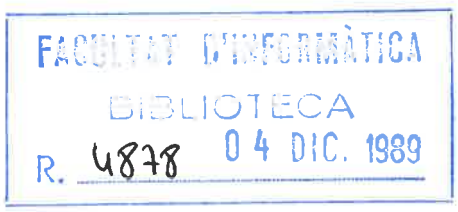
A combinatorial technique for separating counting complexity classes

J. Torán

Report LSI–88–7



Resum: Presentem una nova tècnica combinatòria per a obtenir separacions relativitzades d'algunes classes de complexitat relacionades amb la idea del comptatge, com PP, $\mathbf{G}$ (llindar exacte), i $\oplus$P (paritat). Per a demostrar la seva utilitat incloem tres relativitzacions que separen NP de $\mathbf{G}$, NP de $\oplus$P i $\oplus$P de PP. Altres separacions es dedueixen directament d'aquests resultats, i en conseqüència obtenim un oracle que separa PP de PSPACE, i resol per tant un problema obert per Angluin en [An,80]. A partir de les separacions relativitzadas obtenim separacions absolutes de classes de complexitat associades al comptatge amb temps de còmput afitat de forma logarítmica.

Resumen: Presentamos una nueva técnica combinatória para obtener separaciones relativizadas de algunas clases de complejidad relacionadas con la idea del conteo, como PP, $\mathbf{G}$ (umbral exacto), y $\oplus$P (paridad). Para demostrar su utilidad incluimos tres relativizaciones separando NP de $\mathbf{G}$, NP de $\oplus$P y $\oplus$P de PP. Otras separaciones se deducen directamente de estos resultados, y como consecuencia obtenemos un oráculo separando PP de PSPACE, resolviendo por tanto un problema abierto por Angluin en [An,80]. A partir de las separaciones relativizadas obtenemos separaciones absolutas de clases de complejidad asociadas al conteo con tiempo de cómputo acotado de forma logarítmica.

A combinatorial technique for separating counting complexity classes

Jacobo Torán
Facultat d'Informàtica de Barcelona
Pau Gargallo 5
08028 Barcelona
Spain

Abstract

We introduce a new combinatorial technique to obtain relativized separations of certain complexity classes related to the idea of counting, like PP, $\mathbf{G}$ (exact counting), and $\oplus$P (parity). To demonstrate its usefulness we present three relativizations separating NP from $\mathbf{G}$, NP from $\oplus$P and $\oplus$P from PP. Other separations follow from these results, and as a consequence we obtain an oracle separating PP from PSPACE, thus solving an open problem proposed by Angluin in [An,80]. From the relativized separations to obtain absolute separations for counting complexity classes with log-time bounded computation time.

1. Introduction.

In several previous works, different complexity classes have been defined trying to capture the complexity of computational problems in which counting is involved. This is the case of the class of threshold languages mNP, defined by Simon in [Sim,75]. A language L is in this class if there is a polynomial time Turing machine M such that for every input x, M has at least k accepting computation paths if and only if x is in L, where k is a fixed constant or fraction. This class is placed between NP and PSPACE, contains natural complete problems and it is closely related to Valiant's class #P of functions that count the number of accepting paths of a nondeterministic Turing machine [Va,79]. Simon also shows that the class of threshold languages is the same as the probabilistic class PP, of languages accepted by polynomial time probabilistic Turing machines of unbounded error, [Gi,77]. The languages in this class are those recognized by polynomial time bounded Turing machines which accept an input if and only if more than half of the computation paths accept.

More recently, Wagner introduced in [Wa,86] the counting hierarchy (CH) in a similar way as the polynomial time hierarchy (PH), as a tool to classify several combinatorial problems. Wagner defines the counting quantifier $\mathbf{C}$ inspired in the idea of threshold

machines. As we will see more formally, $\mathbf{C}_{f(x)}^{p(n)}\ y\ P(y)$ means that there are *at least* $f(x)$ strings of length $p(n)$ satisfying predicate P. Similar quantifiers have been defined in [Pa,Sc,86] and [Im,La,88]. If the quantifier $\mathbf{C}$ is placed in front of a polynomial time predicate, it defines a complexity class in the natural way, which we will call also $\mathbf{C}$. It is not hard to see that the class $\mathbf{C}$ coincides with mNP and PP. The counting hierarchy (CH), arises combining quantifier $\mathbf{C}$ with the exitential and universal quantifiers. PH is therefore contained in CH, and every level of the counting hierarchy possess complete problems [Wa,86]. CH can also be characterized in terms of nondeterministic and probabilistic oracle Turing machines [To,88b].

If the counting quantifier $\mathbf{C}$ is defined in a slightly different way, meaning that instead of having *at least* k strings satisfying certain predicate there are *exactly* k strings satisfying it, we obtain the exact counting quantifier $\mathbf{G}$ (also defined in [Wa,86] [Pa,Sc,86], [Im,La,88]). Although the definitions of $\mathbf{C}$ and $\mathbf{G}$ are so similar, as we will see, both quantifiers define complexity classes with very different properties. Concepts related to the counting quantifiers can be seen in the area of threshold circuits [Pa,Sc,86] [Re,87], [Ha,e.a,87].

A more moderate counting class is parity, $\oplus$P, defined in [Pa,Za,83]. A language L belongs to $\oplus$P if there is a nondeterministic Turing machine with an even number of accepting computation paths for inputs is L, and an odd number for inputs that are not in the language. In [Go,Pa,86] it is shown that $\oplus$P arises considering extended Turing machines over an exclusive-or basis; parity functions have been used in [Fu,Sa,Si,84], [Ya,85] and [Hå,86] to obtain lower bounds for bounded depth circuits.

From the relations among the mentioned complexity classes, it is only known that NP and $\mathbf{G}$ are contained in PP. An interesting overview of the developement of counting complexity classes in recent years can be found in [Sc,88].

In this paper we will consider the problem of separating counting complexity classes. Since the first relativized separation of the classes P, and NP [Ba,Gi,So,75], many relativized results have been obtained, indicating the hardness of the separations in the absolute case. The method used in the separation of P and NP, is strengthen in [Ba,Se,79] to separate the second from the third level of PH, and in [An,80] to separate PP from the second level of PH. A completely new method, based in the obtention of lower bounds for bounded deph circuits is considered in [Ya,85] [Hå,86] to separate the whole polynomial time hierarchy.

In this article we introduce a new combinatiorial method to obtain relativized separations of the counting classes mentioned above. Although counting classes have been separated from the polynomial time hierarchy before, [An,80], [Ya,85], [Hå,86], to our knowledge this is the first time that counting classes have been separated from other counting classes. The technique used to obtain our results is new since the methods from previous relativizations do not seem to work for counting complexity classes. The idea is to diagonalize gathering the number of accepting computation paths of the oracle Turing machines in combinatorial formulas in which the oracle is a variable, and then argue over

the formulas using combinatorial techniques and the fact that our machines are polynomial time bounded.

We present three relativizations separating NP from $\mathbf{G}$, NP from $\oplus$P and $\oplus$P from PP. As a consequence we obtain relativizations in which the three classes NP, $\oplus$P, and $\mathbf{G}$ are incomparable (answering a question proposed in [Ca,He,87]) $\oplus$P and PP are incomparable, and NP and $\mathbf{G}$ are strictly contained in PP. These separations also imply a relativization in which PP is different from PSPACE, solving an open problem proposed by Angluin in [An,80], as well as relativized separations of the lower levels of the counting hierarchy. Another consequence of the relativizations presented is the absolute separation of log-time complexity classes. Complexity classes of this kind have been considered before in [Ch,Ko,St,81] and [Sip,83].

The paper is organized as follows: after a section in which the basic definitions are given, we present in section 3 the main lemma in which our constructions are based. We try to motivate it with the example of the separation of NP from $\mathbf{G}$. We apply this lemma in section 4 to obtain the remaining separations, and in this section we also discuss the absolute separation of log-time complexity classes. We finish with a section of conclusions and open problems.

2. Basic definitions.

Most of the notions considered here are well known or can be found in the books on the subject (see for example [Ba,Di,Ga,88]). We will limit ourselves to define the counting classes that we will use.

Definition 1: The polynomial counting quantifier $\mathbf{C}$, is defined as follows: for a polynomial time computable function $f : \Sigma^\star \longrightarrow \mathbb{N}$, a polynomial p and a two argument predicate P,

$$\mathbf{C}^p_{f(x)}y \ : \ P(x,y) \Longleftrightarrow ||\{y : |y| \leq p(|x|) \text{ and } P(x,y)\}|| \geq f(x)$$

If K is a language class, for any set A, $A \in \mathbf{C}K$ if there is a function f in FP, s.t. for every x, $f(x) > 0$, a polynomial p and a language $B \in K$ such that for any $x \in \Sigma^\star$

$$x \in A \Longleftrightarrow \mathbf{C}^p_{f(x)}y \ : \ \langle x,y\rangle \in B$$

For simplicity we will denote the class $\mathbf{CP}$ by $\mathbf{C}$.

In a similar way we define the exact counting quantifier

Definition 2: For a function $f : \Sigma^\star \longrightarrow \mathbb{N}$, $f \in$FP, a polynomial p and a two argument predicate P,

$$\mathbf{G}^p_{f(x)}y \ : \ P(x,y) \Longleftrightarrow ||\{y : |y| \leq p(|x|) \text{and } P(x,y)\}|| = f(x)$$

As in the above case $\mathbf{G}$ can define language classes in the natural way. Since we will always deal with quantifiers ranging over strings of polynomial length, we drop the superscript p from all the quantifiers. Alternating quantifier $\mathbf{C}$ with the existential and universal quantifiers, we obtain the counting hierarchy.

Definition 3: The polynomial counting hierarchy (CH) is the smallest family of language classes satisfying:

i/ P$\in$CH.

ii/ If $K \in$CH then $\exists K$, $\forall K$, and $\mathbf{C}K \in$CH.

Finally the class parity is defined as

Definition 4: $\oplus$P= $\{L \subseteq \Sigma^\star$: there is a nondeterministic polynomial time machine recognizing L with an even number of accepting computation paths for input strings in L and an odd number of accepting computation paths for input strings in $\overline{L}\}$.

3. An oracle separating NP from G.

We start separating the classes NP and $\mathbf{G}$, trying to motivate the technique used. Let $M_1, M_2 \ldots$ be an enumeration of all the probabilistic Turing machines, and $p_1, p_2 \ldots$ an enumeration of the polynomials. W.l.o.g. we can suppose that for every k, M_k has computation time bounded by p_k.

Theorem 5: There is an oracle A such that $\mathrm{NP}^A \not\subset \mathbf{G}^A$.

Proof: For every set A, define $L_A = \{0^n : \exists w(|w| = n \text{ and } w \in A)\}$ clearly, for every set A, $L_A \in \mathrm{NP}^A$. We construct in stages a set A such that $L_A \notin \mathbf{G}^A$.

Stage 0. $A_0 := \emptyset; n_0 := 0$.

Stage s. Let n_s be the smallest integer such that

$n_s > n_{s-1}$

$n_s > \max\{p_i(n_{s-1}) : i < s\}$

$2^{n_s} > p_s(n_s)$

$(\star)$ $A_s := A_{s-1} \cup B$, being $B \subseteq \Sigma^{n_s}$, such that $B \neq \emptyset \iff 0^{n_s} \notin L(M_s, A_{s-1} \cup B)$;

(Here, for a string x and an oracle B, $x \in L(M_s, B)$ means that machine M_s has exactly th accepting computation paths for this input, being th the threshold of the machine for input x.)

Let $A = \bigcup_s A_s$. It is clear, following the same ideas as in [Ba,Gi,So,75], that if we prove the existence of set B in $(\star)$, then the set L_A is not in $\mathbf{G}^A$. In the following, we show that the set B in $(\star)$ always exists.

Notation: $Q^B_{a_1,\ldots,a_k,(b_1,\ldots,b_l)}$ denotes the number of accepting paths from M_s with oracle $A_{s-1} \cup B$ and input 0^{n_s}, in which all the words $a_1 \ldots a_k$ are queried, and none of the words $b_1 \ldots b_l$ is queried to the oracle. For example, $Q^{w_1,w_3}_{w_1,w_2,(w_3,w_4)}$ denotes the number of accepting paths from M_s with oracle $A_{s-1} \cup \{w_1, w_3\}$ and input 0^{n_s}, in which the words w_1 and w_2 are queried, and none of the words w_3, w_4 are queried to the oracle. Notice that we have omited the "{'s" from the set notation, from the superscript of the Q. Also observe that if a word is not queried, it doesn't make any difference if we drop it into, or take it away from the oracle, for example, the above expression $Q^{w_1,w_3}_{w_1,w_2,(w_3,w_4)}$ is equivalent to $Q^{w_1}_{w_1,w_2,(w_3,w_4)}$

The following lemma is needed for proving the result. We omit the proof since it is straightforward. It just says that we can decompose the set of accepting computations quering $w_1 \ldots w_k$ into two: those that query also w_{k+1} and those that do not.

Lemma 6: For any set B and any $k+1$ words $w_1, \ldots w_{k+1} \in \Sigma^{n_s}$ the following equality holds:

$$Q^B_{w_1,\ldots w_k} = Q^B_{w_1,\ldots w_k,w_{k+1}} + Q^B_{w_1,\ldots w_k,(w_{k+1})}$$

(Sometimes we will use the above equality in the form $Q^B_{w_1,\ldots w_k,w_{k+1}} = Q^B_{w_1,\ldots w_k,(w_{k+1})} - Q^B_{w_1,\ldots w_k}$ which should not create any confussion).

In order to operate in a concise form, the defined Q-expresions representing the number of accepting paths with different oracles, will be grouped into a combinatorial formula. The motivation for this is presented with more detail in [To88a].

Notation: For any $B, D \subseteq \Sigma^{n_s}$, with $B \cap D = \emptyset$,

$$J^B_D = \sum_{i=0}^{\|D\|} (-1)^i \sum_{\substack{A \subseteq D \\ \|A\|=i}} Q^{B \cup A}_D$$

We introduce now the main lemma, that will enable us to prove the existence of the oracles separating the classes.

Lemma 7: For any sequence of words $w_1, \ldots, w_k, w_{k+1}$ in Σ^{n_s}, and any set $B \subseteq \Sigma^{n_s}$, with $B \neq \emptyset$ and $B \cap \{w_1, \ldots, w_k, w_{k+1}\} = \emptyset$

$$J^{B \cup \{w_{k+1}\}}_{w_1 \ldots w_k} = J^B_{w_1 \ldots w_k} - J^B_{w_1 \ldots w_k, w_{k+1}}$$

Proof: For the proof, first we decompose the J's into Q's following the definition, and then manipulate the Q's either decomposing them into two by lemma 6, or deleting from the oracle some words that are not queried.

Let $D = \{w_1 \dots w_k\}$.

$$
\begin{aligned}
J_D^{B\cup\{w_{k+1}\}} &= \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} Q_{w_1\dots w_k}^{B\cup\{w_{k+1}\}\cup A} \\
&= \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} (Q_{w_1\dots w_k,w_{k+1}}^{B\cup\{w_{k+1}\}\cup A} + Q_{w_1\dots w_k(w_{k+1})}^{B\cup\{w_{k+1}\}\cup A}) \\
&= \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} (Q_{w_1\dots w_k,w_{k+1}}^{B\cup\{w_{k+1}\}\cup A} + Q_{w_1\dots w_k(w_{k+1})}^{B\cup A}) \\
&= \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} (Q_{w_1\dots w_{k+1}}^{B\cup\{w_{k+1}\}\cup A} + Q_{w_1\dots w_k}^{B\cup A} - Q_{w_1\dots w_{k+1}}^{B\cup A}) \\
&= \sum_{i=0}^{k}((-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} Q_{w_1\dots w_k}^{B\cup A}) + \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} (Q_{w_1\dots w_{k+1}}^{B\cup\{w_{k+1}\}\cup A} - Q_{w_1\dots w_{k+1}}^{B\cup A}) \\
&= J_D^B + \sum_{i=0}^{k}(-1)^i(\sum_{\substack{A\subseteq D\cup\{w_{k+1}\}\\ \|A\|=i+1}} Q_{w_1\dots w_{k+1}}^{B\cup A} - \sum_{\substack{A\subseteq D\\ \|A\|=i+1}} Q_{w_1\dots w_{k+1}}^{B\cup A} - \sum_{\substack{A\subseteq D\\ \|A\|=i}} Q_{w_1\dots w_{k+1}}^{B\cup A}) \\
&= J_D^B + \sum_{i=0}^{k}((-1)^i(\sum_{\substack{A\subseteq D\cup\{w_{k+1}\}\\ \|A\|=i+1}} Q_{w_1\dots w_{k+1}}^{B\cup A}) - Q_{w_1\dots w_{k+1}}^{B} \qquad (3) \\
&= J_D^B - \sum_{i=1}^{k+1}((-1)^i(\sum_{\substack{A\subseteq D\cup\{w_{k+1}\}\\ \|A\|=i}} Q_{w_1\dots w_{k+1}}^{B\cup A}) - Q_{w_1\dots w_{k+1}}^{B} \\
&= J_D^B - J_{D\cup\{w_{k+1}\}}^B
\end{aligned}
$$

Maybe the step taken to obtain the expression in (3) needs some clarification. Observe that in the expression before (3), the two last sums only differ in the size of $||A||$. Since these sums are part of another sum and are multiplied by $(-1)^i$, the terms cancel, the sum "telescopes", remaining only $Q_{w_1\dots w_{k+1}}^B$ and

$$(-1)^k \sum_{\substack{A\subseteq D\\ \|A\|=k+1}} Q_{w_1\dots w_{k+1}}^{B\cup A}$$

but this last term is 0 since $||D|| = k$. □

Lemma 8: For $s \geq 1$, if for every set $R \subseteq \Sigma^{n_s}$, it is true that

$$0^{n_s} \in L(M_s, A_{s-1} \cup R) \Longleftrightarrow R \neq \emptyset$$

then for any nonempty sequence of words $w_1 \ldots w_k$ in Σ^{n_s}, and any oracle B, with $B \neq \emptyset$, $B \neq \Sigma^{n_s}$ and $\{w_1 \ldots w_k\} \cap B = \emptyset$, it holds that $J^B_{w_1 \ldots w_k} = 0$. Moreover, $J^{\emptyset}_{w_1 \ldots w_k} = J^{\emptyset}_{w_1} = Q^{\emptyset} - th$

Proof: By hypothesis, and using the definition of J, for every set $B \neq \emptyset$, $J^B = Q^B = th$.

By lemma 7

$$J^B_{w_1 \ldots w_{k+1}} = J^B_{w_1 \ldots w_k} - J^{B \cup \{w_{k+1}\}}_{w_1 \ldots w_k}$$

We prove the first claim by induction on k.

For $k = 1$, $J^B_{w_1} = J^B - J^{B \cup \{w_1\}} = th - th = 0$

For $k > 1$, $J^B_{w_1 \ldots w_k} = J^B_{w_1 \ldots w_{k-1}} - J^{B \cup \{w_k\}}_{w_1 \ldots w_{k-1}}$ where by induction hypothesis both terms are 0.

The second claim is proved also by induction on k.

For k=1,

$$J^{\emptyset}_{w_1} = Q^{\emptyset}_{w_1} - Q^{w_1}_{w_1} = Q^{\emptyset} - Q^{\emptyset}_{(w_1)} - Q^{w_1} + Q^{w_1}_{(w_1)} = Q^{\emptyset} - Q^{w_1} = Q^{\emptyset} - th$$

For $k > 1$, $J^{\emptyset}_{w_1 \ldots w_k} = J^{\emptyset}_{w_1 \ldots w_{k-1}} - J^{w_k}_{w_1 \ldots w_{k-1}}$, but by the first part of the result, $J^{w_k}_{w_1 \ldots w_{k-1}} = 0$. By induction hypothesis $J^{\emptyset}_{w_1 \ldots w_{k-1}} = J^{\emptyset}_{w_1}$. Thus $J^{\emptyset}_{w_1 \ldots w_k} = J^{\emptyset}_{w_1}$. □

Now we are ready to prove the existence of the set B in $(\star)$.

Lemma 9: For every $s \geq 1$ if $0^{n_s} \notin L(M_s, A_{s-1})$ then there is a set $B \subseteq \Sigma^{n_s}, B \neq \emptyset$, such that $0^{n_s} \notin L(M_s, A_{s-1} \cup B)$.

Proof: Let th be the threshold of the machine for input 0^{n_s}. Suppose that the mentioned set B does not exist, then for every set $B \subseteq \Sigma^{n_s}, B \neq \emptyset, Q^B = th$. We are in the hypothesis of lemma 8.

Let $p = p_{n_s}(n_s)$. Since the running time of M_s on input 0^{n_s} is bounded by p, the machine can make at the most p queries to the oracle on every computation path, and therefore $J^{\emptyset}_{w_1 \ldots w_{p+1}} = 0$. (Recall that $J^{\emptyset}_{w_1 \ldots w_{p+1}}$ is a sum of computation paths in which all words $w_1 \ldots w_{p+1}$ are queried).

On the other hand, by lemma 8, $J^{\emptyset}_{w_1 \ldots w_{p+1}} = J^{\emptyset}_{w_1} = Q^{\emptyset} - th$. It follows that $Q^{\emptyset} = th$, which contradicts the hypothesis since $0^{n_s} \notin L(M_s, A_{s-1})$. □

4. Separations from parity.

The special structure of the class $\oplus$P, will be used in this section to obtain new separations. We start separating NP from $\oplus$P.

Theorem 10: There is an oracle A such that $\mathrm{NP}^A \not\subset \oplus\mathrm{P}^A$.

For every set A, define $L_A = \{0^n : \exists w|w| = n \text{ and } w \in A\}$. Clearly, for every set $A, L_A \in \mathrm{NP}^A$. We construct in stages a set A such that $L_A \notin \oplus\mathrm{P}^A$. Given an enumeration of onondeterministic Turing machines as in theorem 5, we construct oracle A in the same way as before. In every stage s

$(\star\star)$ $A_s := A_{s-1} \cup B$, being $B \subseteq \Sigma^{n_s}$, such that $B \neq \emptyset \iff 0^{n_s} \notin L(M_s, A_{s-1} \cup B)$;

And define $A = \bigcup_s A_s$. Observe that since we are trying to diagonalize away from parity, the machines have now a parity accepting mechanism and the expression $0^{n_s} \notin L(M_s, A_{s-1} \cup B)$ means that machine M_s on input 0^{n_s}, and oracle $A_{s-1} \cup B$ has an odd number of computation paths. It should be clear that if we manage to prove the existence of set B in $(\star\star)$, then the set L_A cannot be in $\oplus\mathrm{P}^A$. In the following, we show that the set B in $(\star\star)$ always exists, we will make use of lemma 7 in the preceding section.

Lemma 11: For $s \geq 1$, if for every set $R \subseteq \Sigma^{n_s}$, it is true that

$$0^{n_s} \in L(M_s, A_{s-1} \cup R) \iff R \neq \emptyset$$

then for any nonempty sequence of words $w_1 \ldots w_k$ in Σ^{n_s}, and any oracle B, with $B \neq \emptyset$, $B \neq \Sigma^{n_s}$ and $\{w_1 \ldots w_k\} \cap B = \emptyset$, it holds that $J^B_{w_1 \ldots w_k}$ is even. Moreover, $J^{\emptyset}_{w_1 \ldots w_k}$ is odd.

Proof: By the definition of acceptance of M_s and the hypothesis, for every set B, $B \neq \emptyset$, J^B is even, and by lemma 7

$$J^B_{w_1 \ldots w_{k+1}} = J^B_{w_1 \ldots w_k} - J^{B \cup \{w_{k+1}\}}_{w_1 \ldots w_k}$$

We prove the first claim by induction on k.

For $k = 1$, $J^B_{w_1} = J^B - J^{B \cup \{w_1\}}$, since J^B and $J^{B \cup \{w_1\}}$ are even, so is $J^B_{w_1}$.

For $k > 1$, $J^B_{w_1 \ldots w_k} = J^B_{w_1 \ldots w_{k-1}} - J^{B \cup \{w_k\}}_{w_1 \ldots w_{k-1}}$, and by induction hypothesis, both members of the right hand side of the equation are even.

The second claim is proved also by induction on k.

For $k = 1$, $J^{\emptyset}_{w_1} = J^{\emptyset} - J^{w_1}$. $J^{\emptyset}$ is odd by hypothesis and J^{w_1} is even by the first part of the result.

For $k > 1$, $J^{\emptyset}_{w_1 \ldots w_k} = J^{\emptyset}_{w_1 \ldots w_{k-1}} - J^{w_k}_{w_1 \ldots w_{k-1}}$, but by the first part of the result, $J^{w_k}_{w_1 \ldots w_{k-1}}$ is even. By induction hypothesis $J^{\emptyset}_{w_1 \ldots w_{k-1}}$ is odd, and it follows that $J^{\emptyset}_{w_1 \ldots w_k}$ is also odd. □

Now we are ready to prove the existence of set B in $(\star\star)$.

Lemma 12: For every $s \geq 1$ there is a set $B \subseteq \Sigma^{n_s}$, such that $B \neq \emptyset \iff 0^{n_s} \notin L(M_s, A_{s-1} \cup B)$.

Proof: Suppose that the mentioned set B does not exist, then we are in the hypothesis of lemma 11.

Let $p = p_{n_s}(n_s)$. Since the running time of M_s on input 0^{n_s} is bounded by p, the machine can make at most p queries to the oracle on each computation path, and therefore $J^{\emptyset}_{w_1 \ldots w_{p+1}} = 0$.

On the other hand, by lemma 11, $J^{\emptyset}_{w_1 \ldots w_{p+1}}$ is odd. This is a contradiction and it follows that the mentioned set B always exists. □

We present now the last separation, this time separating $\oplus$P from PP. This result will bring as a consequence the separation of different classes in the counting hierarchy.

Theorem 13: There is an oracle A such that $\oplus \mathrm{P}^A \not\subset \mathrm{PP}^A$.

For every set A, define $L_A = \{0^n : ||A \cap \Sigma^n|| \text{ is even}\}$. Clearly, for every set A, $L_A \in \oplus \mathrm{P}^A$. We construct in stages a set A such that $L_A \notin \mathbf{C}^A$. Consider again an enumeration of Turing machines and polynomials as in theorem 5.

Stage 0. $A_0 := \emptyset; n_0 := 0$.

Stage s. Let n_s be the smallest integer such that

$n_s > n_{s-1}$

$n_s > \max\{p_i(n_{s-1}) : i < s\}$

$2^{n_s} > p_s(n_s)$

$(\star\star\star)$ $A_s := A_{s-1} \cup B$, being $B \subseteq \Sigma^{n_s}$, such that $0^{n_s} \in L(M_s, A_{s-1} \cup B) \iff ||B||$ is odd;

Let $A = \bigcup_s A_s$. In the following, we show that the set B in $(\star\star\star)$ always exists, which implies that $L_A \notin \mathrm{PP}^A$.

Lemma 14: For $s \geq 1$, if for every set $R \subseteq \Sigma^{n_s}$, it is true that

$$0^{n_s} \in L(M_s, A_{s-1} \cup R) \iff ||R|| \text{ is even}$$

then for any nonempty sequence of words $w_1 \ldots w_k$ in Σ^{n_s}, and any oracle B, with $B \neq \Sigma^{n_s}$ and $\{w_1 \ldots w_k\} \cap B = \emptyset$, it holds that $J^B_{w_1 \ldots w_k} \neq 0$. More precisely, if $||B||$ is even then $J^B_{w_1 \ldots w_k} > 0$ and if $||B||$ is odd then $J^B_{w_1 \ldots w_k} < 0$.

Proof: By the definition of acceptance of M_s and the hypothesis, for every set B, $||B||$ is even $\iff J^B \geq th$, and by lemma 7

$$J^B_{w_1 \ldots w_{k+1}} = J^B_{w_1 \ldots w_k} - J^{B \cup \{w_{k+1}\}}_{w_1 \ldots w_k}$$

By induction on k. (Suppose $||B||$ is even, the odd case is analogous.)

For $k = 1$, $J^B_{w_1} = J^B - J^{B \cup \{w_1\}}$, since $J^B \geq th$ and $J^{B \cup \{w_1\}} < th$, we obtain $J^B_{w_1} > 0$.

For $k > 1$, $J^B_{w_1 \ldots w_k} = J^B_{w_1 \ldots w_{k-1}} - J^{B \cup \{w_k\}}_{w_1 \ldots w_{k-1}}$, and by induction hypothesis $J^B_{w_1 \ldots w_{k-1}} > 0$ and $J^{B \cup \{w_k\}}_{w_1 \ldots w_{k-1}} < 0$. It follows that $J^B_{w_1 \ldots w_k} > 0$. □

Now we are ready to prove the existence of set B in $(\star\star\star)$.

Lemma 15: For every $s > 1$ there is a set $B \subseteq \Sigma^{n_s}$, such that

$$0^{n_s} \in L(M_s, A_{s-1} \cup B) \Longleftrightarrow ||B|| \text{ is odd}$$

Proof: Let th be the threshold of the machine for input 0^{n_s}. Suppose that the mentioned set B does not exist, then we are in the hypothesis of lemma 14.

Let $p = p_{n_s}(n_s)$. Since the running time of M_s on input 0^{n_s} is bounded by p, the machine can make at most p queries to the oracle on each computation path, and therefore $J^{\emptyset}_{w_1 \ldots w_{p+1}} = 0$.

On the other hand, by lemma 14, $J^{\emptyset}_{w_1 \ldots w_{p+1}} > 0$. This is a contradiction and it follows that the mentioned set B always exists. □

We obtain many consequences of the above separations:

Corollary 16: There is a relativization under which the three classes NP, **G** and $\oplus$P are incomparable, $\oplus$P and PP are incomparable and NP and **G** are strictly contained in PP.

Proof: Follows from the above separations, the fact that co-NP is contained in **G** and the boolean properties of the classes considered (see [To,88a]).

We can also separate the low levels of the counting hierarchy.

Corollary 17: There is a relativization under which $\mathbf{C} \neq \exists\mathbf{C}$, $\mathbf{C} \neq \forall\mathbf{C}$.

Proof: Follows from theorem 13 and the fact that $\oplus\mathrm{P} \subseteq \mathrm{P}^{\mathrm{PP}} \subseteq \exists\mathbf{C} \cap \forall\mathbf{C}$. [To,88b].

From this result it is clear that there is an oracle such that $\mathrm{PP}^A \neq \mathrm{PSPACE}^A$ (open since [An,80]).

A final consequence is the absolute separation of classes in the logarithmic time counting hierarchy (LCH). The classes in this hierarchy are defined using logarithmic length bounded quantifiers and machines working in logarithmic time with indirect access to the input. Similar machines have been defined in [Ch,Ko,St,81] and [Si,83]. For a formal definition we refer to [To,88a]. We separate the classes $\mathbf{C}^l$, $\mathbf{C}^l\exists^l$ and $\mathbf{C}^l\forall^l$. These classes are placed under LOGSPACE and contain well known complete problems with respect to projection reducibilities (for example MAJORITY is complete for $\mathbf{C}^l$, [To,88a]). The absolute separation of these classes is therefore important since it shows that certain known problems cannot be computed using only a logarithmic bounded counting quantifier.

For the absolute separation of the classes, we need the following theorem which is a generalization of an observation in [Si,83].

Theorem 18: Let K_1 and K_2 be any two classes in CH. If there is a relativization under which $K_1 \neq K_2$, then the corresponding classes in LCH, K_1' and K_2' are different (in the absolute sense).

Corollary 19: $\exists^l \mathbf{C}^l \not\subset \mathbf{C}^l$ and $\forall^l \mathbf{C}^l \not\subset \mathbf{C}^l$.

5. Conclusions and open problems.

We have introduced a combinatorial technique which allows us to obtain relativized separations of counting complexity classes, bringing new ideas in the theory of relativizations and solving certain problems that were open, like the separation from PP and PSPACE. The obtained relativizations separate the first levels of the counting hierarchy, and imply absolute separations for some log-time complexity classes. The obvious question to ask is if our technique can be used to separate higher levels of the counting hierarchy. Allthough probably with certain modifications of the method the second and third levels of the hierarchy can be separated, we believe that new ideas are needed to separate the whole counting hierarchy. We would also like to point out that in [Be,88], Beigel has recently developed a different technique to obtain the separations of this article.

References

[An,80] D. Angluin: On counting problems and the polynomial-time hierarchy. *Theoret. Comput. Sci.* 12 (1980), 161–173.

[Ba,Gi,So,75] T.P. Baker, J. Gill, and R.M. Solovay: Relativizations of the P=?NP question. *SIAM J. Comput.* 4 (1975), 431–442.

[Ba,Se,79] T.P. Baker, A.L. Selman: A second step toward the polynomial hierarchy. *Theoret. Comput. Sci.* 8 (1979), 177–187.

[Ba,Di,Ga,88] J.L. Balcázar, J. Diaz, and J. Gabarró: *Structural Complexity* (vol. I). Springer-Verlag (1987).

[Be,88] R. Beigel: Relativized counting classes: Relations among thresholds, parity and mods. Manuscript (1988).

[Ca,He,87] J.-Y. Cai, L.A. Hemachandra: On the power of parity. Manuscript (1987).

[Ch,Ko,St,81] A.K. Chandra, D.C. Kozen, L.J. Stockmeyer: Alternation. *Journal ACM* 28 (1981), 114–133.

[Fu,Sa,Si,84] M. Furst, J.B. Saxe, M. Sipser: Parity, circuits, and the polynomial-time hierarchy. *Math. Syst. Th.* 17 (1984), 13–27.

[Gi,77] J. Gill: Computational complexity of probabilistic Turing machines. *SIAM J. Comput.* 6 (1977), 675–695.

[Go,Pa,86] L.Goldschlager, I.Parberry: On the construction of parallel computers from various bases of boolean functions. *Theoret. Comput. Sci.* 43 (1986) 43–58

[Ha,e.a,87] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, G. Turan: Threshold circuits of bounded depth. Proc. 28th FOCS, IEEE (1987) 99–110.

[Hå,86] J. Håstad: Computational limitations for small depth circuits. Ph.D. Thesis, M.I.T. (1986).

[Im,La 88] N. Immerman, E. Lander: Describing graphs: a first-order approach to graph canonization. Report DCS/TR-605 Yale U. (1988).

[Pa,Za,83] C.H. Papadimitriou, S. Zachos: Two remarks on the power of counting. 6th GI Conference on Theoret. Comput. Sci., Lect. Notes in Comp. Sci. 145 (1983), Springer-Verlag, 269–276.

[Pa,Sc,86] I.Parberry, G.Schnitger: Parallel computation with threshold functions. Proc. 1st Structure in Complexity Theory Conference, (1986) Springer-Verlag, 222–289.

[Re,87] J.Reif: On threshold circuits and polynomial computation. Proc. 2nd Structure in Complexity Theory Conference, IEEE (1987) 118-125.

[Sc,88] U.Schöning: The power of counting. Proc. 3rd Structure in Complexity Theory Conference, IEEE, (1988).

[Sim,75] J. Simon: On some central problems in computational complexity. Ph.D. Thesis, Cornell University (1975).

[Sip,83] M. Sipser: Borel sets and circuit complexity. Proc. 15th STOC (1983), 61–69.

[To,88a] J. Torán: Structural properties of the counting hierarchies. Ph.D. Thesis. Facultat d'Informática de Barcelona, (1988).

[To,88b] J. Torán: An oracle characterization of the counting hierarchy. Proc. 3rd Structure in Complexity Theory Conference, IEEE, (1988), 213–223

[Va,79] L.G. Valiant: The complexity of computing the permanent. *Theoret. Comput. Sci.* 8 (1979), 189–201.

[Wa,86] K. Wagner: The complexity of combinatorial problems with succint input representation. *Acta Informatica* 23 (1986), 325–356.

[Ya,85] A.Yao: Separating the polynomial time hierarchy with oracles, 26th FOCS, (1985), 1–10.