A) \neq NP(A) = P(A)$.

Proof. Recall that $\{N_i\}$ is an enumeration of all polynomial time nondeterministic oracle TMs, and the machine N_i has its runtime bounded by polynomial $p_i(n)$. Without loss of generality, we assume that $p_i(n) \leq n^i$. Let $L_{\text{odd}}(A) = \{1^n \mid$ the number of strings of length n which are in A is odd$\}$. Note that $L_{\text{odd}}(A)$ is in $PSPACE(A)$.

The construction of A is done by stage. At stage $n = 2t$, we want to satisfy the requirement

$R_{0,t}$: For each u of length t, $u \in K(A)$ iff $0^t u \in A$.

At stage $n = 2t+1$, we will try to find a suitable i such that the following requirement is satisfied with $m_i = n$.

$R_{1,i}$: there exists an m_i such that $1^{m_i} \in L_{\text{odd}}(A)$ iff N_i rejects 1^{m_i}.

The requirement $R_0 = \bigwedge_{t=1}^{\infty} R_{0,t}$ states that $K(A)$ is in $P(A)$, and hence $NP(A) = P(A)$. The requirement

$R_1 = \bigwedge_{i=1}^{\infty} R_{1,i}$ states that $L_{\text{odd}}(A)$ is not in $NP(A)$. Therefore a set A satisfying all requirements has the property $PSPACE(A) \neq NP(A) = P(A)$.

In each stage n, we will define a set $A(n)$ and a set $\overline{A(n)}$ and an integer β_n. Sets $A(n)$ and $\overline{A(n)}$ are defined so that $A(n)$ is always an extension of $A(n-1)$, $\overline{A(n)}$ is always an extension of $\overline{A(n-1)}$, and $A(n) \cap \overline{A(n)} = \emptyset$. Also, in stage n we don't add any string of length $< n$ to $A(n)$ or $\overline{A(n)}$. β_n is defined to be the length of the longest string which is added to $A(m)$ or $\overline{A(m)}$ in stages $m \leq n$. When $n > \beta_{n-1}$, the construction in stage n can be done without interfering with the constructions made in earlier stages.

Prior to stage 1, assume that $A(0) = \overline{A(0)} = \emptyset$, and let $\beta_0 = 2$. Let all integers i be uncanceled.

Stage $n = 2t$, $t > 0$. We will satisfy requirement $R_{0,t}$. For each string u of length t, we determine whether $u \in K(A(n-1))$. Then, we let $A(n) = A(n-1) \cup \{0^t u\}$ and $\overline{A(n)} = \overline{A(n-1)}$ if $u \in K(A(n-1))$; and $A(n) = A(n-1)$ and $\overline{A(n)} = \overline{A(n-1)} \cup \{0^t u\}$ if $u \notin K(A(n-1))$. After this is done for all u of length t, let $\beta_n = \max\{n, \beta_{n-1}\}$.

Stage $n = 2t+1$. Let i be the least integer which has not been canceled. Let $m = 2i \log n$. If $n \leq \beta_{n-1}$ or $2^n \leq n'_{2i}$ or $\epsilon 2^{n/(m-1)} \leq (m+1)p_i(n)$ (where n'_{2i} is the constant defined in Corollary 4.2), then do nothing. Let $A(n) = A(n-1)$, $\overline{A(n)} = \overline{A(n-1)}$, and $\beta_n = \beta_{n-1}$.

If $n > \beta_{n-1}$ and $2^n > n'_{2i}$ and $\epsilon 2^{n/(m-1)} > (m+1)p_i(n)$, then consider the machine N_i on input 1^n. From Lemma 2.1, we know that for machine N_i and input 1^n, there is a circuit C of depth 2 such that

- (a) the top gate of C is an OR gate with fanin $\leq 2^{p_i(n)}$,
- (b) the bottom fanin of C is $\leq p_i(n)$,
- (c) the variables in C are strings which are queried by N_i on input x over all possible oracles A, and
- (d) for each set A, if we use $\chi_A(y)$ as the input value for each variable y then C outputs 1 iff N_i^A accepts 1^n.

We are going to modify this circuit to a new circuit C' having the following properties. During the modification of C to C', we also define a set B.

- (a') C' has depth $\leq m = 2i \log n$.
- (b') The number of gates in C' is $\leq 2^{(m+1)p_i(n)}$.
- (c') The variables of C' are strings of length n.

(Circuit C' and set B also have nice properties related to the computation of $N_i(1^n)$ and set $L_{\text{odd}}(A)$. We will prove them later.)

Recall that $m = 2i \log n$. The modification of the circuit C is done in $m/2$ steps. Let $C_1 = C$. In each step $j \leq m/2 - 1$, assume that a circuit C_j is given. For each variable y in C_j which is of length $> n$ and is of the form $y = 0^{|u|}u$ for some u, we do the following.

By Lemma 2.2, there is a depth-2 circuit C'_y such that

- (a'') the top fanin of C'_y is $\leq 2^{|u|}$,
- (b'') the bottom fanin of C'_y is $\leq |u|$,
- (c'') the variables in C'_y are of length $\leq |u|$, and
- (d'') for any set A, if we use $\chi_A(z)$ as the input value for each variable z in C'_y then C'_y outputs 1 iff $u \in K(A)$.

Replace y by the circuit C'_y. (That is, the leaf node of C_j with variable y is replaced by the tree C'_y, and the leaf node with the negation $\bar{y}$ of variable y is replaced by the tree of the dual circuit of C'_y.) Let C_{j+1} be the new circuit with all such variables y in C_j replaced by circuit C'_y.

Assume that a variable $y = 0^{|u|}u$ in C_j is replace by C'_y. Then, by Lemma 2.2, each variable w in circuit C'_y is of length $|w| \leq |u| \leq |y|/2$. Note that all variables y in $C_1 = C$ have length $\leq p_i(n) = n^i$. So, after $\log(p_i(n)) - 1 = i \log n - 1 = m/2 - 1$ steps, there is no variable y of length $> n$ in $C_{m/2}$ which is of the form $0^{|u|}u$ for any u. (Note that $n > \beta_0 = 2$.)

In step $m/2$, we replace each variable y in $C_{m/2}$ which is of length $< n$ by the constant $\chi_{A(n-1)}(y)$, and replace each variable y in $C_{m/2}$ which is of length $> n$ but not of the form $0^{|u|}u$ by a constant 0. (Also, each $\bar{y}$ is replaced by the opposite value.) Let C' be the resulting circuit, and let $B = \{y \mid |y| > n, y \text{ occurs in } C_{m/2} \text{ and } y \text{ is not of the form } 0^{|u|}u\}$.

We verify that the final circuit C' satisfies the properties (a')—(c') listed above. First, in each step $j \leq m/2-1$, we replaced some variables in C_j by depth-2 circuits. So, C_{j+1} has depth two plus the depth of C_j. Thus, $C_{m/2}$ has depth at most m. Since we only replaced variables in $C_{m/2}$ by constants, the final circuit C' also has depth at most m.

Next, to check (b'), we note that every gate in C has fanin $\leq 2^{p_i(n)}$. Furthermore, by Lemma 2.2, every gate in circuit C'_y has the same bound for its fanin. Therefore, without combining adjacent gates of the same type, each gate of C' has fanin $\leq 2^{p_i(n)}$. That is, the total number of gates in C' is at most $(2^{p_i(n)})^{m+1} = 2^{(m+1)p_i(n)}$.

For condition (c'), we note that all variables of length not equal to n are replaced by constants or other circuits. So, the only variables left in C' are of length n.

Now from the inequality $\epsilon 2^{n/(m-1)} > (m+1)p_i(n)$, we can apply Corollary 4.2 to circuit C' and conclude that C' does not compute the parity of the 2^n variables $x \in \Sigma^n$. So, we can find a set $D \subseteq \Sigma^n$ such that $1^n \in L_{\text{odd}}(D)$ iff C' outputs 0 when variables z are given values $\chi_D(z)$.

Define $A(n) = A(n-1) \cup D$, $\overline{A(n)} = \overline{A(n-1)} \cup B$, $\beta_n = \max\{n, p_i(n)+1\}$, and cancel i. Stage $n = 2i+1$

is complete.

Let $A = \cup_{n=1}^{\infty} A_n$. We need to verify that A satisfies every requirement $R_{0,t}$, $t > 0$, and $R_{1,i}$, $i > 0$. For requirement $R_{0,t}$, $t > 0$, we note that in Stage $n = 2t$, we have assigned all strings $0^t u$, $|u| = t$, to $A(n)$ or $\overline{A(n)}$ such that $u \in K(A(n-1)) \Leftrightarrow 0^t u \in A(n)$. Since A agrees with $A(n-1)$ on strings of length $< n$, we have $u \in K(A(n-1)) \Leftrightarrow u \in K(A)$. Furthermore, once a string $0^t u$ is added to $A(n)$ or $\overline{A(n)}$, its membership in A is never changed in later stages. So, $0^t u \in A(n) \Leftrightarrow 0^t u \in A$. The only thing left to check is that in earlier stages $n' < n$, no string of the form $0^t u$, with $|u| = t$, has been put in $A(n')$ or $\overline{A(n')}$. This is true because in an even stage $n' = 2t'$ we never add any string of length longer than n' to $A(n')$ or $\overline{A(n')}$ and in an odd stage $n' = 2t' + 1$ we only add strings of length n' or strings which are not of the form $0^{|u|}u$ to $A(n')$ or $\overline{A(n')}$.

To verify requirement $R_{1,i}$, we note that the inequality $\epsilon 2^{n/(m-1)} > (m+1)p_i(n)$ is satisfied by almost all integer n. Therefore, each integer i will eventually be canceled. Assume that i is canceled in stage $n = 2t+1$. We want to show that $1^n \in L_{\text{odd}}(A)$ iff N_i^A rejects 1^n. To show this, we note that circuit C' constructed in stage n satisfies the following property.

(d') When we use $\chi_A(w)$ as the input value for each variable w in C', C' outputs 1 iff N_i^A accepts 1^n. (Note that the set A here is the fixed set defined by $A = \cup_{n=0}^{\infty} A_n$ which satisfies requirements $R_{0,t}$ for all $t > 0$.)

This property can be proved by induction by showing that all circuits C_j, $1 \le j \le m/2$, constructed in stage n satisfy it. We omit the details.

Now, observe that, in stage n, the set D is chosen such that $1^n \in L_{\text{odd}}(D)$ iff C' outputs 0 when each variable z is given the input value $\chi_D(z)$. Since we have added set D to A, the following holds when each variable z is given the input value $\chi_A(z)$:

$$1^n \in L_{\text{odd}}(A) \Leftrightarrow 1^n \in L_{\text{odd}}(D)$$
$$\Leftrightarrow C' \text{ outputs } 0 \Leftrightarrow N_i^A \text{ rejects } 1^n.$$

Thus requirement $R_{1,i}$ is satisfied when i is canceled. This completes the proof of Theorem 4.3. □

Now we extend Theorem 4.3 to more general cases.

Theorem 4.4. For each $k > 0$, there exists a set A such that $PSPACE(A) \neq \Sigma_k^P(A) = \Pi_k^P(A) \neq \Sigma_{k-1}^P(A)$.

Proof. The proof is a combination of the constructions in the proofs of Theorems 3.2 and 4.3. We only give an outline of the proof. We consider only the cases when $k > 1$. (The case $k = 1$ needs a different but simpler proof, we omit its proof.) We will construct a set A to satisfy three sets of requirements:

$R_{0,n}$: for all strings u of length n, $u \notin K^k(A) \Leftrightarrow (\exists v_1, |v_1| = n)(\forall v_2, |v_2| = n)\cdots(Q_k v_k, |v_k| = n)\ 0uv_1v_2\ldots v_k \in A$.

$R_{1,i}$: there exists an n_i such that $1^{n_i} \in L_k(A)$ iff the ith $\Sigma_{k-1}^{P,1}$-predicate $\sigma_i^{k-1}(A; 1^{n_i})$ is false.

$R_{2,j}$: There exists an m_j such that $1^{m_j} \in L_{\text{odd}}(A)$ iff the jth $\Sigma_k^{P,1}$-predicate $\sigma_j^k(A; 1^{m_j})$ is false.

As we argued before, the requirement $R_0 = \bigwedge_{n=1}^{\infty} R_{0,n}$ implies $\Sigma_k^P(A) = \Pi_k^P(A)$ and the requirement $R_1 = \bigwedge_{i=1}^{\infty} R_{1,i}$ implies that $\Sigma_k^P(A) \neq \Sigma_{k-1}^P(A)$. Furthermore, the requirement $R_2 = \bigwedge_{j=2}^{\infty} R_{2,j}$ implies that $L_{\text{odd}}(A) \notin \Sigma_k^P(A)$ and hence $PSPACE(A) \neq \Sigma_k^P(A)$.

In our construction, we will consider three types of stages. At stage $\alpha = (k+1)n$, we try to satisfy requirement $R_{1,i}$ for the integer i such that $2i$ is the least uncanceled integer. (If the least uncanceled integer is an odd integer, then do nothing.) Assume that $2i$ is the least uncanceled integer and α is sufficiently large. Then, we satisfy $R_{1,i}$, with $n_i = n$, by doing almost the same thing as in stage α of the construction in the proof of Theorem 3.2; the only difference is that at the end of the stage, we turn on a flag: $F =$ **true**, and cancel the integer $2i$ (instead of i).

At stage $\alpha = (k+1)n+2$, we try to satisfy requirement $R_{2,j}$ for the integer j such that $2j+1$ is the least uncanceled integer. The action in this stage is similar to the construction in an odd stage of the proof of Theorem 4.3.

That is, when α is sufficiently large and $\alpha > \beta_{\alpha-1}$, we consider the jth $\Sigma_k^{P,1}$-predicate $\sigma_j^k(A; 1^\alpha)$. By Lemma 2.1, there is a circuit C associated with $\sigma_j^k(A; 1^\alpha)$ having the following properties:

(a) the depth of C is $\le k+1$,

(b) each gate of C has fanin $\le 2^{p_j(\alpha)}$,

(c) the bottom fanin of C is $\le p_j(\alpha)$,

(d) the variables in C are strings of length $\le p_j(\alpha)$, and

(e) for every set A, if every variable y in C is given the value $\chi_A(y)$, then C outputs 1 iff $\sigma_j^k(A; 1^\alpha)$ is true.

Similarly to the construction in Theorem 4.3, we modify circuit C into C' such that it satisfies the following properties:

(a') the depth of C' is $\le m = 2k \log p_j(\alpha)$,

(b') the number of gates in C' is $\le 2^{(m+1)p_j(\alpha)}$, and

(c') all variables in C' are strings of length α.

The modification of C into C' is similar to the modification in an odd stage of the construction in Theorem 4.3. The main idea is to replace each variable y of length $> \alpha$ and of the form $y = 0uv$, $|v| = k|u|$, by the dual circuit of the circuit C'_y, where C'_y is a depth-$(2k)$ circuit satisfying conditions of Lemma 2.2 (with respect to the string u). (The reason of using the dual circuit of C'_y instead of C'_y itself is that we want to get the relation $u \notin K^k(A) \Leftrightarrow y \in A$.) Note that for each y of the form

$0uv$, with $|v| = k|u|$, the variables in circuit C'_y are of length $\leq |u| \leq |y|/(k+1) \leq |y|/2$. Thus, as argued in the proof of Theorem 4.3, the circuit C will be expanded into a new circuit with no variable y of length $> \alpha$ having the form $y = 0uv$, $|v| = k|u|$, and with depth $\leq 2k \log p_j(\alpha)$. By replacing all other variables of length $> \alpha$ by constant 0 (and form the set B) and all other variables z of length $< \alpha$ by value $\chi_{A(\alpha-1)}(z)$, we obtain a new circuit C' satisfying the above conditions (a')—(c'). Choose a set $D \subseteq \Sigma^\alpha$ such that $1^n \in L_{\text{odd}}(D)$ iff C' outputs 0 when each variable z is given value $\chi_D(z)$. Let $A(\alpha) = A(\alpha-1) \cup D$, $\overline{A(\alpha)} = \overline{A(\alpha-1)} \cup B$, and $\beta_\alpha = \max\{\alpha, p_j(\alpha)+1\}$. Finally, we cancel integer $2j+1$ and turn off the flag: $F =$ **false**.

We can prove, similarly to the case in the proof of Theorem 4.3, that circuit C' satisfies the following property:

(d') Assume that A is an extension of $A(\alpha)$ and $A \cap \overline{A(\alpha)} = \emptyset$. Also assume that for all u of length $n < |u| \leq p_j(\alpha)/(k+1)$, $u \notin K^k(A) \Leftrightarrow (\forall v, |v| = k|u|)0uv \in A$. Then, if we give $\chi_A(z)$ as input value for each variable z in C', then C' outputs 1 iff $\sigma_j^k(A; 1^\alpha)$ is true.

Note that, in the above, without the assumption that $u \notin K^k(A) \Leftrightarrow (\forall v, |v| = k|u|)0uv \in A$, the circuit C' may not compute exactly the predicate $\sigma_j^k(A; 1^\alpha)$, because we replaced each variable y of the form $0uv$ by a circuit computing the value $1 - \chi_{K^k(A)}(u)$ which is not necessarily equal to $\chi_A(y)$ even if set A satisfies the requirements $R_{0,|u|}$. Therefore, we need this stronger assumption. We will show later that this stronger assumption can indeed be satisfied in our construction.

At stage $\alpha = (k+1)n + 1$, we satisfy requirement $R_{0,n}$. For each u of length n, we determine whether $u \in K^k(A(\alpha-1))$, and try to find a set $B \subseteq 0u\Sigma^{kn}$ such that

$$(*) \qquad u \notin K^k(A(\alpha-1)) \Leftrightarrow (\exists v_1, |v_1| = n) \cdots (Q_k v_k, |v_k| = n)0uv_1 \ldots v_k \in B.$$

This set B will be determined as follows: if the flag F is true, then search for a set B satisfying both $(*)$ and $B \cap (A(\alpha-1) \cup \overline{A(\alpha-1)}) = \emptyset$; if the flag is false, then let $B = \emptyset$ when $u \in K^k(A(\alpha-1))$ and $B = 0u\Sigma^{kn}$ when $u \notin K^k(A(\alpha-1))$. Finally, let $A(\alpha) = A(\alpha-1) \cup B$, $\overline{A(\alpha)} = \overline{A(\alpha-1)}$.

Note that the flag F is turned *on* whenever an even integer $2i$ is canceled. When $2i$ is canceled in stage α', some strings of the form $0uv$, $|u| = n$ and $|v| = kn$, may have been added to set $A(\alpha')$ or $\overline{A(\alpha')}$. However, later in stage $\alpha = (k+1)n + 1$, before the flag F is turned off, a set B satisfying both $(*)$ and $B \cap (A(\alpha-1) \cup \overline{A(\alpha-1)}) = \emptyset$ can always be found, as proved by Lemma 3.1. (Note the by setting $\beta_{\alpha'}$ to be the length of the longest string added to $A(\alpha')$ or $\overline{A(\alpha')}$, we know that the flag F will not be turned off until in some stage $\alpha > \beta_{\alpha'}$.)

The flag F will be turned *off* when we cancel an odd integer $2j+1$. Suppose we cancel $2j+1$ at stage α'', then we must have $\alpha'' > \beta_{\alpha''-1}$, and hence no strings of length $\geq \alpha''$ are in set $A(\alpha''-1)$ or in set $\overline{A(\alpha''-1)}$. Thus, in later stage $\alpha = (k+1)n + 1$, before the flag is turned on, we have $0u\Sigma^{kn} \cap (A(\alpha-1) \cup \overline{A(\alpha-1)}) = \emptyset$ for all u of length n. So, in stage α, the choice of the set B can be made free from interference of earlier stages.

This completes the construction of set A. Note that by setting β_α to be the length of the longest string added to $A(m)$ or $\overline{A(m)}$ for all $m \leq \alpha$, we prevent the possible interference between stages $(k+1)n$ and stages $(k+1)n+2$. By the discussions in the construction, set A satisfies requirements $R_{0,n}$ for all n and $R_{1,i}$ for all i. The only thing left to check for requirement $R_{2,j}$ is that set A satisfies the stronger assumption of (d'): for all u of length $n < |u| \leq p_j(\alpha)/(k+1)$, $u \notin K^k(A) \Leftrightarrow (\forall v, |v| = k|u|)0uv \in A$. We note that by the definition of β_α the flag F is off at stage $\alpha' = (k+1)|u| + 1$. Therefore, in stage α', we satisfy the requirement $R_{0,|u|}$ by letting $0u\Sigma^{k|u|} \subseteq A$ if $u \notin K^k(A)$ and $0u\Sigma^{k|u|} \cap A = \emptyset$ if $u \in K^k(A)$. This shows that the assumption, and hence the conclusion, of property (d') of stage α is satisfied by A. As a consequence, requirement $R_{2,j}$ is satisfied when $2j+1$ is canceled. This completes the proof of the theorem. □

5. Open Questions

In this paper, we have constructed oracles which collapses the polynomial time hierarchy to exactly the kth level. Furthermore, relative to different oracles, the class $PSPACE$ may either collapse to the kth level of the polynomial-time hierarchy or may be different from the polynomial time hierarchy. Several questions about the relativized polynomial time hierarchy however remain open. First, note that the set $L_{\text{odd}}(A)$ is actually in the class $D\#P(A)$ (see, for definition, Angluin [1]). Thus, our results together with Yao's [12] result actually showed that relative to some oracles, the class $D\#P$ may be separated from the polynomial time hierarchy while the hierarchy may have either finite or infinite levels. An interesting question here is to find an oracle to separate the class $PSPACE$ from $D\#P$.

Heller [7] has constructed oracles X and Y such that $\Sigma_2^P(X) = \Pi_2^P(X) \neq \Delta_2^P(X)$ and $\Sigma_2^P(Y) = \Delta_2^P(Y) \neq \Sigma_1^P(Y)$. It would be interesting to see whether these results could be extended to the kth level of the polynomial time hierarchy.

References

1. D. Angluin, On counting problems and the polynomial time hierarchy, *Theoret. Comput. Sci.* **12** (1980), 161–173.

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. M. Furst, J. Saxe and M. Sipser, Parity, circuits, and the polynomial time hierarchy, *Math. Systems Theory* **17** (1984), 13–27.

5. J. T. Hastad, *Computational Limitations for Small-Depth Circuits*, (Ph.D. Dissertation, MIT), MIT Press, Cambridge, 1987.

6. J. T. Hastad, Almost optimal lower bounds for small depth circuits, *Proc. of 18th ACM Symp. on Theory of Computing* (1986), 6–20.

7. H. Heller, Relativized polynomial hierarchies extending two levels, *Math. systems Theory* **17** (1984), 71–84.

8. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

9. M. Sipser, Borel sets and circuit complexity, *Proc. of 15th ACM Symp. on Theory of Computing* (1983), 61–69.

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

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

12. A. Yao, Separating the polynomial-time hierarchy by oracles, *Proc. of 26th IEEE Symp. on Foundations of Computer Science* (1985), 1–10.