Counting the number of solutions

A survey of recent inclusion results in the area of counting classes

Jacobo Torán*
Departament L.S.I.
U. Politècnica de Catalunya
Pau Gargallo 5
08028 Barcelona, Spain

1. Introduction

Research in the area of counting complexity classes has been very fruitful during the last years. These complexity classes are defined in terms of nondeterministic machines for which the accepting mechanism is a predicate on the number of accepting paths or solutions of the machine. Examples of these classes are NP, PP (probabilistic polynomial time) [10] , US (unique solutions) [8], $\mathbf{G}$P (exact counting) [29], $\oplus$P (parity P) [16], and $\mathrm{MOD}_k\mathrm{P}$ (modulo classes) [6] (we include later definitions of the classes treated in this survey). All these classes have complete problems, and are included in PSPACE. Although the classes are interesting in themselves, an important further reason for studying them, as we will see, is the fact that they can help to understand properties of some better known classes like NP, PSPACE or the polynomial time hierarchy (PH).

In the last two years many absolute inclusion results and relativized separations of counting classes have been obtained, which have clarified the relations among these classes. Interesting overviews of the problems solved in the area can be seen in [2] and [21]. In this paper we will concentrate in the most recent inclusion results. We will look at them under a new point of view and show that many of these results can be unified (and in some cases simplified) under the concept of *lowness*, a notion first defined by Schöning for the classes in the polynomial time hierarchy [18]. Intuitively a set (or a class) is low for a complexity class $\mathcal{K}$, if it does not provide $\mathcal{K}$ with any additional power when used as oracle.

We study in section 2 classes of sets which are low for PP, and show that complexity classes defined by machines with a bounded number of solutions (*Few* classes) have this property [14]. We prove that the bounded error probability class BPP is also low for PP [14]. In section 3 we give a characterization of the class $\mathrm{P}^{\mathrm{NP[log]}}$ in terms of "fewness" which can be used to show the inclusion $\mathrm{P}^{\mathrm{NP[log]}} \subseteq \mathrm{PP}$ [7]. Finally we study in section 4 low classes for P^{PP} and obtain that $\oplus$P and NP fall into this category. A direct consequence of this fact is the surprising result by Toda which says that all classes in the PH are included in P^{PP} [23].

* Supported by ESPRIT-II Basic Research Actions Program of the EC under contract No. 3075 (project ALCOM).

Complexity classes

We define now some of the less known complexity classes used in this survey. For definitions of classes like P, NP, PH, PSPACE, FP or concepts like polynomial time reductions we refer the reader to the basic books in the area [3,19,31].

Definition 1: For a nondeterministic machine M and a string $x \in \Sigma^\star$, let $acc_M(x)$ be the number of accepting computation paths of M on input x. Analogously, for a nondeterministic oracle machine M, an oracle A, and a string $x \in \Sigma^\star$, $acc_M^A(x)$ is the number of accepting paths of M^A with input x.

Definition 2: [10] A language L is in the class PP (probabilistic polynomial time) if there is a nondeterministic polynomial time machine M and a function $f \in$ FP such that for every $x \in \Sigma^\star$,

$$x \in L \iff acc_M(x) \geq f(x).$$

Originally the acceptance mechanism of this class required the number solutions to be more than one half of the total number of paths. However, this definition is equivalent to the above one [22].

Definition 3: [16] A language L is in the class $\oplus$P (parity P) if there is a nondeterministic polynomial time machine M such that for every $x \in \Sigma^\star$,

$$x \in L \iff acc_M(x) \text{ is odd.}$$

Definition 4: [10] A language L is in the class BPP (bounded error probabilistic polynomial time) if there is a nondeterministic machine M with running time bounded by a polynomial p, and such that for every $x \in \Sigma^\star$, the ratio between accepting and rejecting paths is $\geq \frac{3}{4}$.

$$x \in L \implies acc_M(x) \geq 3 \cdot 2^{p(|x|)-2}$$
$$x \notin L \implies acc_M(x) \leq 2^{p(|x|)-2}$$

For each polynomial q, this ratio can be amplified to $1 - 2^{-q(|x|)}$ [20,32].

For a complexity class $\mathcal{K}$ we will denote by $\mathrm{P}^{\mathcal{K}}$ the class of sets polynomial time Turing reducible to a set in $\mathcal{K}$. Some straightforward relations among the defined classes are BPP $\subseteq$ PP, and $\oplus$P $\subseteq \mathrm{P}^{\mathrm{PP}}$. It is also known that $\mathrm{P}^{\oplus \mathrm{P}} = \oplus$P [16] and that $\mathrm{P}^{\mathrm{BPP}}$ =BPP [32], i.e., $\oplus$P and BPP are closed under polynomial time Turing reducibility. There are relativizations separating all three classes [4,17,26]. Based on the class PP, and in a similar way as it is done with NP in PH, it is possible to define a hierarchy of complexity classes (PP, $\mathrm{PP}^{\mathrm{PP}}$ ...), which is called the counting hierarchy [25,29].

Classes as operators

Based on the definition of certain language classes, it can be useful to consider operators acting over predicates, obtaining this way new characterizations of complexity classes. For example, based on the class NP, the operator $\exists^p$ has been defined.

Definition 5: Given a complexity class $\mathcal{K}$, $\exists^p\mathcal{K}$ is the class of languages L for which there is a set A in $\mathcal{K}$ and a polynomial q such that for every $x \in \Sigma^\star$

$$x \in L \iff \exists y,\ |y| \in \Sigma^{q(|x|)} \text{ and } \langle x, y\rangle \in A.$$

It is well known that NP=$\exists^p$P, and there is an alternative characterization of the classes in PH in terms of the $\exists^p$ an $\forall^p$ operators. The classes BPP and PP have also been used to define operators, [20,29,33], and in fact the original definition of the counting hierarchy was in term of the operator version of PP. In section 3 we will define another operator based on a counting complexity class.

Lowness

The idea of lowness as a way to measure the complexity of a set or class of sets, was first introduced in [18] and developed in [12]. Initially this concept was used for classes in PH and was translated to the counting hierarchy in [14,24,25]. A set is low for a complexity class, if it cannot "help" the class when used as oracle. More formally,

Definition 6: For a language L and a complexity class $\mathcal{K}$ (which has a senseful relativized version), we will say that L is low for $\mathcal{K}$ if $\mathcal{K}^L = \mathcal{K}$. For a language class C, C is low for $\mathcal{K}$ if for every language L in C, $\mathcal{K}^L = \mathcal{K}$.

As we will see, many results about counting complexity classes can be interpreted in terms of lowness.

2. A small number of solutions, lowness for PP

The definitions of complexity classes given in the previous section are based on the number of accepting paths of nondeterministic Turing machines. If such machines have polynomial running times, the number of accepting paths for a given input can range over a set which is exponential in the size of the input. The question of whether the intractability of the languages in these complexity classes could be caused by the large variation in the number of possible solutions, has provoked important research [28]. It is therefore natural to define complexity classes by bounding the number of accepting paths of nondeterministic Turing machines, and study whether the problems in these classes are feasible to compute.

The first complexity class defined following this idea was Valiant's class UP (unambiguous NP) [27] of languages accepted by nondeterministic Turing machines with at most one accepting path for every input. This class plays an important role in the areas of one-way functions and cryptography. UP was generalized in a natural way by allowing a polynomial number of accepting paths. This gives rise to the class FewP defined by Allender [1] in connection with the notion of P-printable sets.

Definition 7: [1] A language L is in the class FewP if there is a nondeterministic polynomial time machine M and a polynomial p such that for every $x \in \Sigma^\star$,

i) $acc_M(x) \leq p(|x|)$
ii) $x \in L \iff acc_M(x) > 0$

From the definition follows immediately UP $\subseteq$ FewP $\subseteq$ NP. Although the number of solutions in a machine computing a problem in FewP is bounded by a polynomial, the exact number is not fixed beforehand, and it can range over a polynomial space. In [9] the interesting inclusion FewP$\subseteq \oplus P$ was obtained. The proof of this result is similar to the one of the next theorem, which tells us that for languages in FewP we can construct nondeterministic Turing machines having either f or $f-1$ solutions for some function f in FP. This fact is the key to prove the lowness of FewP for some interesting counting classes.

Theorem 8: [14] For every $L \in$ FewP, there is a nondeterministic polynomial time Turing machine M' and a a function $f \in$ FP such that

$$x \in L \implies acc_{M'}(x) = f(x) - 1$$
$$x \notin L \implies acc_{M'}(x) = f(x).$$

Proof: Let M be a nondeterministic machine for L with polynomial time bound p. Let q be a polynomial bounding acc_M. W.l.o.g we can asume that every computation path of $M(x)$ has length $r(|x|)$ for some polynomial r. Since fro every x $acc_M(x)$ is bounded, it is possible to construct dor every integer $t \leq q(|x|)$ a nondeterministic machine having exactly $\binom{acc_M(x)}{t}$ accepting paths. Such a machine on input x just has to guess t different computation paths of $M(x)$ and accept iff all these paths accept. We want to use this fact to construct a nondeterministic machine having exactly

$$\sum_{k=1}^{q(|x|)} (-1)^k \binom{acc_M(x)}{k}$$

accepting paths on input x. By the binomial theorem, this number is equal to 0 if $acc_M(x) = 0$ and equal to -1 if $acc_M(x) > 0$. There is a problem since in the above sum there are negative quantities and it is not possible to have machines with a negative number of accepting paths. This can be avoided; if we need to construct a machine with $-t$ accepting paths, we can build one with $2^s - t$ accepting paths (for sufficient large s) just by constructing one with t solutions and switching accepting and rejecting paths. The sum can now be performed and at the end there will be a residue of 2^s. The process is described by the following machine M':

input x;
guess k, $0 \leq k \leq q(|x|)$;
guess $y_1 < \cdots < y_k \in \Sigma^{r(|x|)}$;
if for every i, $1 \leq i \leq k$ y_i is an accepting path for x,
    then $test :=$ **true**
    else $test :=$ **false**;
if $(test \wedge \mathbf{even}(k)) \vee (\neg test \wedge \mathbf{odd}(k))$
    then accept

else reject.

The number of accepting paths of $M'(x)$ is

$$\sum_{k>0,\mathrm{even}(k)}^{q(|x|)} \binom{acc_M(x)}{k} + \sum_{k>0,\mathrm{odd}(k)}^{q(|x|)} 2^{k\cdot r(x)} - \binom{acc_M(x)}{k} =$$

$$\sum_{k=1}^{q(|x|)} \binom{acc_M(x)}{k}(-1)^k + f(x)$$

being f a function in FP. Therefore

$$acc_{M'}(x) = \begin{cases} f(x) - 1 & \text{if } x \in L; \\ f(x) & \text{if } x \notin L \end{cases}$$

□

Corollary 9: [9] FewP is low for $\oplus$P.
Follows from the above theorem and the fact $\oplus \mathrm{P}^{\oplus \mathrm{P}} = \oplus \mathrm{P}$ [16].

We will show now that the class FewP is also low for the class PP. In order to do this we prove a theorem similar to the previous one stating that given a nondeterministic Turing machine with an oracle in FewP, we can construct a new machine without oracle, that has exactly the same number of accepting paths as the first one plus some additional number of paths, which can be computed in polynomial time.

Theorem 10: For every nondeterministic polynomial time machine M and every set $L \in$ FewP, there is a nondeterministic machine M' and a function $g \in$ FP such that

$$acc_{M'}(x) = acc_M^L(x) + g(x)$$

Proof: It is divided in two parts. We first show that the oracle set L can be substituted by another one that only needs to be queried once by machine M. We then use theoren 8 to obtain the result.

Claim: There is a set $R \in$ FewP and a polynomial p such that

$$acc_M^L(x) = \|\{y \in \Sigma^{p(|x|)} \mid \langle x, y\rangle \notin R\}\|$$

Proof of claim: Since $L \in$ FewP, there is a nondeterministic machine M_L such that $y \in L \iff acc_{M_L}(y) > 0$ Consider the following language R':

$$\begin{aligned} \langle x, w\rangle \in R' \quad :\Longleftrightarrow \quad & w = \langle z, \langle y_1, a_1, v_1\rangle, \ldots, \langle y_k, a_k, v_k\rangle\rangle \text{ and } M \text{ on input } x \\ & \text{following the computation path } z, \text{making the queries} \\ & y_1, \ldots, y_k \text{ in this order and continuing with answer "yes" for } y_j \\ & \text{if } a_j = 1 \text{ and with answer "no" if } a_j = 0, \text{ accepts,} \\ & \text{and for every } j = 1, \ldots, k : \text{either } a_j = 0,\ v_j = \lambda \text{ and } y_j \notin L \\ & \text{or } a_j = 1,\ y_j \in L \text{ and } v_j \text{ is the lex. first accepting path of } M_L(y_j) \end{aligned}$$

Observe that R' is the complement of a set in FewP. Also, since for every input x and every accepting path z of $M(x)$, there is exactly one string $w = \langle z, v\rangle$ such that $\langle x, w\rangle \in R'$, we obtain

$$acc_M^L(x) = \|\{w \in \Sigma^* \mid \langle x, w\rangle \in R'\}\|.$$

The claim easily follows by taking the complement of R' and padding the w's.

Using theorem 8, we can now construct a nondeterministic machine M' for the FewP language R fulfilling

$$acc_{M'}\langle x, w\rangle = \begin{cases} f(\langle x, w\rangle), & \langle x, w\rangle \in R \\ f(\langle x, y\rangle) + 1, & \langle x, w\rangle \notin R \end{cases}$$

being f a function in FP. Consider the nondeterministic polynomial time machine M'' which on input x, guesses a string $w \in \Sigma^{p(|x|)}$ and simulates machine M' on input $\langle x, w\rangle$, accepting iff M' accepts. The number of accepting paths of M'' on input x is

$$\underbrace{2^{p(|x|)} f(\langle x, w\rangle)}_{:= g(x)} + \underbrace{\|\{y \in \Sigma^{p(|x|)} \mid \langle x, w\rangle \notin R\}\|}_{acc_M^L(x)}$$

□

Corollary 11: [14] FewP is low for PP.

Proof: Let A be a language in PP^L, with $L \in$ FewP. By the definition of PP there is a nondeterministic machine M and a function $f \in$ FP such that for every input $x \in \Sigma^\star$,

$$x \in L \iff acc_M^L(x) \geq f(x).$$

By theorem 10 there is a machine M' and a function $g \in$ FP with $acc_{M'}(x) = acc_M^L(x) + g(x)$, and therefore

$$x \in L \iff acc_{M'}(x) \geq f(x) + g(x),$$

which implies $L \in$ PP. □

Cai and Hemachandra introduce in [9] the new complexity class Few defined also in terms of nondeterministic machines with a bounded number of accepting paths. This class is a generalization of FewP but with a more flexible acceptance mechanism. We will prove that this generalization is also low for PP. In order to avoid confusion with the class FewP and some other classes defined in the next section, we will denote the class Few by Fewpaths.

Definition 12: [9] A language L is in the class Fewpaths if there is a nondeterministic polynomial time machine M, a polynomial time predicate Q, and a polynomial p such that for every $x \in \Sigma^\star$,

i) $acc_M(x) \leq p(|x|)$

ii) $x \in L \iff Q(x, acc_M(x))$

Fewpaths does not seem to be a subclass of NP. It is obvious that FewP $\subseteq$ Fewpaths, and it was shown in [9] that this class is closed under bounded truth-table reductions.

Köbler shows in [13] that Fewpaths is included in $\mathrm{P}^{\mathrm{FewP}}$. To prove this result, one has to count the number of accepting paths of a machine with a bounded number of solutions; the naive idea of doing binary search in the set $\{\langle x, k\rangle \mid$ there are at least k accepting paths for $x\}$ cannot be used since although k cannot be too large, this set is in NP but not neccessarily in FewP. There is however a more subtle way to obtain the result using a prefix search technique and constructing all the possible accepting paths. From the fact Fewpaths $\subseteq \mathrm{P}^{\mathrm{FewP}}$, the lowness properties of Fewpaths follow directly from the ones of FewP, since sets which are Turing reducible to low sets for a certain class, are also low for that class.

Corollary 13: [9,14] Fewpaths is low for $\oplus$P and for PP.

In the next theorem the idea behind the lowness properties of the *Few* classes can be best observed. Languages computed by machines with a small number of solutions can be recognized by other machines in which the exact number of accepting paths is fixed to be an arbitrary function in FP of the original number of accepting paths. The theorem is quite strong since function g in the statement can be any function in FP.

Theorem 14: [14] For every machine M with a polynomially bounded number of accepting paths, and every function g in FP from $\Sigma^\star \times \mathbb{N}$ to $\mathbb{N}$, there is a nondeterministic polynomial time machine M' and a polynomial r such that for every $x \in \Sigma^\star$,

$$acc_{M'}(x) = g(x, acc_M(x)) + 2^{r(|x|)}.$$

Lowness of Fewpaths for other counting classes has been proved in [6].

Another class that is also low for PP, is the probabilistic complexity class BPP, a class with a very different "flavour" as the Few classes.

Theorem 15: [14] BPP is low for PP.

Proof: Let L be in PP^A for a set $A \in BPP$. There is a nondeterministic polynomial time machine M and a polynomial p such that $(|x| = n)$

$$x \in L \Longleftrightarrow acc_M^A(x) \geq 2^{p(n)-1} + 1.$$

Since $\mathrm{P}^{\mathrm{BPP}} = \mathrm{BPP}$, [32], in every computation path of M only one question needs to be asked to a BPP oracle, and there is a predicate Q in BPP satisfying

$$x \in L \Longleftrightarrow \|\{y \in \Sigma^{p(n)} \mid Q(x,y)\}\| \geq 2^{p(n)-1} + 1.$$

It is well know that the probability of correctness in BPP can be amplified. We can find a predicate R in P and a polynomial q such that

$$\begin{aligned} Q(x,y) &\implies \|\{z \in \Sigma^{q(n)} \mid R(x,y,z)\}\| \geq (1 - 2^{-2p(n)})2^{q(n)} \\ Q(x,y) &\implies \|\{z \in \Sigma^{q(n)} \mid R(x,y,z)\}\| \leq 2^{-2p(n)}2^{q(n)} \end{aligned}$$

We now have

$$\begin{aligned} x \in L &\implies \|\{yz \in \Sigma^{p(n)+q(n)} \mid P(x,y,z)\}\| \geq (2^{p(n)-1} + 1)(1 - 2^{-2p(n)})2^{q(n)} \\ x \notin L &\implies \|\{yz \in \Sigma^{p(n)+q(n)} \mid P(x,y,z)\}\| \leq (2^{p(n)-1} + 2^{p(n)-1-2p(n)})2^{q(n)} \end{aligned}$$

and therefore L is in PP, since $2^{p(n)}2^{-2p(n)} < 1 - 2^{-2p(n)}$ (the sum over all error probabilities is less than the probability gained by one accepting path of M). □

Although BPP is low for PP, it does not seem to be low for $\oplus$P since there are relativizations under which this is not true [24]. The class of sparse sets in NP is another example of a low class for PP that does not seem to be low for $\oplus$P either [14]. A characterization of all the sets which are low for PP is an interesting open question. Such a characterization is known for the low sets for NP, [18], and for the low sets for $\oplus$P [16]. These classes are NP$\cap$co-NP and $\oplus$P, respectively.

3. Fewpaths as operator, $\Theta_2^p \subseteq$ PP

The class $\mathrm{P}^{\mathrm{NP[log]}}$ of languages accepted by polynomial time machines allowed to query an NP oracle at most $O(\log n)$ times has received a great deal of atention in recent years. The class has natural complete languages and many surprising characterizations, for example, it can be characterized as the class of sets truth-table reducible to NP, or as the class of sets log space Turing reducible to NP. Due to its many natural properties Wagner proposed to include this class in a refined version of the polynomial time hierarchy, denoting it by Θ_2^p [30]. The inclusions NP$\subseteq \Theta_2^p \subseteq \Delta_2^p$ are straightforward. We will show in this section that the class Θ_2^p is included in PP. This result was proved in an elegant way in [7] by showing that PP is closed under parity reductions. We present a different proof which can be seen as a generalization of theorem 14. This proof uses a new characterization of Θ_2^p in terms of "fewness".

We mentioned in the introduction that it can be fruitful to consider complexity classes acting as operators over classes of predicates. We use here the class Fewpaths to define an operator. A close look at this class shows that its languages can be decided computing the (bounded) number of strings of a set in P, and checking a polynomial time predicate. This can be naturally generalized to computing the number of strings of sets in other complexity classes.

Definition 16: Let $\mathcal{K}$ be a complexity class. A language L is in the class Fewpaths$\mathcal{K}$ if there is a set A in $\mathcal{K}$, a polynomial time predicate Q and two polynomials p, q such that for every $x \in \Sigma^\star$, the function $f(x) = \|\{\langle x, y\rangle \mid y \in \Sigma^{q(|x|)},\ \langle x, y\rangle \in A\}\|$ satisfies

i) $f(x) \leq p(|x|)$

ii) $x \in L \iff Q(x, f(x))$

Clearly, Fewpaths=FewpathsP. We see what happens if we apply the operator over the complexity class NP.

Theorem 17: FewpathsNP=Θ_2^p.

Proof: From left to right, let $L \in$ FewpathsNP and A be the set in NP for L. The function $f(x) = \|\{\langle x, y\rangle \mid y \in \Sigma^{q(|x|)},\ \langle x, y\rangle \in A\}\|$ can be computed doing binary search with just $O(\log n)$ queries in the NP set $\{\langle x, k\rangle \mid$ there are at least k strings $\langle x, y\rangle \in A\}$. The result in the other direction follows from a similar argument as the one from [15], used there to prove that the functions computable in Δ_2^p are reducible to optimization functions. Let L be a set in Θ_2^p, computed by a polynomial time machine M that for a constant c queries $c\log(n)$ many times the NP oracle B. Consider the set

$$A = \{\langle x, y\rangle \mid |y| = c\log(|x|) \text{ and there is a string } z = z_1 \ldots z_{|y|},\ z \geq y,$$
$$\text{which considered as a string of oracle answers for } M(x) \text{ satisfies}$$
$$\text{for every } i = 1 \ldots |y| \text{ the condition that } z_i = 1 \text{ if the } i\text{-th string}$$
$$\text{queried following the answers of } z \text{ is in } B\}.$$

Set A belongs to NP and for every string x the number of strings $\langle x, y\rangle$ in A is equal to the value of the greatest string (in lex. order) of oracle answers in $M(x)$ for which the "yes" answers are correct, which is exactly the string of correct answers. Observe that the number of strings $\langle x, y\rangle$ in A is bounded by $|x|^c$. Once the oracle answers are known the rest of the computation of M can be simulated by a polynomial time predicate. □

Using the new characterization of Θ_2^p we show now that this class in included in PP. For this result we need the following lemma which is not hard to prove.

Lemma 18: Let L be a language. $L \in$ PP iff there are two nondeterministic polynomial time machines M and M' such that

$$L = \{x \in \Sigma^* \mid acc_M(x) \geq acc_{M'}(x)\}.$$

Theorem 19: [7] $\Theta_2^p \subseteq$ PP

Proof: Let L be a set in Θ_2^p, by the above characterization there is a set A in NP, a predicate Q and two polynomials p, q such that for every $x \in \Sigma^\star$, $f(x) = \|\{\langle x, y\rangle \mid y \in$

$\Sigma^{q(|x|)},\ \langle x,y\rangle \in A\}\| \leq p(|x|)$ and $x \in L$ iff $Q(x,f(x))$. Let M_A be the nondeterministic Turing machine computing A. We construct machines M_1, M_2 such that

$$\begin{aligned} x \in L &\implies acc_{M_1}(\langle x, f(x)\rangle) > acc_{M_2}(\langle x, f(x)\rangle) \\ x \notin L &\implies acc_{M_1}(\langle x, f(x)\rangle) < acc_{M_2}(\langle x, f(x)\rangle) \end{aligned}$$

By the lemma, this inequalities imply the result.

M_k: **input** $\langle x, i\rangle$;
if $i > p(|x|)$ **then reject**;
$(*)$ **guess** i strings $\langle x, y\rangle$ in A;
$k = 1$: accept iff $Q(x,i)$
$k = 2$: accept iff $\neg Q(x,i)$

Step $(*)$ is implemented by
guess $y_1 < \cdots < y_i \in \Sigma^{q(|x|)}$;
guess $v_1 \ldots v_i$;
if : $\forall j:$ v_j is an accepting path of $M_A(\langle x, y_j\rangle)$
then continue
else reject;

Observe that machine M_k on input $\langle x, i\rangle$ does not have any accepting paths if either $f(x) < i$ or ($k = 1$ and $\neg Q(x,i)$) or ($k = 2$ and $Q(x,i)$). The number of accepting paths of these machines is at most exponential and there is a polynomial t such that

$$acc_{M_k}(\langle x, i\rangle) < 2^{t(|x|)}$$

We can define nondeterministic machines M_1', M_2' such that

$$acc_{M_k'}(x) = \sum_{i=0}^{p(|x|)} acc_{M_k}(\langle x, i\rangle) 2^{it(|x|)}.$$

Since 2^t is greater than the number of possible accepting paths of M_1 and M_2, the relation between the accepting paths of M_1' and M_2' depends only on $Q(x, f(x))$. We then have

$$\begin{aligned} x \in L &\implies acc_{M_1}(x, f(x)) > acc_{M_2}(x, m_x) \implies acc_{M_1'}(x) > acc_{M_2'}(x) \\ x \notin L &\implies acc_{M_1}(x, f(x)) < acc_{M_2}(x, m_x) \implies acc_{M_1'}(x) < acc_{M_2'}(x) \end{aligned}.$$

□

Observe that this last result is not a lowness result. Showing that Θ_2^p is low for PP would be equivalent to showing that NP is low for PP, and would imply PH $\subseteq$ PP. This result would be hard to prove since recently Beigel [5] has obtained a relativization in which P^{NP}, a slightly stronger class than Θ_2^p is not included in PP.

4. An unbounded number of solutions, lowness for P^{PP}

In section 2 we have seen that certain complexity classes computed by nondeterministic machines with a restricted number of accepting paths are low for PP. We will consider now classes in which the number of paths is unbounded, like NP or $\oplus$P. We will observe that the proof of Toda's result stating that the polynomial time hierarchy is Turing reducible to PP [23] shows in fact lowness of these classes for P^{PP}. The class P^{PP}, of sets polynomial time Turing reducible to PP, can be considered as the counting hierarchy analogon of the class Δ_2^p in the polynomial time hierarchy.

In order to prove the lowness of $\oplus P$ the following theorem is needed.

Theorem 20: [23] For every $L \in \oplus\mathrm{P}$, there is a polynomial p such that for every polynomial $q \geq p$ there is a nondeterministic polynomial time Turing machine M satisfying

$$x \in L \implies acc_M(x) = \alpha_x \cdot 2^{q(|x|)} - 1 \text{ for some integer } \alpha_x > 0$$
$$x \notin L \implies acc_M(x) = \alpha_x \cdot 2^{q(|x|)} \text{ for some integer } \alpha_x \geq 0$$

Observe that the statement of this theorem is similar to the one of theorem 8 for the class FewP. The difference is that here the exact number of accepting paths for machine M on an input x is not known, but it depends on integer α_x. There are relativizations under which a result like theorem 8 cannot be achieved for the case of classes with an unbounded number of accepting paths [26]. However the above result is strong enough for proving the lowness of $\oplus$P for P^{PP}.

Theorem 21: [23] $\oplus$P is low for P^{PP}.

Proof: We show that $\mathrm{PP}^{\oplus\mathrm{P}}$ is included in P^{PP}. The result follows taking the Turing reducibility closure of both classes.

Let L be a language in $\mathrm{PP}^{\oplus\mathrm{P}}$. Since $\mathrm{P}^{\oplus\mathrm{P}} = \oplus\mathrm{P}$ [16], a probabilistic machine with an oracle in $\oplus$P just needs to query the oracle once. Therefore, there is a language $A \in \oplus\mathrm{P}$ and polynomial p such that for every $x \in \Sigma^\star$

$$x \in L \iff \|\{\langle x, y\rangle \mid y \in \Sigma^{p(|x|)},\ \langle x, y\rangle \in A\}\| > 2^{p(|x|)-1}.$$

By the theorem above, there is a suitable polynomial $q > p$ and a nondeterministic machine M satisfying

$$\langle x, y\rangle \in A \implies acc_M(\langle x, y\rangle) = \alpha \cdot 2^{q(|\langle x,y\rangle|)} - 1 \text{ for some integer } \alpha > 0$$
$$\langle x, y\rangle \notin A \implies acc_M(\langle x, y\rangle) = \alpha \cdot 2^{q(|\langle x,y\rangle|)} \text{ for some integer } \alpha \geq 0.$$

Consider the nondeterministic machine M' which on input x guesses a string $y \in \Sigma^{p(|x|)}$ and simulates $M(\langle x, y\rangle)$. The number of accepting paths of M' is

$$acc_{M'(x)} = \beta \cdot 2^{q'(|x|)} - f(x),$$

for some integer β. $f(x)$ is the number of strings $\langle x, y\rangle$ in A and therefore $f(x) > 2^{p(|x|)-1}$ iff $x \in L$.

In order to decide whether x is in L, one can compute $acc_{M'(x)}$, which can be done by a deterministic polynomial time machine with an oracle in PP, making then one modulo $2^{q'(|x|)}$ operation and obtaining $f(x)$. □

We can now prove the lowness of NP and the whole polynomial time hierarchy for the class P^{PP}.

Theorem 22: [23] NP is low for P^{PP}.

Proof: The inclussion $\mathrm{NP} \subseteq \mathrm{BPP}^{\oplus \mathrm{P}}$ was shown in [28]. Since all the inclusion results we mentioned relativize, we have

$$\mathrm{P}^{\mathrm{PP}^{\mathrm{NP}}} \subseteq \mathrm{P}^{\mathrm{PP}^{\mathrm{BPP}^{\oplus \mathrm{P}}}} = \mathrm{P}^{\mathrm{PP}^{\oplus \mathrm{P}}} = \mathrm{P}^{\mathrm{PP}}$$

The second and third equalities follow from the lowness of BPP and $\oplus$P for PP and P^{PP} respectively. □

Corollary 23: [23] PH is low for P^{PP}.

Proof: We show by induction on the level of the hierarchy, k, the lowness of every class Σ_k^p in PH. For $k = 1$, Σ_1^p=NP, and the lowness of NP is shown in the previous theorem. For the induction step, let us suppose that Σ_k^p is low for P^{PP}. We have

$$\mathrm{P}^{\mathrm{PP}^{\Sigma_{k+1}^p}} = \mathrm{P}^{\mathrm{PP}^{\mathrm{NP}^{\Sigma_k^p}}} = \mathrm{P}^{\mathrm{PP}^{\Sigma_k^p}} = \mathrm{P}^{\mathrm{PP}}$$

the second equality holds since theorem 22 relativizes. □

Corollary 24: [23] $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{PP}}$.

We believe that it must be possible to make a direct proof of the lowness of NP for P^{PP}, without having to go through BPP and $\oplus$P, but to our knowledge such a proof has not been obtained yet. The last results show the power of the class P^{PP}, and it is natural to ask what other classes are low for it. It could be the case that PP is low for P^{PP}; such a result would imply the collapse of the counting hierarchy. It can be even the case P^{PP}=PSPACE, a statement which not too many people would have believed before Toda's result. Relativized separations of the contrary results (the separations) are not known either, and further research in the area seems promising.

Aknowledgements

I would like to thank Johannes Köbler for providing me with short proofs for the results in section 2, and José Balcazar, Josep Díaz and Birgit Jenner for helpful comments on the paper.

References

[1] E. Allender: The complexity of sparse sets in P. Proc. 1st Structure in Complexity Theory Conference, Lect. Notes in Comp. Sci., (1986) 1–11.

[2] E. Allender and K. Wagner: Counting hierarchies: polynomial time and constant depth circuits. *Bulletin of the EATCS* 40, (1990) 49–57.

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

[4] R. Beigel: Relativized counting classes: Relations among thresholds, parity and mods. *Journal of Comput. Syst. Sci.* To appear.

[5] R. Beigel: Polynomial interpolation, threshold circuits and the polynomial hierarchy. Manuscript, Jan. 1990.

[6] R. Beigel, J. Gill and U. Hertrampf: Counting classes: thresholds, parity mod and fewness. Proc. STACS 90, Lect. Notes in Comp. Sci., (1990) 49–57.

[7] R. Beigel, L. Hemachandra and G. Wechsung: On the power of probabilistic polynomial time. Proc. 4th Structure in Complexity Theory Conference, IEEE (1989) 225–230.

[8] A. Blass and Y. Gurevich. On the unique satisfiability problem. *Information and Control* 55 (1982), 80–88.

[9] J.Y. Cai, L.A. Hemachandra: On the power of parity polynomial time. Proc. STACS 89, Lect. Notes in Comp. Sci., (1989) 229–239.

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

[11] A.V. Goldberg, M. Sipser: Compression and ranking. Proc. 17th STOC Conference (1985), 440–448.

[12] K. Ko and U. Schöning: On circuit-size complexity and the low hierarchy in NP. *SIAM J. Comput.* 14 (1985), 41–51.

[13] J. Köbler: Strukturelle Komplexität von Anzahlproblemen. Ph.D. Thesis. Universität Stuttgart, (1989).

[14] J. Köbler, U. Schöning, S. Toda, J. Torán: Turing machines with few accepting computations and low sets for PP. Proc. 4th Structure in Complexity Theory Conference, IEEE (1989) 208–216.

[15] M.W. Krentel: The complexity of optimization problems. Proc. 18th STOC Conference, (1986) 69–76.

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

[17] C. Rackoff: Relativized questons involving probabilistic algorithms. *J. Assoc. Comput. Math.* 29 (1982) 261–268.

[18] U. Schöning: A low and a high hierarchy within NP. *Journal Comput. Syst. Sci.* 27 (1983), 14–28.

[19] U. Schöning: *Complexity and structure.* Lect. Notes in Comp. Sci. 211, Springer-Verlag (1985).

[20] U. Schöning: Probabilistic complexity classes and lowness. Proc. 2nd Structure in Complexity Theory Conference, IEEE, (1988), 2–8.

[21] U. Schöning: The power of counting. Proc. 3rd Structure in Complexity Theory Conference, IEEE, (1988), 1–9.

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

[23] S. Toda: On the computational power of PP and $\oplus$P, Proc. 30th FOCS Conference, (1989) 514–519.

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

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

[26] J. Torán: A combinatorial technique for separating counting complexity classes. Proc. 16th ICALP Conference, Lecture notes in Comp. Science, (1989), 733–745.

[27] L. Valiant: The relative complexity of cheking and evaluating. *Inform. Proc. Letters* 5 (1976), 20–23.

[28] L. Valiant and V. Vazirani: NP is as easy as detecting unique solutions. *Theoretical Comp. Science* 47 (1986), 85–93.

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

[30] K. Wagner: Bounded query computations. Proc. 3rd Structure in Complexity Theory Conference, IEEE, (1988), 260–277.

[31] K. Wagner and G. Wechsung: *Computational complexity*. Reidel (1986).

[32] S. Zachos: Robustness of probabilistic complexity classes under definitional perturbations. *Information an control* 54 (1982), 143–154.

[33] S. Zachos and H. Heller: A decisive characterization of BPP. *Information an control* 69 (1986), 125–135.