Turing Machines With Few Accepting Computations And Low Sets For PP

Johannes Köbler[a], Uwe Schöning[a], Seinosuke Toda[b], Jacobo Torán[c]

[a] *Abteilung Theoretische Informatik, Universität Ulm, 89069 Ulm, Germany*

[b] *Department of Computer Sciences, University of Electrocommunications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo 182, Japan*

[c] *Departamento L.S.I., U. Politècnica de Catalunya, Pau Gargallo 5, E-08028 Barcelona, Spain*

1 Introduction

The intractability of the complexity class NP has motivated the study of subclasses that arise when certain restrictions on the definition of NP are imposed. For example, the study of sparse sets in NP [Ma82], the study of the probabilistic classes whithin NP [Gi77], and the study of low sets in NP for the classes in the polynomial time hierarchy [Sc83], have been three main research streams in the area of complexity theory, and have clarified many structural aspects of the class NP.

In this paper we study two different ways to restrict the power of NP: We consider languages accepted by nondeterministic polynomial time machines with a small number of accepting paths in case of acceptance, and also investigate subclasses of NP that are low for complexity classes not known to be in the polynomial time hierarchy.

The first complexity class defined following the idea of bounding the number of accepting paths was Valiant's class UP (unique P) [Va76] of languages accepted by nondeterministic Turing machines that have exactly one accepting computation path for strings in the language, and none for strings not in the language. This class plays an important role in the areas of one-way functions and cryptography, for example in [GrSe84] it is shown that P$\neq$UP if and only if one-way functions exist. The class UP can be generalized in a natural way by allowing a polynomial number of accepting paths. This gives rise to the class FewP defined by Allender [Al85] in connection with the notion of P-printable sets.

We study complexity classes defined by such path-restricted nondeterministic polynomial time machines, and show results that exploit the fact that the machines for these classes have a bounded number of accepting computation paths. We will not only consider these subclasses of NP, namely UP and FewP, but also the class Few, an extension of FewP defined by Cai and Hemachandra [CaHe89], in which the accepting mechanism of the machine is more flexible.

The three classes UP, FewP and Few are all defined in terms of nondeterministic machines with a bounded number of accepting paths for every input string, but for the last two classes this number is not known beforehand, and can range over a space of polynomial size. We show in Section 3 that a polynomial number of accepting paths implies an exact number of such paths (for another machine). We prove that for every language L in the mentioned classes a polynomial time nondeterministic machine can be constructed that has exactly $f(x)+1$ accepting paths for strings x in L, and $f(x)$ accepting paths for strings x that are not in L where f is a polynomial time computable function. This fact extends a result in [CaHe89], where it was proved that the classes FewP and Few are included in $\oplus$P. From our result follows additionally that FewP and Few are contained in the counting class $\mathrm{C}_=\mathrm{P}$ (exact counting), [Wa86], thus answering a question proposed in [Sc88].

We use the above result to prove in Section 4 lowness properties of the class Few. The concept of lowness for the classes in the polynomial time hierarchy was first introduced in [Sc83]. This idea was translated to the classes in the counting hierarchy in [Tor88a] and [Tor88b]. Intuitively, a set A is low for a complexity class K if A does not increase the computational power of K when used as oracle; $K^A = K$. We prove that Few is low for the complexity classes PP, $\mathrm{C}_=\mathrm{P}$, and $\oplus$P (parity-P, [PaZa83]), showing $\mathrm{PP}^{\mathrm{Few}}=\mathrm{PP}$, $\mathrm{C}_=\mathrm{P}^{\mathrm{Few}} = \mathrm{C}_=\mathrm{P}$ and $\oplus\mathrm{P}^{\mathrm{Few}} = \oplus\mathrm{P}$.

In Section 5 we consider some other interesting sets that are low for the class PP. We prove that all sparse sets in NP, as well as the sets in the probabilistic class BPP are PP-low. The proofs of these results relativize, and as a consequence we obtain more complex sets than the above ones, that are also PP-low.

The lowness results are used in the last part of the paper to obtain positive relativizations of the questions $\mathrm{NP}\stackrel{?}{\subseteq}\mathrm{C}_=\mathrm{P}$, $\mathrm{NP}\stackrel{?}{\subseteq}\oplus\mathrm{P}$ and $\oplus\mathrm{P}\stackrel{?}{\subseteq}\mathrm{PP}$. The corresponding relativized classes have been separated in [Tor88a], and more recently in [Be88]. We show here that if the mentioned separations can be done using sparse oracles, then they imply absolute separations. Results of this kind (positive relativizations) have been obtained before for the case of the polynomial time hierarchy in [LoSe86] and [BaBoSc86] (see also [Sc85]).

2 Basic Definitions and Results

The notation used althrough the paper is the common one. We present here definitions of the less known complexity classes mentioned in this article.

Definition 2.1: For a nondeterministic machine M and a string $x \in \Sigma^*$, let $acc_M(x)$ be the number of accepting computation paths of M with input x. Analogously, for a nondeterministic oracle machine M, an oracle A, and a string $x \in \Sigma^*$, $acc_M^A(x)$ is the number of accepting paths of M^A with input x.

Definition 2.2: 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 $x \in \Sigma^*$,

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

$ii)$ $x \in L \iff acc_M(x) > 0$

By the definition, it is clear that UP$\subseteq$FewP$\subseteq$NP. Another interesting path-restricted class, which is not known to be in NP, is the class Few, an extention of FewP with a more powerful accepting mechanism. This class was introduced by Cai and Hemachandra in [CaHe89].

Definition 2.3: A language L is in the class Few if there is a nondeterministic polynomial time machine M, a polynomial time predicate Q, and a polynomial p such that for every $x \in \Sigma^*$,

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

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

It is obvious that FewP$\subseteq$Few. It was shown in [CaHe89] that this class is closed under bounded truth-table reductions.

We say that a nondeterministic polynomial time machine M is a Few machine if there is a polynomial p such that for every $x \in \Sigma^*$, $acc_M(x) \leq p(|x|)$.

By applying a binary search technique under the NP oracle $\{\langle x, k\rangle \mid acc_M(x) \geq k\}$ it is easy to see that Few $\subseteq \mathrm{P}^{\mathrm{NP}}[O(\log n)]$. This is in contrast to the following theorem.

Theorem: Few $\subseteq \mathrm{P}^{\mathrm{FewP}}$.

Proof: Let $L \in$ Few be witnessed by a Few machine M and a polynomial time predicate Q, i.e., $x \in L \iff Q(x, acc_M(x))$ for all $x \in \Sigma^*$. Let p be the polynomial time bound of M and let q be a polynomial with $acc_M(x) \leq q(|x|)$ for all $x \in \Sigma^*$. Consider the polynomial time predicate

$$R(x, y) \iff y \text{ is accepting path of } M \text{ on input } x.$$

Then we get $acc_M(x) = \|\{y \in \Sigma^{\leq p(|x|)} \mid R(x, y)\}\|$ for all $x \in \Sigma^*$. Define an oracle set in FewP as follows.

$$\text{PREFIX} = \{(x, y) \mid \exists z : R(x, yz)\}.$$

Using this set as oracle, the following oracle machine M' computes $acc_M(x)$ and determines the membership of x in L.

```
input: x;
S := {λ};  { λ is the empty string }
k := 0;
repeat
   T := ∅;
   for all y ∈ S do
      begin
         if R(x, y) then k := k + 1;
         if (x, y0) ∈ PREFIX then T := T ∪ {y0};
         if (x, y1) ∈ PREFIX then T := T ∪ {y1}
      end;
   S := T
until S = ∅;
if Q(x, k) then
   accept
else
   reject.
```

M' searches through the "prefix tree" whose nodes are labeled with elements of the set $\{y \mid (x, y) \in \mathrm{PREFIX}\}$. The search starts at the root and continues level by level, and the number of y with $R(x, y)$ is counted. Since $acc_M(x)$ is polynomially bounded, the cardinality of $\{y \mid \mathrm{PREFIX}(x, y)\}$ is also polynomially bounded in $|x|$. Therefore, the set T can only reach polynomial size in each application of the loop. Since the loop is repeated at most $(p(|x|) + 1)$ times, the algorithm operates in polynomial time. $\Box$

A language in Few can be recognized with only logarithmically many queries to an NP oracle. In contrast to this, using an oracle in FewP seems to require polynomially many queries.

Next we define the complexity classes PP, $\mathsf{C}_{=}\mathrm{P}$ and $\oplus$P that are also defined considering the number of computation paths of a nondeterministic machine, but in this case the number of paths is not necessarily polynomially bounded. These classes were first introduced in [Gi77],[Wa86], and [PaZa83], respectively.

Definition 2.4: A language L is in the class PP if there is a nondeterministic polynomial time machine M and a function $f \in \mathrm{FP}$ such that for every $x \in \Sigma^*$,

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

PP is called $\mathsf{C}\mathrm{P}$ in the notation of [Wa86]. This notation can be generalized to other language classes K; a language L is in $\mathsf{C}K$ if there is a function f in FP, a polynomial p and a set A in K such that for every x in Σ^*,

$$x \in L \iff ||\{y|\ |y| \leq p(|x|) \text{ and } \langle x, y\rangle \in A\}|| \geq f(x)$$

Definition 2.5: A language L is in the class $\mathrm{C}_=\mathrm{P}$ if there is a nondeterministic polynomial time machine M and a function $f \in$FP such that for every $x \in \Sigma^*$,

$$x \in L \Longleftrightarrow acc_M(x) = f(x).$$

Definition 2.6: A language L is in the class $\oplus$P if there is a nondeterministic polynomial time machine M such that for every $x \in \Sigma^*$,

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

It is known that Few$\subseteq \oplus$P [CaHe89] and $\mathrm{C}_=\mathrm{P}\subseteq$PP [Ru85]. In [Tor88a] relativizations are presented under which the classes NP, $\mathrm{C}_=\mathrm{P}$ and $\oplus$P are all incomparable.

Closely related to the language class PP, is the function class #P, defined by Valiant in [Va79]

Definition 2.7: A function $f : \Sigma^* \longrightarrow \mathbb{N}$ is in #P if there is a nondeterministic polynomial time machine M such that for every x in Σ^*, $f(x) = acc_M(x)$.

3 Few Accepting Paths Imply an Exact Number of such Paths

In this section we will show that for every Few machine M and every FP function $g : \Sigma^* \times \mathbb{N} \longrightarrow \mathbb{N}$, a nondeterministic polynomial time machine M' can be constructed with the property that for every input $x \in \Sigma^*$, M' has exactly $acc_{M'}(x) = g(x, acc_M(x)) + 2^{p(|x|)}$ accepting paths, for a certain polynomial p. From this result follows directly that the complexity class Few is included in $\mathrm{C}_=\mathrm{P}$ and $\oplus$P. First we introduce a technical lemma that will help us to handle the number of accepting paths of a nondeterministic machine.

Lemma 3.1: Let $b : \Sigma^* \times \Sigma^* \longrightarrow \mathbb{Z}$ be a function in FP, q a polynomial, and M a nondeterministic polynomial time machine. Then there is a nondeterministic polynomial time machine M' and a polynomial r such that for every $x \in \Sigma^*$,

$$acc_{M'}(x) = \sum_{k=0}^{q(|x|)} b(x,k)\binom{acc_M(x)}{k} + 2^{r(|x|)}$$

Proof: For machine M, there is a polynomial time predicate Q and a polynomial p such that for every input string x, $acc_M(x) = ||\{y \in \Sigma^{p(|x|)} \mid Q(x,y)\}||$. Consider machine M'' described by the following program:

input x;
guess $k,\ 0 \leq k \leq q(|x|)$;
if $b(x,k) = 0$ **then reject**
else
  guess $y \in \{1, \ldots, |b(x,k)|\}$;
  guess $y_1 < \ldots < y_k \in \Sigma^{p(|x|)}$;
  if $Q(x, y_i)$ for every i, $1 \leq i \leq k$
    then $test$:=**true**
    else $test$:=**false**;
  if ($test$ **and** $b(x,k) > 0$) **or** ($\neg test$ **and** $b(x,k) < 0$)
    then accept
    else reject.

For every guessed k, if $b(x,k)$ is positive, then $M''(x)$ has $b(x,k)\binom{acc_M(x)}{k}$ accepting paths, and it has $|b(x,k)| \cdot [\binom{2^{p(|x|)}}{k} - \binom{acc_M(x)}{k}]$ accepting paths if $b(x,k)$ is negative. Therefore, altogether $M''(x)$ has

$$b(x,0)\binom{acc_M(x)}{0} + b(x,1)\binom{acc_M(x)}{1} + \ldots +$$
$$+b(x,q(|x|))\binom{acc_M(x)}{q(|x|)} + h(x)$$

accepting paths where h is the function in FP defined by

$$h(x) = \sum_{k, b(x,k)<0} |b(x,k)| \cdot \binom{2^{p(|x|)}}{k}.$$

Clearly there is a polynomial r such that for every string $x \in \Sigma^*$, $h(x) \leq 2^{r(|x|)}$. We obtain the desired machine M' by increasing the number of accepting paths of M''. The computation tree of $M'(x)$ consists of two subtrees: one of them has exactly $2^{r(|x|)} - h(x)$ accepting paths, and the other one is the computation tree of $M''(x)$. $M'(x)$ has then $acc_{M'}(x) = 2^{r(|x|)} - h(x) + acc_{M''}(x) = 2^{r(|x|)} + \sum_{k=0}^{q(|x|)} b(x,k)\binom{acc_M(x)}{k}$ accepting paths. □

If the machine considered is a Few machine, then there is a polynomial q bounding acc_M, and for every x, $acc_M(x)$ can only take values in $\{0, \ldots, q(|x|)\}$. This fact, as we will see next, allows us to calculate for every function $g : \Sigma^* \times \mathbb{N} \longrightarrow \mathbb{N}$, values for $b(x,0), \ldots, b(x, q(|x|))$ satisfying

$$\sum_{k=0}^{q(|x|)} b(x,k)\binom{acc_M(x)}{k} = g(x, acc_M(x)) \qquad (*)$$

There are two important points to be taken into consideration in the calculation of b: In first place, the value of $\sum_{k=0}^{q(|x|)} b(x,k)\binom{m}{k}$ depends only on the values of $b(x,0), \ldots, b(x,m)$.

Therefore, if there are values for $b(x,0),\ldots,b(x,m)$, satisfying equality $(*)$ in the case $acc_M(x)\le m$, the above equality would hold independently of the values given to $b(x,m+1),\ldots,b(x,q(|x|))$. The second consideration is that after $b(x,0),\ldots,b(x,m)$ have been given values satisfying $(*)$ in case $acc_M(x)\le m$, a value for $b(x,m+1)$ can be found so that $(*)$ is also true in case $acc_M(x)=m+1$. This fact follows from the equality

$$\sum_{k=0}^{q(|x|)} b(x,k)\binom{m+1}{k} = \sum_{k=0}^{m} b(x,k)\binom{m+1}{k} + b(x,m+1)$$

from which the value of $b(x,m+1)$ can be obtained from $b(x,0),\ldots,b(x,m)$ and $g(x,m+1)$. To prove our result it is only left to show that if $g\in$FP, then the values of b can also be computed in polynomial time.

Theorem 3.2: For every Few machine M and every function g in FP from $\Sigma^*\times\mathbb{N}$ to $\mathbb{N}$, there is a nondeterministic polynomial time machine M' and a polynomial r such that for every $x\in\Sigma^*$, $acc_{M'}(x)=g(x,acc_M(x))+2^{r(|x|)}$.

Proof: Let q be a polynomial such that for every $x\in\Sigma^*$, $acc_M(x)\le q(|x|)$, and let $b:\Sigma^*\times\mathbb{N}\longrightarrow\mathbb{Z}$ be a function in FP satisfying for every $m, 0\le m\le q(|x|)$,

$$\sum_{k=0}^{q(|x|)} b(x,k)\binom{m}{k} = g(x,m)$$

By Lemma 3.1, there is a nondeterministic polynomial time machine M' and a polynomial r with

$$acc_{M'}(x) = \sum_{k=0}^{q(|x|)} b(x,k)\binom{acc_M(x)}{k} + 2^{r(|x|)} = g(x,acc_M(x)) + 2^{r(|x|)}$$

accepting paths. As stated above, $b(x,k)$ can be computed inductively:

$$b(x,0) := g(x,0)$$
$$b(x,k+1) := g(x,k+1) - \sum_{i=0}^{k} b(x,i)\binom{k+1}{i}$$

for $k=0,\ldots,q(|x|)-1$ and $b(x,k):=0$ for $k>q(|x|)$. It is clear that if the values of b do not become too large, then the function is in FP. We will see that these values are bounded. For a string $x\in\Sigma^*$ let $g_{\max}$ be the maximum of the values of $|g(x,k)|$, for $k=0,\ldots,q(|x|)$. We show by induction on k that

$$|b(x,k)| \le c_k := g_{\max}\cdot 2^{(\sum_{i=0}^{k} i)} = g_{\max}\cdot 2^{k(k+1)/2}$$

We have

$$
\begin{aligned}
|b(x,0)| &\leq g_{\max} = c_0, \\
|b(x,k+1)| &\leq g_{\max} + \sum_{i=0}^{k} |b(x,i)| \cdot \binom{k+1}{i} \\
&\leq g_{\max} + \sum_{i=0}^{k} c_i \binom{k+1}{i} \\
&\leq g_{\max} + c_k \sum_{i=0}^{k} \binom{k+1}{i} \quad = \quad g_{\max} + c_k(2^{k+1}-1) \\
&\leq c_k 2^{k+1} = c_{k+1}
\end{aligned}
$$

□

We use the above result to show the inclusion of Few in the classes $\mathrm{C_{=}P}$ and $\oplus$P.

Corollary 3.3: For every language L in Few there is a nondeterministic polynomial time machine M' and a function $f \in$FP such that for every string $x \in \Sigma^*$:
if $x \in L$ then $acc_{M'}(x) = f(x) + 1$
if $x \notin L$ then $acc_{M'}(x) = f(x)$

Proof: Let L be a language in Few, M a Few machine and Qa polynomial time predicate such that for every string x, $x \in L \Longleftrightarrow Q(x, acc_M(x))$. Define function g as

$$
g(x,m) = \begin{cases} 1 & \text{if } Q(x,m) \\ 0 & \text{if } \neg Q(x,m) \end{cases}
$$

By Theorem 3.2, there is a nondeterministic polynomial time machine M' and a polynomial r with $acc_{M'}(x) = g(x, acc_M(x)) + 2^{r(|x|)}$, therefore

$$
acc_{M'}(x) = \begin{cases} 2^{r(|x|)} + 1 & \text{if } x \in L \\ 2^{r(|x|)} & \text{if } x \notin L \end{cases}
$$

The result follows since the function f defined by $f(x) := 2^{r(|x|)}$ is in FP. □

Corollary 3.4:

i) Few $\subseteq \mathrm{C_{=}P}$

ii) Few $\subseteq \oplus$P [CaHe89]

Part *i*) of the corollary answers an open problem proposed in [Sc88].

4 Lowness of Few

We will see in this section that the class Few is low for the complexity classes PP, $\mathrm{C}_=\mathrm{P}$ and $\oplus$P. The concept of lowness for classes in the polynomial time hierarchy was introduced in [Sc83]. We extend the concept here to other complexity classes.

Definition 4.1: For a language L and a complexity class K (which has a senseful relativized version $K^{()}$), we will say that L is low for K (L is K–low) if $K^L = K$. For a language class C, C is low for K if for every language L in C, $K^L = K$.

In order to show the lowness properties of Few, first we need a lemma which states that a nondeterministic machine querying an oracle in Few can be simulated by another machine of the same type with the same number of accepting paths that queries just one string on every path to another oracle in Few.

Lemma 4.2: For every nondeterministic polynomial time machine M and every language $A\in$Few, there is a nondeterministic polynomial time machine M' and a language $A'\in$FewP such that for every $x \in \Sigma^*$, $acc_M^A(x) = acc_{M'}^{A'}(x)$ and $M'(x)$ queries just one string to the oracle in every computation path.

Proof: Let M be a polynomial time nondeterministic machine, with an oracle A in Few. There is a polynomial time predicate Q and a Few machine M'' such that for every $x \in \Sigma^*$, $x \in A \Longleftrightarrow Q(x, acc_{M''}(x))$.

Consider the nondeterministic oracle machine M' described by the following algorithm:

> **input** x;
> **guess** $w = (z, (q_1, y_1^1, \ldots, y_{i_1}^1), \ldots, (q_k, y_1^k, \ldots, y_{i_k}^k))$
> { computation path of M, queries made to the oracle following this path, and accepting computation paths of machine M'' for the guessed queries }
> **if** z is an accepting path for $M(x)$ in which exactly the sequence of oracle queries $q_1, \ldots, q_k$ is made, and every query q_j is answered "yes" if and only if $Q(q_j, i_j)$, and for every j, $y_1^j < \ldots < y_{i_j}^j$, and $y_1^j, \ldots, y_{i_j}^j$ are accepting paths of $M''(q_j)$ **then**
>   **if** $w \in A'$ **then** reject
>   **else** accept
> **end.**

The oracle for the algorithm is the set $A'\in$FewP

$$A' = \{(z, (q_1, y_1^1, \ldots, y_{i_1}^1), \ldots, (q_k, y_1^k, \ldots, y_{i_k}^k)) \mid \exists\, j, y \text{ such that}$$
$$y \text{ is an accepting path of } M''(q_j) \text{ and } y \neq y_1^j, \ldots, y_{i_j}^j\}$$

The algorithm guesses the accepting computation paths for the queries of M, and then checks that it has not guessed "too many" of these paths. Then, the query to A' (answered negatively) assures that all such paths have been guessed, and therefore membership in A of the queries made by machine M, is correctly decided. Observe that there is a polynomial p (depending on A and M) such that for every input string x, and every guessed string w in M' that leads to acceptance, $|w| \leq p(|x|)$, and therefore the machine runs in polynomial time. Note also that in every accepting computation path, the answer to the oracle has to be answered negatively.

Oracle set A' belongs to FewP since $A \in$Few, and therefore for every possible query q_j, there are at most a polynomial number of accepting paths for machine M'' with input q_j. □

Theorem 4.3: For every nondeterministic polynomial time oracle machine M and every language $A \in$Few, there is a nondeterministic polynomial time machine M' and a polynomial q such that for every $x \in \Sigma^*$, $acc_{M'}(x) = acc_M^A(x) + 2^{q(|x|)}$.

Proof: Let M be a nondeterministic polynomial time machine and A a language in Few. By (the proof of) Lemma 4.2, it is not hard to see that there is a predicate $R \in$FewP, and a polynomial p such that for every $x \in \Sigma^*$, $acc_M^A(x) = ||\{y \in \Sigma^{p(|x|)} \mid \neg R(x, y)\}||$.

By Theorem 3.2, there is a nondeterministic polynomial time machine M'' and a polynomial r such that for every pair (x, y), $M''(x, y)$ has exactly $2^{r(|(x,y)|)}$ accepting paths if $R(x, y)$ is true, and it has exactly $2^{r(|(x,y)|)} + 1$ accepting paths otherwise. Define a function h by $h(x) = 2^{r(|(x,0^{p(|x|)})|)}$, and consider the following nondeterministic machine M':

> With input x, M' guesses a string y of length $p(|x|)$. Then M' simulates M'' with input (x, y).

$M'(x)$ has then $2^{p(|x|)}h(x) + ||\{y \in \Sigma^{p(|x|)} \mid \neg R(x, y)\}|| = 2^{p(|x|)}h(x) + acc_M^A(x)$ accepting paths. A small modification of M' increases the number of its accepting paths, as in the proof of Lemma 3.1. Therefore, it follows that there is a polynomial q for which $acc_{M'}(x) = acc_M^A(x) + 2^{q(|x|)}$. □

A direct consequence of the above theorem is that Few is low for the classes PP, $\Subset$P and $\oplus$P.

Corollary 4.4:

i) Few is PP-low.

ii) Few is $\Subset$P-low.

iii) Few is $\oplus$P-low.

Observe that the last result in the corollary can also be obtained as a consequence of the results Few$\subseteq \oplus$P [CaHe89] and $\oplus\mathrm{P}^{\oplus\mathrm{P}} = \oplus$P [PaZa83].

It is not hard to see, looking at the proofs, that the above results relativize. More precisely, for every oracle set A, the classes $\mathrm{PP}^{\mathrm{Few}^A}$, $\mathrm{C_{=}P}^{\mathrm{Few}^A}$ and $\oplus\mathrm{P}^{\mathrm{Few}^A}$, are included in PP^A, $\mathrm{C_{=}P}^A$ and $\oplus\mathrm{P}^A$, respectively. We will make use of the relativized version of the results in Section 6.

5 Other Low Sets for PP

In this section we show that sparse sets in NP and BPP sets are low for PP. It is interesting to observe that these two classes of sets have also been shown to be low for complexity classes in the polynomial time hierarchy (NP$\cap$SPARSE is low for Δ_2^p, and BPP is low for Σ_2^p [KoSc86], [ZaHe86], [Sch85]), as opposed to the class Few which is not known to be low for any class in PH.

To obtain the results we need the following technical lemma which is straightforward to prove:

Lemma 5.1: Let $L \subseteq \Sigma^*$ be a language and A an oracle set. The following statements are equivalent:

i) L is in PP^A.

ii) There are two functions $f, g : \Sigma^* \longrightarrow \mathbb{N}$, $f \in$FP and $g \in \#\mathrm{P}^A$ such that

$$L = \{x \in \Sigma^* \mid g(x) > f(x)\}$$

iii) There are two functions $f, g : \Sigma^* \longrightarrow \mathbb{N}$, $f, g \in \#\mathrm{P}^A$ such that

$$L = \{x \in \Sigma^* \mid g(x) > f(x)\}$$

Theorem 5.2: NP$\cap$SPARSE is low for PP.

Proof: Let $A \in \mathrm{NP} \cap \mathrm{SPARSE}$ and p be a polynomial such that $\|A^{\leq n}\| \leq p(n)$. Let $A = L(M_A)$ for a nondeterministic machine M_A and let q be a polynomial time bound for M_A.
By Lemma 5.1, for $L \in \mathrm{PP}^A$ there is an NP oracle acceptor M and a function $f \in$ FP such that

$$\begin{aligned} x \in L &\implies acc_M^A(x) > f(x) \\ x \notin L &\implies acc_M^A(x) < f(x) \end{aligned}$$

Let r be a polynomial time bound for M and let $m_x := \|A^{\leq r(|x|)}\|$. We construct M_1, M_2 such that

$$\begin{array}{lcl} x \in L & \Longrightarrow & acc_{M_1}(\langle x, m_x\rangle) > acc_{M_2}(\langle x, m_x\rangle) \\ x \notin L & \Longrightarrow & acc_{M_1}(\langle x, m_x\rangle) < acc_{M_2}(\langle x, m_x\rangle) \end{array}$$

M_k: **input** $\langle x, i\rangle$;
if $i > p(r(|x|))$ **then reject**;
$(*)$ **guess** a set $S \subseteq A^{\leq r(|x|)}$, $\|S\| = i$;
$k = 1$: simulate $M^S(x)$.
$k = 2$: **guess** $y \in \{1, \ldots, f(x)\}$.

Step $(*)$ is implemented by

guess $a_1 < \cdots < a_i \in \Sigma^{\leq r(|x|)}$;
guess $w_1, \ldots, w_i \in \Sigma^{\leq q(r(|x|))}$;
if $\forall j :\ w_j$ is an accepting path of $M_A(a_j)$
then continue ("$S = \{a_1, \ldots, a_i\}$")
else reject;

There is a polynomial t such that

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

We can define NP acceptors M_1', M_2' such that

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

We then have

$$\begin{array}{lclcl} x \in L & \Longrightarrow & acc_{M_1}(x, m_x) > acc_{M_2}(x, m_x) & \Longrightarrow & acc_{M_1'}(x) > acc_{M_2'}(x) \\ x \notin L & \Longrightarrow & acc_{M_1}(x, m_x) < acc_{M_2}(x, m_x) & \Longrightarrow & acc_{M_1'}(x) < acc_{M_2'}(x) \end{array}$$

□

At this point, the natural question to ask is whether sparse sets in NP are also low for the classes C=P and ⊕P. We believe that this might not be the case, or at least would be very hard to prove, since in [Tor88a] it is shown that there is a relativization separating the class of sparse sets in NP from C=P and from ⊕P, and therefore these sets cannot be low for C=P and for ⊕P in the relativized case.

The proof technique from Theorem 5.2 can be used to show another interesting result related with bounded query classes [BoLoSe84]. Let $Q(M, x, A)$ denote the set of queries made by machine M with oracle A on input x. Let $Q(M, x) = \bigcup_A Q(M, x, A)$, and let $\mathrm{PP}_b(A)$ denote the class of sets accepted by oracle PP-machines satisfying that for some polynomial p, $||Q(M, x)|| \leq p(|x|)$, for every input x. By using a modification of the technique shown above, it can be proved that $\mathrm{PP}_b(\mathrm{NP})=\mathrm{PP}$. As an immediate consequence, we have $\mathrm{P}^{\mathrm{NP}[\log]}$ included in PP, a result that was first proved in [BeHeWe89]. These considerations will appear in the paper [Tod89].

We observe next that the probabilistic class BPP is also PP-low.

Theorem 5.3: BPP is low for PP.

Proof: Let L be in PP^A for a set $A \in BPP$. There is an NP oracle acceptor M and a polynomial $p \geq 1$ such that $(|x| = n)$

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

Because $P^{BPP} = BPP$, there is a BPP-predicate Q such that for all x, $|x| = n$,

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

By the amplification lemma for BPP we can find a P-predicate R and a polynomial q such that

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

We now have

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

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). □

We finish this section making the observation that the above two lowness proofs relativize (as well as the ones from previous sections), and one can use this fact to obtain "more complex" low sets. For example if a set A is low for PP, then if L is sparse and in NP^A, then L is also low for PP, since PP^L is in PP^A and therefore also in PP.

6 Positive Relativizations

The complexity classes NP, PP, $\mathrm{C\!\!\!=\!P}$ and $\oplus$P seem all to be different, although a proof of any separation would imply immediately P$\neq$PSPACE, and therefore the question is hard to answer. It is easier to separate the classes in relativized worlds; this has been done in [Tor88a] and in [Be88]. We will show here that if the relativized separation of the classes could be done using sparse oracles, then this would imply that the classes are different. Actually, the separation results in [Tor88a] are done with non-sparse oracles. These results are on the same line as the positive relativizations for the classes in the polynomial time hierarchy obtained in [LoSe86] and [BaBoSc86].

Definition 6.1: For a language A define the function $print_A : \{0\}^* \longrightarrow \Sigma^*$ as

$$print_A(0^n) = \langle a_1, a_2, \ldots, a_k \rangle$$

where $a_1, a_2, \ldots, a_k$ are the lexicographically first strings in A of length less than or equal to n.

Lemma 6.2: Let S be a sparse language. The function $print_S$ can be computed in polynomial time relative to an oracle in FewP^S.

Proof: For a sparse language S, consider the set

$$L_S = \{\langle y, z\rangle \mid \text{there is a string } w \in S, \text{such that } y \leq w < z \text{ (in lex. order)}\}$$

L_S is in FewP^S since for every string $\langle y, z\rangle$ there is only a polynomial number of strings between y and z in S, and therefore there is only a polynomial number of possible witnesses for membership of $\langle y, z\rangle$ in L_S. The function $0^n \mapsto print_S(0^n)$ can be computed in polynomial time by iterating a binary search process in L_S. □

Theorem 6.3:

i) $\mathrm{NP} \subseteq \mathrm{C_=P} \Longleftrightarrow$ for every sparse oracle S, $\mathrm{NP}^S \subseteq \mathrm{C_=P}^S$.

ii) $\mathrm{NP} \subseteq \oplus\mathrm{P} \Longleftrightarrow$ for every sparse oracle S, $\mathrm{NP}^S \subseteq \oplus\mathrm{P}^S$.

iii) $\oplus\mathrm{P} \subseteq \mathrm{PP} \Longleftrightarrow$ for every sparse oracle S, $\oplus\mathrm{P}^S \subseteq \mathrm{PP}^S$.

Proof: *i)* The direction from right to left is straightforward. For the other direction, let S be a sparse set and let A be a language in NP^S computed by a nondeterministic polynomial time machine M. Consider the set

$$A' = \{\langle x, a_1, \ldots, a_k\rangle \mid M \text{ accepts } x \text{ with oracle } \{a_1, \ldots, a_k\} \}$$

There is a polynomial q such that for every string $x \in \Sigma^*$,

$$x \in A \Longleftrightarrow \langle x, print_S(0^{q(|x|)})\rangle \in A'.$$

It is clear that $A' \in$NP and by the hypothesis, $A' \in \mathrm{C_=P}$. Therefore, by Lemma 6.2, in order to compute A we need first a computation in P^{FewP^S} to obtain $print_S(0^{q(|x|)})$, and then a $\mathrm{C_=P}$ predicate to decide whether $\langle x, print_S(0^{q(|x|)})\rangle$ belongs to A'. Therefore, $A \in \mathrm{C_=P}^{\mathrm{FewP}^S}$, but by the (relativized version of the) lowness results of Section 4, $\mathrm{C_=P}^{\mathrm{FewP}^S} = \mathrm{C_=P}^S$.

For *ii*) and *iii*), the proof is completely analogous, considering that by the results of Section 4, FewP is also low for $\oplus$P and for PP. $\square$

Acknowledgement

The third author would like to thank Osamu Watanabe for useful discussions about this investigation.

References

[Al85] E.W. Allender. *Invertible Functions.* Ph.D. dissertation, Georgia Inst. of Techn., 1985.

[BaBoSc86] J.L. Balcázar, R.V. Book, and U. Schöning. The polynomial-time hierarchy and sparse oracles. *Journ. Assoc. Comput. Mach.* 33 (1986): 603–617.

[Be88] R. Beigel. Relativized counting classes: Relations among thresholds, parity, and mods. Manuscript (1988).

[BeHeWe89] R. Beigel, L.A. Hemachandra and G. Wechsung. $\mathrm{P}^{\mathrm{NP}[\log]} \subseteq \mathrm{PP}$. *Forth Structure in Complexity Theory Conf.*, IEEE, 1989.

[BoLoSe84] R. Book, T.J. Long and A. Selman. Quantitative relativizations of complexity classes. *SIAM Jour. Comp.* 13 (1984): 461–487.

[CaHe89] J. Cai and L.A. Hemachandra. On the power of parity. *Symp. Theor. Aspects of Comput. Sci.*, Lecture Notes in Computer Science, Springer-Verlag, (1989), 229–240.

[Gi77] J. Gill. Computational complexity of probabilistic complexity classes. *SIAM Journ. Comput.* 6 (1977): 675–695.

[GrSe84] S. Grollmann and A.L. Selman. Complexity measures for public-key cryptosystems. *25th Symp. Found. Comput. Sci.*, 495-503, IEEE, 1984.

[KoSc86] K.I. Ko and U. Schöning. On circuit-size complexity and the low hierarchy in NP. *SIAM Jour. Comput.* 14 (1986): 41–51.

[LoSe86] T.J. Long and A.L. Selman. Relativizing complexity classes with sparse sets. *Journ. of the Assoc. Comput. Mach.* 33 (1986): 618–628.

[Ma82] S.A. Mahaney. Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis. *Journ. Comput. Syst. Sci.* 25 (1982): 130–143.

[PaZa83] C.H. Papadimitriou and S.K. Zachos. Two remarks on the power of counting. *6th GI Conf. on Theor. Comput. Sci.*, Lecture Notes in Computer Science 145, 269–276, Springer-Verlag, 1983.

[Sc83] U. Schöning. A low and a high hierarchy within NP. *Journ. Comput. Syst. Sci.* 27 (1983): 14–28.

[Sc85] U. Schöning. *Complexity and Structure.* Lecture Notes in Computer Science 211, Springer-Verlag, 1985.

[Sc88] U. Schöning. The power of counting. *Proc. 3rd Structure in Complexity Theory Conf.*, 2–9, IEEE, 1988.

[Tod89] S. Toda. On restricted access to NP oracles in probabilistic computations. Manuscrip, 1989.

[Tor88a] J. Torán. *Structural Properties of the Counting Hierarchies.* Doctoral dissertation, Facultat d'Informatica, UPC Barcelona, Jan. 1988.

[Tor88b] J. Torán. An oracle characterization of the counting hierarchy. *Proc. 3rd Struct. Complexity Theory Conf.*, 213–223, IEEE, 1988.

[Va76] L.G. Valiant. The relative complexity of checking and evaluating. *Inform. Proc. Lett.* 5 (1976): 20–23.

[Va79] L.G. Valiant. The complexity of computing the permanent. *Theoret. Comput. Sci.* 8 (1979): 410–421.

[Wa86] K.W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Inform.* 23 (1986): 325–356.

[ZaHe86] S. Zachos and H. Heller. A decisive characterization of BPP. *Information and Control* 69 (1986): 125–135.