MATHEMATICAL PROBLEMS IN COMPUTATION THEORY
BANACH CENTER PUBLICATIONS, VOLUME 21
PWN – POLISH SCIENTIFIC PUBLISHERS
WARSAW 1988

ELEMENTARY PROPERTIES OF THE CLASS OF NONDETERMINISTIC POLYNOMIAL TIME COMPUTABLE FUNCTIONS

JAMES P. JONES

Department of Mathematics and Statistics,
University of Calgary, Calgary, Alberta, Canada

The well-known problem of Cook [2] and Karp [5] as to whether $P = NP$ is usually formulated in terms of relations (predicates). P usually denotes the set of relations decidable in deterministic polynomial time and NP denotes the set of relations decidable in nondeterministic polynomial time. In this paper we consider the analogous problem for functions. Suppose PF denotes the set of all functions computable in deterministic polynomial time and NPF denotes the set of all functions computable in nondeterministic polynomial time. One might expect that the $PF = NPF$ problem is just a restatement of the $P = NP$ problem, i.e., that the two problems are equivalent. This may in fact actually be the case. However, that would imply that the $P = NP$ and $P = NP \cap coNP$ problems are equivalent. This is because in this paper we prove that the $PF = NPF$ problem is equivalent to the $P = NP \cap coNP$ problem.

THEOREM 1. *$PF = NPF$ if and only if $P = NP \cap coNP$.*

The classes PF and NPF have very natural, computationally reasonable definitions, invariant under different computational models, Turing machines, random access machines, etc. An vague definition of NPF can be found in paper [12]. However, there the distinction between function and relation is not clearly made. It seems to us important to distinguish these two concepts. As an example of the difficulties which could arise, by blurring this distinction, consider the functions $y = 2^x$, $y = x!$ and $y = \begin{bmatrix} 2x \\ x \end{bmatrix}$. If we could obtain values $f(x)$ for these three functions, $y = f(x)$, in polynomial time, then a constant c, would exist such that $|f(x)| \leqslant |x|^c$ holds for all x, where $|x|$ denotes the (binary) *length* of x. Of course this is not the case. So these functions should not belong to *PF*. However, considered as relations, their graphs belong to *P*. One can decide in polynomial time, given x and y whether $y = f(x)$ because when both x and y are input, the time is a polynomial function of $|x|$ and $|y|$ (or $|x|+|y|$).

To give exact definitions for the classes *PF* and *NPF* it is convenient to use the notion of register machine. One has to be somewhat careful because the ordinary (Minsky [10]) definition of Register Machine (Program Machine) is not polynomial time equivalent to the Turing Machine. However (see paper [4]), one can obtain polynomial time equivalence by adding two new command, (41) and (42) to the usual (Minsky [10]) list. This same approach Jones–Matijasevič [4] can also be used to give an exact register machine definition of *NPF*. We need only append the nondeterministic *branch* command ((51)) on page 827). Of course, the value of a nondeterministic computation is often ambiguous. For computability of functions it is therefore necessary to append some type of requirement on uniqueness of output. There are several ways to do this. One simple and computationally very natural way to do it is to limit the class of register machines to those which have at most one (and hence exactly one) accepting computation on each input. The important thing is that all inputs x are accepted and produce the same unique value y.[1]

Thus *PF* is the class of (total, single valued) functions computable in polynomial time and *NPF* is the class of (total, single valued) functions computable in nondeterministic polynomial time. From the definitions given here it should be possible to prove that *PF* is the same as the class $\mathscr{L}$ of Cobham [1] and the same as the class Π of Frumkin [3]. One can show that the class *NPF* is equal to the class $N\Pi \cap N\Pi^c$, where $N\Pi$ and $N\Pi^c$ are the (nondeterministic) function classes of Frumkin [3].

To further clarify the definitions we will give examples of functions in *PF* and *NPF*. It will be seen that many natural, well-known number theoretic functions are examples.

Functions known to be in PF

$$x+y, \quad x\cdot y, \quad x \dot{-} y, \quad \lfloor x/y \rfloor, \quad \lceil \sqrt{x} \rceil,$$

$$\gcd(x, y), \quad \operatorname{lcm}(x, y), \quad \operatorname{rem}(x, y), \quad \operatorname{rem}(x^y, z), \quad \operatorname{rem}\left(\begin{bmatrix} y \\ x \end{bmatrix}, 2\right),$$

$$\left(\frac{x}{y}\right), \quad d(y, i), \quad |x|, \quad 2^{|x|}, \quad 2^{|x|\cdot|y|}, \quad x \,\&\, y.$$

[1] It is also possible to define the class *NPF* using Turing machines with oracles. *NPF* is exactly the class of functions computed in polynomial time by Turing machines with sets from $NP \cap \mathrm{co}\,NP$ in the oracle.

Here $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the *floor* and *celling* of x. rem(x, y) denotes the remainder after x is divided by y. The function $d(y, i) = \text{rem}(\lfloor y/2^i \rfloor, 2)$ denotes the ith binary *digit* of y. The function $|x| = \lceil \log_2(x+1) \rceil$ is the length of x. The symbol & denotes *logical and*, $\left(\frac{x}{y}\right)$ denotes the Jacoby symbol.

Functions known to be in NPF

$\Phi(x) =$ The number of integers $\leqslant x$ and relatively prime to x .

$\tau(x) =$ The number of positive divisors of x.

$\sigma(x) =$ The sum of the divisors of x.

$\varrho(x) =$ The smallest prime divisor of x.

$\chi_P(x) =$ The characteristic function of the set of primes.

$\lambda(d) = 1$ or 0 according as $x^2 - dy^2 = -1$ has a solution or not.

The first three examples are the classical number theoretic functions. For example, the first function mentioned above is Euler's ϕ-function. A proof that these six functions belong to *NPF* would use the theorem of Pratt [11] that the set of primes belongs to $NP \cap \text{co}\, NP$. The last example requires also work of Lagarias [6] who proved that the set $\{d\colon x^2 - dy^2 = -1$ has a solution$\}$ belongs to $NP \cap coNP$. In this last example, one can of course also use the ordinary Pell equation, $x^2 - dy^2 = 1$. However, if this is done then the function $\lambda(d)$ would be in *PF*. As it stands, at present it is not known whether any of the given six functions lies in *PF*. These are open problems. A positive solution to any one of the first five would have the interesting consequence that the set of primes belongs to P (also an open question).

Examples of relations in P

$x \leqslant y$,	$z = x \cdot y$,	$\{x\colon \exists z\ x = z^2\}$,
$x \mid y$,	$z = x^y$,	$\{x\colon \exists y, z\ x = y^z\}$,
$z = \left[\dfrac{y}{x}\right]$,	$z = x!$	*The set of Mersenne primes.*

Some other known results which help to clarify the relationship between P and NP are the following. First one can mention the result of Karp [5]: *A predicate A belongs to NP if and only if there exists a predicate B in P such that for all natural numbers x*

$$x \in A \Leftrightarrow (\exists y)[|y| \leqslant |x|^c \text{ and } B(x, y)]. \tag{1}$$

Another, newer characterization of sets in NP is the following: Let A be any number theoretic predicate (relation). Then $A \in NP$ *if and only if there exist functions F and G, constructible from* + (*addition*), $\cdot$ (*multiplication*) *and* & (*logical and*), *and such that for all natural numbers x*

$$x \in A \Leftrightarrow (\exists y_1, \ldots, y_n)[|y_i| \leqslant |x|^c \text{ and } F(x, y_1, \ldots, y_n) = G(x, y_1, \ldots, y_n)]. \tag{2}$$

For a proof of this see paper [4]. A similar theorem is proved in [8]. (An important remaining open problem in this subject is whether the logical and operation, &, can be deleted from (2).)

As might be expected, the relationships between P, PF, NP and NPF are closely connected with the relationship between a function and its graph. So we consider next the graph. In this paper we shall denote the graph of the function f by $G(f)$. Note that the graph, $G(f) = \{(x, y)\colon y = f(x)\}$ is a relation. We shall also consider the relation $H(f) = \{(x, y)\colon y \leqslant f(x)\}$ and the relation $D(f) = \{(x, i)\colon \mathrm{rem}(\lfloor f(x)/2^i \rfloor, 2) = 1\}$. Note that for the relation $D(f)$ we have $(x, i) \in D(f)$ if and only if the ith binary digit of $f(x)$ is 1.

First we take up the case where the function f is a characteristic function (0-1 valued function). In this paper the characteristic function of a set A is denoted by χ_A. The definition is that $\chi_A(x) = 1$ or 0 according as $x \in A$ or not. $(\exists c)[|f(x)| \leqslant |x|^c]$ is shorthand for $(\exists c, d)(\forall x)[|f(x)| \leqslant d|x|^c]$.

Lemma 1. $\chi_A \in PF \Leftrightarrow A \in P$.

Lemma 2. $\chi_A \in NPF \Leftrightarrow A \in NP \cap coNP$.

Lemma 3. $f \in PF \Rightarrow G(f) \in P \,\&\, (\exists c)[|f(x)| \leqslant |x|^c]$.

Lemma 4. $f \in NPF \Leftrightarrow G(f) \in NP \,\&\, (\exists c)[|f(x)| \leqslant |x|^c]$.

Proof of 4. The direction $\Rightarrow$ is trivial. For $\Leftarrow$ suppose $G(f) \in NP$ and $|f(x)| \leqslant |x|^c$. Let M be a nondeterministic machine which in nondeterministic polynomial time decides whether $(x, y) \in G(f)$. Let M' be a machine which, on input x, chooses y, with $|y| \leqslant |x|^c$ and uses M to check whether $(x, y) \in G(f)$ and, if so, gives output y. Then M' nondeterministically computes f. Hence $f \in NPF$.

Lemma 5. *For* $|f(x)| \leqslant |x|^c$, $G(f) \in NP \Leftrightarrow D(f) \in NP \cap coNP$.

Proof. In the direction $\Rightarrow$, we may suppose that we have been given a nondeterministic machine M_0 which accepts (x, y) when $y = f(x)$. From M_0 it is easy to construct a nondeterministic machine M accepting (x, i) when $d(y, i) = 1$ and a nondeterministic machine M' accepting (x, i) when $d(y, i) = 0$. Conversely, for the $\Leftarrow$ direction, given such machines M and M' one can construct M_0.

Lemma 6. $f \in PF \Leftrightarrow D(f) \in P \,\&\, (\exists c)[|f(x)| \leqslant |x|^c]$.

Proof. The direction $\Rightarrow$ is trivial. For the $\Leftarrow$ direction, suppose $D(f) \in P$. Then the characteristic function of $D(f)$ is in PF, i.e., the function $d'(x, i)$ = the ith binary digit of $f(x)$. If i is sufficiently large, say, $i \geqslant |x|^c \geqslant |f(x)|$, then $d'(x, i) = 0$. So, for any x, the value of $f(x)$ can be obtained from $d'(x, i)$ by

$$f(x) = \sum_{i=0}^{|x|^c} d'(x, i)\, 2^i.$$

Hence the function $f \in PF$.

LEMMA 7. $f \in NPF \Leftrightarrow D(f) \in NP \cap coNP \,\&\, (\exists c)[|f(x)| \leqslant |x|^c]$.

Proof. By Lemma 4 and Lemma 5.

THEOREM 1. $PF = NPF \Leftrightarrow P = NP \cap coNP$.

Proof. In the direction $\Rightarrow$, use Lemmas 1 and 2. In the direction $\Leftarrow$, use Lemmas 6 and 7.

Remark. In Lemma 5 we can replace $D(f)$ by $H(f)$. (The proof then becomes easier. The $\Rightarrow$ direction is trivial. The direction $\Leftarrow$ uses only $(x, y) \in G(f) \Leftrightarrow (x, y) \in H(f)$ & $(x, y+1) \notin H(f)$.) In Lemma 6 one can also replace $D(f)$ by $H(f)$. However then the proof becomes more difficult.

(*Proof.* $\Leftarrow$ Fix x and put $h(y) = \chi(x, y)$, where χ here denotes the characteristic function of $H(f)$. Let $n = |x|^c$. Then $|f(x)| \leqslant n$, so $f(x) < 2^n$. Define a function g, by bounded recursion: Put $g(1) = 0$ and put $g(i+1) = g(i) + h(g(i) + 2^{n-i})2^{n-i}$. Then $f(x) = g(n+1)$. This proves $\Leftarrow$. (The direction $\Rightarrow$ is again trivial.))

From the above remark it follows also that we can replace $D(f)$ by $H(f)$ in Lemma 7. Hence the entire proof of Theorem 1 can be carried out using only $H(f)$ and not mentioning $D(f)$ at all. $H(f)$ is very useful when considering the relation between NPF and M. A. Frumkin's classes $N\Pi$ and $N\Pi^c$. One can prove that when f is a polynomially bounded function, i.e., $|f(x)| \leqslant d|x|^c$, then $f \in N\Pi \Leftrightarrow H(f) \in NP$ and also $f \in N\Pi^c \Leftrightarrow H(f) \in coNP$. Hence Lemma 7 implies $NPF = N\Pi \cap N\Pi^c$.

We next define a function f which possesses a certain type of completeness property.

DEFINITION.

$$f(a, m) = \begin{cases} \min z\,[z^2 \equiv a (\operatorname{mod} m)] & \text{if } a \text{ is a quadratic residue mod } m, \\ m & \text{if } a \text{ is a quadratic nonresidue mod } m. \end{cases}$$

LEMMA 8. *For the above function f we have $H(f) \in coNP$.*

Proof. Note that f can be defined by bounded minimalization:

$$f(a, m) = \min z\,[(m-z)\operatorname{rem}(z^2-a, m) = 0].$$

Hence

$$f(a, m) < y \Leftrightarrow (\exists z < y)[(m-z)\operatorname{rem}(z^2-a, m) = 0].$$

Also

$$f(a, m) < y \Leftrightarrow [m < y] \vee (\exists z)[z < y \wedge z^2 \equiv a (\operatorname{mod} m)].$$

THEOREM 2. *For the above function f, we have $f \in PF \Leftrightarrow P = NP$.*

Proof. This follows from the theorem of Adleman and Manders [8], [9] to the effect that the problem $(\exists z)[z \leqslant c \wedge z^2 \equiv a (\operatorname{mod} m)]$ is NP-complete. For $m \neq 0$, $c < m$, this problem is equivalent to $f(a, m) \leqslant c$ and hence reducible to the problem of evaluating f.

The function f is complete in the sense that if $f \in PF$, then $P = NP$. Conversely, if $P = NP$, then $f \in PF$ by Lemma 8 and Lemma 6 (with $D(f)$ replaced by $H(f)$).

THEOREM 3. *For the above function f, if $f \in NPF$, then*

$$P = NP \Leftrightarrow P = NP \cap coNP.$$

Proof. By Theorems 1 and 2.

Remark. It is an open problem whether the function f belongs to NPF. Evidently f belongs to M. A. Frumkin's class $N\Pi^c$. However, it is not known whether the function f belongs to Frumkin's class $N\Pi$. The closely related function $h(a, m) = m \dot{-} f(a, m)$ apparently belongs $N\Pi$ for it is easily obtained by bounded maximilization, $h(a, m) = \max u\left[u \cdot \operatorname{rem}((m-u)^2 - a, m) = 0\right]$. This function h can also replace f in Theorems 2 and 3. Hence it would be interesting to know if Frumkin's classes $N\Pi$ and $N\Pi^c$ are closed under composition.

Acknowledgements. These results constitute a portion of a lecture delivered November 14, 1985 in the Stefan Banach Centre in connection with the Semester on Mathematical Problems in Computation Theory organized by the Polish Academy of Sciences. The author wishes to thank Prof. Helena Rasiowa of the Polish Academy of Sciences for the opportunity of presenting these results, Prof. S. A. Cook of the Department of Computer Science, University of Toronto and Dr. S. R. Buss of the Math. Sciences Research Center, Berkeley, for valuable advice on the subject of this paper.

References

[1] A. Cobham, *The intrinsic computational difficulty of functions*, in Logic, Methodology and Philosophy of Science, Y. Bar-Hillel, Editor, North-Holland, Amsterdam 1965, 24–30.

[2] S. A. Cook, *The complexity of theorem proving procedures*, Proc. Third Annual ACM Symposium on the Theory of Computing, May 1971, 151–158.

[3] M. A. Frumkin, *Complexity question in number theory*, in Theory of Computational Complexity, D. Yu. Grigoriev and A. O. Sliscenko editors, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. 118 (1982), 188–210 (in Russian). English translation: Jour. Soviet Math. 29 (1985), 1502–1517.

[4] J. P. Jones and Y. V. Matijasevič, *Register machine proof of the theorem on exponential diophantine representation of enumerable sets*, Journal of Symbolic Logic 49 (1984), 818–829. MR 85i 03139.

[5] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, editors, Plenum Press, New York 1972, 85–103. MR 51 14644.

[6] J. C. Lagarias, *On the computational complexity of determining the solvability or unsolvability of the equation $x^2 - dy^2 = -1$*, Trans. Amer. Math. Soc. 260 (1980), 485-508.

[7] K. Manders and L. Adleman, *NP-complete decision problems for quadratic polynomials*, Proc. 8th Annual ACM Symposium on the Theory of Computing (1976), 23–29.

[8] —, —, *Diophantine complexity*, Proc. 17th Annual IEEE Symposium on Foundations of Computer Science (Houston, Texas, 1976), IEEE Computer Society, Long Beach, California, 1976, 81–88, MR 56 #7314.

[9] –, –, *NP-Complete decision problems for binary quadratics*, Jour. Computer and System Sciences 16 (1978), 168–184.

[10] M. Minsky, *Computation: Finite and Infinite Machines*, Prentice Hall, Englewood Cliffs, New Jersey 1967. MR50 9050.

[11] V. R. Pratt, *Every prime has a succinct certificate*, SIAM Jour. Computing 4 (1975), 214–220.

[12] L. G. Valient, *Relative complexity of checking and evaluating*, Information Processing Letters 5 (1976), 20–23.

Presented to the semester
Mathematical Problems in Computation Theory
September 16–December 14, 1985