Theoretical Computer Science 2 (1976) 73–76.

MINIMAL PAIRS OF POLYNOMIAL DEGREES WITH SUBEXPONENTIAL COMPLEXITY*

Michael MACHTEY

Departments of Mathematics and Computer Sciences, Purdue University, West Lafayette, Ind. 47907, USA

Communicated by A. Meyer
Received March 1975

Abstract. The goal of extending work on relative polynomial time computability from computations relative to sets of natural numbers to computations relative to arbitrary functions of natural numbers is discussed. The principal techniques used to prove that the honest subrecursive classes are a lattice are then used to construct a minimal pair of polynomial degrees with subexponential complexity; that is two sets computable by Turing machines in subexponential time but not in polynomial time are constructed such that any set computable from both in polynomial time can be computed directly in polynomial time.

1. Introduction

The preponderence of the work on relative computational complexity, particularly that on relative polynomial time computability, deals with computations relative to sets. There are significant conceptual and technical difficulties in trying to extend this work to computations relative to arbitrary functions. It might be hoped that results and techniques from subrecursive classes (e.g., the elementary classes) would extend to relative polynomial time computability, and some work in this direction has been begun by Constable [1] and Mehlhorn [5]. Despite the fact that this work on proposed notions of relative polynomial time computation for arbitrary functions is as yet incomplete, the techniques used to study the honest subrecursive classes can be used to obtain results about polynomial time computability relative to sets.

In this paper the principal techniques used to prove that the honest subrecursive classes are a lattice (Machtey [4]) will be used to construct a minimal pair of polynomial degrees with subexponential complexity. That is, two sets A and B will be constructed such that both A and B can be computed in subexponential time but not in polynomial time and such that if C is any set computable from both A and B in polynomial time then C can be computed directly in polynomial time. A construction of a minimal pair of polynomial degrees was obtained independently

* An earlier version of this work was presented at the Project MAC Conference on Concrete Complexity, August, 1973. The research was supported in part by NSF Grant No. GJ27127A1.

by Ladner [3] using quite different techniques and yielding much higher bounds on the complexity of the sets constructed. Because of the extremely low bounds on the complexity of minimal pairs constructed with the methods of this paper, it is highly likely that there are minimal pairs of polynomial degrees computable in nondeterministic polynomial time (assuming of course that $\mathcal{P} \neq \mathcal{NP}$; i.e., that there is a set computable in nondeterministic polynomial time which is not computable in deterministic polynomial time). Thus there would be totally "independent" problems solvable in nondeterministic but not in deterministic polynomial time.

2. Results

We restrict our goals to the construction of minimal pairs, which enables us to simplify definitions and notation and to utilize previous work. The natural numbers $\mathbf{N}$ are represented in binary, and for $x \in \mathbf{N}$, $|x|$ denotes the length of (the representation of) x. We deal with computable functions from $\mathbf{N}$ to $\mathbf{N}$; $f^{(n)}$ stands for the n-fold composition of f with itself, while f^n stands for the function whose value at x is $(f(x))^n$; we use x^2, etc. to stand for the function whose value at x is x^2, etc.; and subsets of $\mathbf{N}$ are identified with their characteristic functions. We define

$$\bar{f}(n) = \max\{|f(x)|: |x| = n\}.$$

All computations are by multi-tape Turing machines (with oracles if appropriate), and the time of a computation is the number of basic machine steps executed (including oracle interrogations). The time of a computation is always measured as a function of the length of the input. $\mathrm{Comp}(g) < f$ means that there is a Turing machine which computes g which on any input of length n takes time less than $f(n)$. Finally, let $\mathcal{P} = \{f: \mathrm{Comp}(f) < x^n + m$ for some $n, m \in \mathbf{N}\}$.

For any function f we define $\mathcal{M}(f)$ to be the smallest set of functions containing f and $x^2 + 1$ which is closed under composition; $g < \mathcal{M}(f)$ means that g is bounded by some member of $\mathcal{M}(f)$. Paralleling Constable [1] and Mehlhorn [5], we define $\mathcal{L}(f)$ to be the set of all functions computable from f in time bounded by some member of $\mathcal{M}(\bar{f})$ by an oracle Turing machine with separate input and output tapes for the oracle. The separate input and output tapes for the oracle insure that a computation is "charged" for the length of every oracle output which is used. We need the following simple facts about these definitions:

(1) $A \subseteq \mathbf{N}$ implies $\mathcal{L}(A)$ is the set of functions computable from A in polynomial time;

(2) $g \in \mathcal{L}(f)$ and $\bar{f}$ nondecreasing imply $\mathcal{L}(g) \subseteq \mathcal{L}(f)$;

(3) $g \in \mathcal{L}(f)$, $\mathrm{Comp}(f) < t$ and t nondecreasing imply $\mathrm{Comp}(g) < \mathcal{M}(t)$;

(4) $h \geq x^2 + 1$, h nondecreasing, and $g < \mathcal{M}(h)$ imply $g < h^{(n)}$ for some n.

The first and fourth facts are clear. To establish the second fact let $h \in \mathcal{L}(g)$, T_h be a Turing machine computing h from g in time $< \mathcal{M}(\bar{g})$, and T_g be a Turing machine computing g from f in time $< \mathcal{M}(\bar{f})$. If we replace the g-oracle on T_h by T_g we get a "machine" which computes h from f in time $< \mathcal{M}(\bar{f})$ (note that $\bar{g} < \mathcal{M}(\bar{f})$). A Turing machine which simulates this "machine" introduces at most a polynomial loss of time for the simulation. The third fact is established by a similar

standard argument which replaces an f-oracle by a Turing machine which computes f, remembering that f cannot be computed in time less than $\bar{f}$.

Assume that f and g are nondecreasing functions which cannot be computed in polynomial time. Define $F = \max(x^2+1, \bar{f})$ and $G = \max(x^2+1, \bar{g})$, and assume further that $M = \min(F, G)$ is bounded by a polynomial and that $\mathrm{Comp}(f) < F^{(n)}$ for some n and $\mathrm{Comp}(g) < G^{(n)}$ for some n (we might term this last property p-honesty). We can now make some observations about functions with the properties assumed for f and g. From (2) we know that $\mathscr{P} \subseteq \mathscr{L}(f) \cap \mathscr{L}(g)$, and we want to conclude that in fact $\mathscr{P} = \mathscr{L}(f) \cap \mathscr{L}(g)$.

If $h \in \mathscr{L}(f) \cap \mathscr{L}(g)$ then (3) and (4) together with the p-honesty of f and g yield that $\mathrm{Comp}(h) < F^{(n)}$ for some n and $\mathrm{Comp}(h) < G^{(n)}$ for some n. Thus for some n, $\mathrm{Comp}(h) < \min(F^{(n)}, G^{(n)})$ by the "almost-parallel" computation property of Turing machine time: A Turing machine can simulate two Turing machine computations of the same value in parallel and obtain the value in time not much greater than the faster of the two computations. To conclude that $\mathrm{Comp}(h)$ is bounded by a polynomial, we apply the following lemma. This lemma was brought to the author's attention by Robert Solovay, and it also provides a considerable simplification of the proof that the elementary honest classes are a lattice by providing a nearly trivial proof of Lemma 3.3 in [4] for the case where the reducibility is "elementary in".

Lemma. *Let $F, G \geq x$ be nondecreasing functions, and let $M = \min(F, G)$. Then for any $n \geq 1$, $\min(F^{(n)}, G^{(n)}) \leq M^{(2n-1)}$.*

Proof. In computing $M^{(2n-1)}(x)$, either F is used at least n times or G is used at least n times. The lemma follows by the properties assumed for F and G. □

Now assume that $A \in \mathscr{L}(f)$ and $B \in \mathscr{L}(g)$ such that $A, B \notin \mathscr{P}$ (such sets will exist if $\bar{f}$ and $\bar{g}$ are infinitely often large enough to allow diagonalization over $\mathscr{P}$). Since $\mathscr{L}(f) \cap \mathscr{L}(g) = \mathscr{P}$, by (1) and (2) we have that if $C \in \mathscr{L}(A) \cap \mathscr{L}(B)$ then $C \in \mathscr{P}$. Thus A and B will give us a minimal pair of polynomial degrees.

We now construct specific functions f and g which satisfy our assumptions. Let $h(0) = 0$ and $h(x+1) = 2^{h(x)}$. If $h(2y) \leq x < h(2y+1)$ for some y, let $f(x) = x$ and $g(x) = 2^{h(2y)}$. If $h(2y+1) \leq x < h(2y+2)$ for some y, let $f(x) = 2^{h(2y+1)}$ and $g(x) = x$. Note that f and g are nondecreasing, bounded by 2^x, and each equals 2^x for infinitely many arguments; moreover, $\min(f, g) = x$. If we define F and G as above, then F and G are also non-decreasing, bounded by 2^x, and each equals 2^x infinitely often; moreover, $\min(F, G) = x^2 + 1$. Furthermore, f and g are p-honest: $\mathrm{Comp}(f) < F^2$ and $\mathrm{Comp}(g) < G^2$. In fact, if we use the straightforward computational methods of first locating x between successive values of h and then applying the appropriate case, we get running times for f and g which are less than $\bar{f}\,|\bar{f}|^2$ and $\bar{g}\,|\bar{g}|^2$, respectively.

Finally, by standard compression techniques there are sets $A, B \notin \mathscr{P}$ such that $\mathrm{Comp}(A) \leq \mathrm{Comp}(f) < F^2 \leq 2^{2x}$ and such that $\mathrm{Comp}(B) \leq \mathrm{Comp}(g) < G^2 \leq 2^{2x}$. [For example see Hartmanis and Stearns [2]: $2^{|x|^2}$ is honest and majorizes the

polynomials, therefore $\mathcal{P}$ is contained in the complexity class determined by $2^{|x|^2}$; since $\mathrm{Comp}(f)$, $\mathrm{Comp}(g) \geq 2^{\cdot}$ infinitely often we have

$$\inf_{n\to\infty} \frac{(2^{|x|^2})^2}{\mathrm{Comp}(f)} = 0 \quad \text{and} \quad \inf_{n\to\infty} \frac{(2^{|x|^2})^2}{\mathrm{Comp}(g)} = 0,$$

and thus the complexity classes determined by $\mathrm{Comp}(f)$ and $\mathrm{Comp}(g)$ contain sets not in $\mathcal{P}$.] Since $\mathrm{Comp}(A) < F^2$ we have $A \in \mathcal{L}(f)$ (A is computable from f in time bounded by $\mathcal{M}(\bar{f})$ without even consulting the oracle), and since $\mathrm{Comp}(B) < G^2$ we have $B \in \mathcal{L}(g)$. We have proved

Theorem. *There are sets $A, B \notin \mathcal{P}$ such that* $\mathrm{Comp}(A)$, $\mathrm{Comp}(B) < 2^{2^x}$ *and such that if C is a set computable from both A and B in polynomial time then $C \in \mathcal{P}$.*

It is now straightforward to use the methods presented above to construct minimal pairs of polynomial degrees with even lower bounds on their complexity; for example, we could get a minimal pair with complexity bounded by $2^{|x|^{\|x\|}}$. Details are omitted out of compassion for the typesetter.

References

[1] R. L. Constable, Type two computational complexity, *Proc. Fifth ACM Symp. Theory Computing* (1973) 108–121.

[2] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms, *Trans. Am. Math. Soc.* **117** (1965) 285–306.

[3] R. E. Ladner, On the structure of polynomial time reducibility, *J.ACM* **22** (1975) 155–171.

[4] M. Machtey, The honest subrecursive classes are a lattice, *Information and Control* **24** (1974) 247–263.

[5] K. Mehlhorn, Polynomial and abstract subrecursive classes, *Proc. Sixth ACM Symp. Theory Computing* (1974) 96–109.