A characterization of $PF^{NP\|} = PF^{NP[\log]}$

A. Lozano

Report LSI-93-7-R

A Characterization of $\mathrm{PF}^{\mathrm{NP}\|} = \mathrm{PF}^{\mathrm{NP}[\log]}$ *

Antoni Lozano
Universitat Politècnica de Catalunya

Abstract

Some implications of the fact $\mathrm{PF}^{\mathrm{NP}\|} = \mathrm{PF}^{\mathrm{NP}[\log]}$ are known in terms of equalities between complexity classes of sets, but a characterization in the same terms is still unknown. Here we provide a characterization in terms of polynomial-time p-enumerators.

1 Introduction

This note is centered on two functional classes defined as restrictions of $\mathrm{PF}^{\mathrm{NP}}$: the classes $\mathrm{PF}^{\mathrm{NP}\|}$ and $\mathrm{PF}^{\mathrm{NP}[\log]}$. In the first case we restrict the queries to be made in parallel (it is not allowed that a query depends on the answers of the others), and in the second one, we allow at most $O(\log n)$ serial queries for inputs of length n.

At a first glance, there may seem to exist a parallelism between these function classes and the set classes P^{NP}, $\mathrm{P}^{\mathrm{NP}\|}$, and $\mathrm{P}^{\mathrm{NP}[\log]}$. If we want to trace a correspondence between them, we should compare Δ_2^p with $\mathrm{PF}^{\mathrm{NP}}$ and the last two (which are equal and known as Θ_2^p) with $\mathrm{PF}^{\mathrm{NP}\|}$. This relationship is supported by the fact that $\Delta_2^p = \Theta_2^p$ if and only if $\mathrm{PF}^{\mathrm{NP}} = \mathrm{PF}^{\mathrm{NP}\|}$ [1].

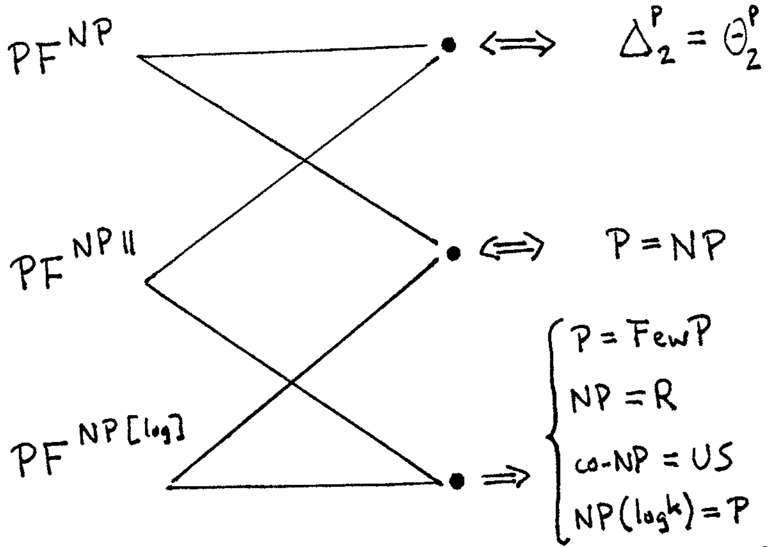
In the case of function classes, we have also the class $\mathrm{PF}^{\mathrm{NP}[\log]}$. This class was studied by Krentel who showed that it is different from $\mathrm{PF}^{\mathrm{NP}}$ unless P = NP [5]. It is also known that it cannot be equal to $\mathrm{PF}^{\mathrm{NP}\|}$ unless P = FewP, NP = R, [7, 6], co-NP = US, and $\mathrm{NP}(\log^k) = \mathrm{P}$ [3].

The following figure summarizes the known implications obtained from the equality

*This research was partially supported by ESPRIT-II Basic Research Actions Program of the European Community under contract No. 3075 (project ALCOM), and by *Acciones Integradas* No. 131B

[1] From right to left, restrict the functions to one bit. From left to right, apply the operator /bit [4] (consider the functions where every bit can be computed in the class). For an alternative proof, see Selman [6].

of these function classes.



It is still an open problem obtaining a characterization of the equality $\mathrm{PF}^{\mathrm{NP}\|} = \mathrm{PF}^{\mathrm{NP[log]}}$ in terms of set collapses like the above ones, or to prove some stronger collapse. Here we give a characterization in terms of polynomial-time p-enumerators which says, basically, that the NP oracle in $\mathrm{PF}^{\mathrm{NP[log]}}$ is not needed at all, and can be substituted by an arbitrary oracle, having the equivalence:

Corollary 2.5 $\mathrm{PF}^{\mathrm{NP}\|} \subseteq \mathrm{PF}^{\mathrm{NP[log]}} \iff \mathrm{PF}^{\mathrm{NP}\|} \subseteq \bigcup_X \mathrm{PF}^{X[\log]}$

2 Main Result

We use the definition of s-enumerators from [1].

Definition 2.1 *Let $f : \Sigma^* \to \mathbb{N}$, and $s : \mathbb{N} \to \mathbb{N}$ be two functions. The function E is an s-enumerator for f if $\forall x\ E(x)$ is a list of at most $s(|x|)$ strings in which $f(x)$ appears.*

We say that a function has a p-enumerator if it has a p-enumerator for some polynomial p.

The concepts of polynomial-time p-enumerators and Kolmogorov simple images for functions are connected in [2]. This connection can also be written in the following way.

Theorem 2.2 $\mathrm{PF}^{\mathrm{NP}\|} \subseteq \bigcup_X \mathrm{PF}^{X[\log]}$ *if and only if* $\mathrm{PF}^{\mathrm{NP}\|}$ *has polynomial-time p-enumerators.*

Proof *If.* Let $f \in \mathrm{PF}^{\mathrm{NP}\|}$ and E be a p-enumerator for f. Define

$X = \{\langle x, i, b\rangle \mid f(x)$ appears in the j-th position in $E(x)$ and the i-th bit of j is $b\}$.

It is clear that we can compute $f(x)$ in polynomial time by making queries to X.

Only if. Let f be a function in $\mathrm{PF}^{\mathrm{NP}\|}$. Then, there exists some polynomial-time transducer M that, for any input x of length n, computes $f(x)$ by making $O(\log n)$ many queries to some oracle X. A polynomial-time p-enumerator for f would generate a list with the output of $M(x)$ corresponding to every possible sequence of answers to the queries (which are polynomially many). □

Note that the set X defined in the previous proof belongs to the class Θ_2^p. Now we prove that there exists another such set belonging to NP which provides a characterization of the equality $\mathrm{PF}^{\mathrm{NP}\|} \subseteq \mathrm{PF}^{\mathrm{NP}[\log]}$ in terms of polynomial-time p-enumerators.

Theorem 2.3 $\mathrm{PF}^{\mathrm{NP}\|} \subseteq \mathrm{PF}^{\mathrm{NP}[\log]}$ *if and only if* $\mathrm{PF}^{\mathrm{NP}\|}$ *has polynomial-time p-enumerators.*

Proof *If.* Let f be a function in $\mathrm{PF}^{\mathrm{NP}\|}$ and call M to the polynomial-time oracle transducer that computes it by making queries to SAT. We define a polynomial-time algorithm computing f that makes $O(\log n)$ queries to the oracle

$$Y = \{\langle x_1, \ldots, x_s\rangle \# b_1 \ldots b_s \mid \forall i, 1 \leq i \leq s,\ \chi_{\mathrm{SAT}}(x_i) = b_i\}$$

on inputs of length n. Then, we show that these queries can be answered with only $O(\log n)$ queries to SAT.

Fix an input x for f and suppose that $x_1, \ldots, x_s$ are the strings queried to the NP oracle by $M(x)$. Let g be the function that outputs $\chi_{\mathrm{SAT}}(x_1) \cdots \chi_{\mathrm{SAT}}(x_s)$ on x. It is clear that g belongs to the class $\mathrm{PF}^{\mathrm{NP}\|}$ and then, by the hypothesis, it has a polynomial-time p-enumerator, say E.

Let $E(x) = (w_1, \ldots, w_m)$, and consider the matrix S that has the w_i's as its rows; that is to say, row i in S is w_i and column j corresponds to the value of the query x_j in the different outputs of the p-enumerator E (associating the value 0 to the answer NO and 1 to YES). We denote with $S_{i,j}$ the bit located in row i and column j.

The number of rows of S is polynomial in the size of the input x. As each row of the matrix is a possible answer vector for $x_1, \cdots, x_s$ and we know that one of them is correct, we are interested in deleting incorrect rows and thus reduce them to one. The following algorithm reduces the size of the matrix S in a constant factor of 4.

(1) Suppose that there is a column in S, j, such that its number of zeroes varies between $\frac{1}{4}s$ and $\frac{3}{4}s$. In this case, if we find out the membership of x_j to SAT, we can delete at least $\frac{1}{4}$ of the rows. Thus, we make the query $x_j\#0$ to Y.

(2) Suppose that every column in S has strictly less than $\frac{1}{4}s$ or more than $\frac{3}{4}s$ zeroes. Let b_j be the bit that appears more times in column j. We say that row i is *average up to column* j if for all k, $1 \leq k \leq j$, $S_{i,k} = b_k$. Row i is *average* if it is average up to column m.

We consider two subcases:

(2.1) Assume that at least $\frac{1}{4}$ of the rows are average. In this case, we ask $\langle x_1, \ldots, x_m\rangle \# b_1 \cdots b_m$ to Y. If the answer is YES, then we know the answer to all the queries, and the algorithm halts. If the answer is NO, then the row corresponding to the correct answers is not average, and we can delete at least $\frac{1}{4}$ of the rows.

(2.2) Here assume that less than $\frac{1}{4}$ of the rows are average. So, we have that more than $\frac{3}{4}s$ rows are average up to column 1, by the hypothesis of case (2), while less than $\frac{1}{4}$ of them are average up to column m. Taking into account that at most $\frac{1}{4}$ of the rows which are average up to a certain column can lose this condition in the next column, there must exist a column j_0, $1 \leq j_0 \leq m$, such that the number of rows which are average up to j_0 is between $\frac{1}{4}s$ and $\frac{3}{4}s$.

We only have to know if the correct row is average up to j_0 or not in order to delete $\frac{1}{4}$ of the rows. We can do this by asking $\langle x_1, \ldots, x_{j_0}\rangle \# b_1 \ldots b_{j_0}$ to the oracle.

If we apply repeteadly the above algorithm to S, we can reduce the rows to one by making only $O(\log n)$ queries to Y.

Assume that $w = \langle x_1, \ldots, x_m\rangle \# b_1 \cdots b_m$ is a query to Y. Define w_0 (w_1) as the boolean formula that consists of the *or* (*and*) of the x_i's such that $b_i = 0$ ($b_i = 1$). It holds that $x \in Y \iff w_0 \notin \text{SAT} \wedge w_1 \in \text{SAT}$. Therefore, for each query to Y we only need to make two queries to SAT, and the above algorithm can work with $O(\log n)$ queries to SAT.

Only if. This direction follows from the previous theorem. □

The two preceeding theorems give us the following equivalence.

Corollary 2.4 $\text{PF}^{\text{NP}\|} \subseteq \text{PF}^{\text{NP[log]}} \iff \text{PF}^{\text{NP}\|} \subseteq \bigcup_X \text{PF}^{X\text{[log]}}$

3 Discussion and Further Research

We can see two interesting paths to continue this research: obtaining a new characterization of $\text{PF}^{\text{NP}\|} = \text{PF}^{\text{NP[log]}}$ in terms of set collapses and/or getting stronger implications specially with a stronger hypothesis such as $\text{PF}^{\text{NP}\|} = \text{PF}^{\text{NP}\|\text{[log]}}$

In the first case, it is noted in [6] that in the proofs that obtain set collapses from $\text{PF}^{\text{NP}\|} = \text{PF}^{\text{NP[log]}}$, the power of NP in $\text{PF}^{\text{NP[log]}}$ is not used, and it is argued that, because of this reason, we may be far obtaining a complete characterization of this problem in terms of set collapses. Now we know by Corollary 2.4 that the power of NP in $\text{PF}^{\text{NP[log]}}$ would not be needed at all in any characterization of the above equality. In this sense, Corollary 2.4 can be useful to obtain such a characterization.

As for the second point, note that the proof of Theorem 2.2 also shows the following fact.

Corollary 3.1 $\mathrm{PF}^{\mathrm{NP}\|} \subseteq \bigcup_X \mathrm{PF}^{X\|[\log]}$ *if and only if* $\mathrm{PF}^{\mathrm{NP}\|}$ *has polynomial-time p-enumerators.*

Although the requirement of nonadaptiveness can be made for an arbitrary oracle (X in the corollary), we would like to know what are the implications if we require nonadaptiveness to the queries to NP, that is to say, whether the inclusion $\mathrm{PF}^{\mathrm{NP}\|} \subseteq \mathrm{PF}^{\mathrm{NP}\|[\log]}$ is equivalent to the above ones or we can derive stronger collapses from it.

References

[1] J. -Y. Cai, L. Hemachandra. Enumerative Counting is Hard. *Information and Computation* **82** (1989) 34–44.

[2] F. Green and J. Torán. Kolmogorov Complexity of #P Functions. Research Report LSI-91-2, Universitat Politècnica de Catalunya, 1991.

[3] B. Jenner and J. Toán. Computing with Parallel Queries to NP. Research report LSI-93-8-R, Universitat Politècnica de Catalunya, 1993.

[4] J. Kõblerand T. Thierauf. Complexity Classes with Advice. *Fifth Annual IEEE Structure in Complexity Theory Conference* (1990) 305–315.

[5] M. Krentel. The Complexity of Optimization Problems. *J. Computer and System Sci. 36* (1988) 490–509.

[6] A. Selman. A Note on Adaptive vs. Nonadaptive reductions to NP (preliminary draft). Research Report. Boston University, July, 1990.

[7] S. Toda. On Polynomial-Time Truth-Table reducibilities of Intractable Sets to P-Selective Sets. *Mathematical Systems Theory* **24** (1991), 69–82.

Departament de Llengatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

List of research reports (1993).

LSI-93-1-R "A methodology for semantically enriching interoperable databases", Malú Castellanos.

LSI-93-2-R "Extraction of data dependencies", Malú Castellanos and Fèlix Saltor.

LSI-93-3-R "The use of visibility coherence for radiosity computation", X. Pueyo.

LSI-93-4-R "An integral geometry based method for fast form-factor computation", Mateu Sbert.

LSI-93-5-R "Temporal coherence in progressive radiosity", D. Tost and X. Pueyo.

LSI-93-6-R "Multilevel use of coherence for complex radiosity environments", Josep Vilaplana and Xavier Pueyo.

LSI-93-7-R "A characterization of $PF^{NP\|} = PF^{NP[\log]}$", Antoni Lozano.

LSI-93-8-R "Computing functions with parallel queries to NP", Birgit Jenner and Jacobo Torán.

LSI-93-9-R "Simple LPO-constraint solving methods", Robert Nieuwenhuis.

LSI-93-10-R "Parallel approximation schemes for problems on planar graphs", Josep Díaz, María J. Serna, and Jacobo Torán.

Internal reports can be ordered from:

Nuria Sánchez
Departament de Llenguatges i Sistemes Informàtics (U.P.C.)
Pau Gargallo 5
08028 Barcelona, Spain
secrelsi@lsi.upc.es