

Some Complexity Issues on the Simply Connected Regions of the Two-Dimensional Plane

(*Extended Abstract*)

Arthur W. Chou¹ and Ker-I Ko²

March 9, 1993

1 Introduction

This paper studies the computational complexity of subsets of the plane \mathbf{R}^2 . We propose a general framework in which continuous problems in computational complex analysis can be studied in the context of discrete complexity theory (i.e., the NP theory). This framework is based on the bit-operation model used in recursive analysis [Pour-El and Richards, 1989] and complexity theory of real functions of Ko and Friedman [1982]. It is an extension of the polynomial-time measure theory studied in Chapter 5 of Ko [1991].

The fundamental notion in this study is the class of bounded subsets of the plane \mathbf{R}^2 whose membership problem is polynomial-time solvable. We define two such notions: the polynomial-time approximable sets and the polynomial-time recognizable sets. Informally, a subset $S \subseteq \mathbf{R}^2$ is *polynomial-time approximable* if there is a machine M which, on a given point $\mathbf{z} \in \mathbf{R}^2$ and an integer n , determines whether \mathbf{z} is in S within time polynomial in n and admitting errors only on a set $E \subseteq \mathbf{R}^2$ of measure 2^{-n} . A subset $S \subseteq \mathbf{R}^2$ is *polynomial-time recognizable* if there is a machine M which on a given point $\mathbf{z} \in \mathbf{R}^2$ and an integer n , determines whether \mathbf{z} is in S within time polynomial in n and admitting errors only on points \mathbf{z} that are within a distance 2^{-n} of the *boundary* of S .³

¹Department of Mathematics and Computer Science, Clark University, Worcester, MA 01610; e-mail: achou@vax.clarku.edu.

²Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11794; e-mail: keriko@sbcs.sunysb.edu. Part of this research was done while this author was visiting Department of Computer Science of Princeton University. Research of this author is supported in part by NSF grant CCR 9121472.

To demonstrate that these two notions of polynomial-time computable sets are natural and interesting, we show that a strong version of polynomial-time recognizable sets (with only one-sided errors) characterizes precisely the zeros of polynomial-time computable functions defined on \mathbf{R}^2 (with a polynomial inverse modulus at zeros). In addition, the relation between the above two notions of polynomial-time computable sets has a close connection to the well-known open question of whether polynomial-time probabilistic computation (on integers) is strictly stronger than polynomial-time deterministic computation (on integers).

A basic object in computational complex analysis is a bounded, simply connected region; that is, a bounded, connected open set with no hole (or, equivalently, whose complement is connected). The boundary curve is a natural representation for such a set. We study some fundamental issues concerning such regions with polynomial-time boundary curve representations. The most critical problem is the *membership problem*: must a simply connected region with a polynomial-time computable boundary curve be polynomial-time recognizable? This question turns out to be closely related to the *winding number problem* of computing the winding numbers of a given curve. Not surprisingly, we apply the notion of $\#P$ -completeness to characterize the complexity of these problems. Namely, if a curve is polynomial-time computable then the winding number problem is solvable in polynomial-time using a function $f \in \#P$ as an oracle; conversely, a polynomial-time computable curve could be designed in such a way that its winding number problem encodes a discrete $\#P$ -complete problem. For the membership problem, it immediately follows that it can also be solved in polynomial time relative to an oracle $f \in \#P$. In addition, we can show that it is not necessarily polynomial-time recognizable, unless weak one-way functions do not exist (more precisely, $P = UP$). It remains an open ques-

³The notion of a machine running in polynomial time and the notion of the boundary of a set S will be formally defined in Sections 2 and 3.

tion whether the gap between the upper bound $P^{\#P}$ and the lower bound UP could be narrowed.

In addition to the membership problem, we also investigate a few other important questions involving simply connected regions. The *distance problem* is to determine the distance between a simply connected region (with a polynomial-time computable boundary curve) and a given point in \mathbf{R}^2 . This problem is equivalent to the minimization problem on one-dimensional real functions, and we can show that the distance problem is polynomial-time solvable if and only if $P = NP$. The *area problem* is to determine the measure of a simply connected region (represented by a polynomial-time computable boundary curve). It is shown that the area problem for regions whose boundary curves are rectifiable (i.e., having a finite length) is polynomial-time solvable if and only if $FP = \#P$. As a surprise, however, we show that if the boundary curve is not rectifiable, then the area problem is much harder: there exists a polynomial-time computable function f defining a simple closed curve Γ (i.e., f is one-to-one on $[0, 1]$ except that $f(0) = f(1)$) whose interior has a nonrecursive measure. The third problem is the *length problem* (or, the *coastline problem*) which asks us to determine the length of a given polynomial-time computable, rectifiable curve. We show, again, a strong negative result: there exists a polynomial-time computable simple curve which has a finite length but the length is a nonrecursive real number. The proofs of the last two results are based on the modified Peano space-filling curves which have recently found interesting applications in the construction of fractal sets [Mandelbrot, 1983]. (Our particular construction is similar to the curves called the Lebesgue-Osgood monsters in Mandelbrot [1983].)

The above results demonstrate an interesting relation between the continuous problems on two-dimensional regions and the complexity theory of discrete computation. It is a first step toward a general complexity theory of complex analysis.

Remarks on notation. We write $\ell(w)$ to denote the length of a character string w , and write $\|A\|$ for the cardinality of a finite set A . The notation $|x|$ is reserved for the absolute value of a real or a complex number.

2 Computational Model for Real Functions

The basic computational objects in continuous computation are dyadic rationals $\mathbf{D} = \{m/2^n : m \in \mathbf{Z}, n \in \mathbf{N}\}$. We let \mathbf{D}_n denote the class of dyadic rationals of the form $m/2^n$. For real numbers x and y , we write $\langle x, y \rangle$ to denote a point in the plane \mathbf{R}^2 . We often write \mathbf{z} to denote a point in \mathbf{R}^2 . For convenience, we use the L_∞ -metric for the space \mathbf{R}^2 ; thus, for two

points $\mathbf{z}_1 = \langle x_1, y_1 \rangle, \mathbf{z}_2 = \langle x_2, y_2 \rangle \in \mathbf{R}^2$, $|\mathbf{z}_1 - \mathbf{z}_2| = \max\{|x_1 - x_2|, |y_1 - y_2|\}$.⁴ For any point $\mathbf{z} \in \mathbf{R}^2$ and any nonempty set $S \subseteq \mathbf{R}^2$, we let $\delta(\mathbf{z}, S)$ be the distance between \mathbf{z} and S ; i.e., $\delta(\mathbf{z}, S) = \inf\{|\mathbf{z} - \mathbf{y}| : \mathbf{y} \in S\}$. We write $N(\mathbf{z}; \epsilon)$ to denote the open neighborhood of the point \mathbf{z} of radius ϵ ; using the L_∞ -metric, this is the open square centered at \mathbf{z} and having length 2ϵ at each side. For a set $S \subseteq \mathbf{R}^2$, we let $\mu^*(S)$ be the outer measure of S , and let $\mu(S)$ be the measure of S if S is measurable. We let S^c denote the complement of set S .

The concept of polynomial-time computable real functions used in this paper was first introduced in Ko and Friedman [1982] (see Ko [1991] for a complete treatment). A real number $x \in \mathbf{R}$ is represented by a *Cauchy function* $\phi : \mathbf{N} \rightarrow \mathbf{D}$ that binary converges to x in the sense that $|\phi(n) - x| \leq 2^{-n}$. A real number x is *recursive*, or *computable*, if there exists a computable Cauchy function ϕ that binary converges to x . A real number x is *polynomial-time computable* if there exists a polynomial-time computable Cauchy function ϕ that binary converges to x (i.e., $\phi(n)$ can be computed by a Turing machine in $p(n)$ moves for some polynomial p). In the following, we will write “P-computable” to mean the term “polynomial-time computable.”

The computational model for real functions is the oracle Turing machines.

Definition 2.1 (a) A function $f : [0, 1] \rightarrow \mathbf{R}$ is recursive, or computable, if there exists an oracle Turing machine M such that for any oracle function ϕ that binary converges to a real number $x \in [0, 1]$, and any integer n , the machine M outputs a dyadic rational e such that $|e - f(x)| \leq 2^{-n}$. In other words, the oracle machine computes the operator that maps a Cauchy function for x to a Cauchy function for $f(x)$.

(b) A function $f : [0, 1] \rightarrow \mathbf{R}$ is polynomial-time computable (or, P-computable) if it is computable by an oracle Turing machine that operates in polynomial time (i.e., $M^\phi(n)$ always halts in $p(n)$ moves for some polynomial p).

An equivalent definition for P-computable real functions will be used in this paper. We say a function $f : [0, 1] \rightarrow \mathbf{R}$ has a *polynomial modulus* if there exists a polynomial p such that $|x - y| \leq 2^{-p(n)}$ implies $|f(x) - f(y)| \leq 2^{-n}$.

Proposition 2.2 A function $f : [0, 1] \rightarrow \mathbf{R}$ is P-computable iff

- (i) f has a polynomial modulus, and
- (ii) there exist a Turing machine M and a polynomial p such that for any integer n and any dyadic ra-

⁴The choice of L_∞ -metric does not affect our main results. Indeed, all results in this paper also hold with respect to the usual L_2 -metric.

tional $d \in \mathbf{D}_n$, $M(d, n)$ outputs, in time $p(n)$, a dyadic rational e such that $|e - f(d)| \leq 2^{-n}$.

The above definition and proposition can be extended to functions mapping $[0, 1]$ to \mathbf{R}^2 or functions mapping $[0, 1]^2 \rightarrow \mathbf{R}$ in a natural way.

Remark. The above computational model for real functions is based on the model in recursive analysis [Pour-El and Richards, 1989]. It is consistent with the ones used in many other works involving real-valued computation, including Schönhage [1982], Lovász [1986] and Papadimitriou and Tsitsiklis [1986]. It is different from the real RAM model used in, e.g., Traub et al. [1988] and Blum et al. [1989]. See Ko [1991] for more discussions on the models for continuous functions.

3 Polynomial-Time Recognizable Sets

We are interested in characterizing the class of sets $S \subseteq \mathbf{R}^2$ whose membership problems are solvable in polynomial time. Intuitively, the membership problem of a subset S of \mathbf{R}^2 is solvable if there exists a machine M that for each given point $\mathbf{z} \in \mathbf{R}^2$ determines whether \mathbf{z} is in S ; that is, if there is a machine M that computes the characteristic function χ_S of S . In our computational model, the point $\mathbf{z} = \langle x, y \rangle$ is naturally presented to the machine as two oracle functions ϕ and ψ that binary converge to x and y , respectively, and the machine M is an oracle Turing machine. In this setting, we note that the oracle machine M cannot solve the membership problem for set S absolutely correct because it, within a finite number of moves, does not have the ability to distinguish between two close but distinct points in \mathbf{R}^2 (cf. Proposition 2.2). Thus, for a nontrivial theory, we must allow the machine M to make errors, while we require that the errors are under control. In this section, we present four different formulations of polynomial-time solvable subsets of $[0, 1]^2$, and consider their relationship.

In the following, for any set $S \subseteq \mathbf{R}^2$, we let Γ_S be the set of all points \mathbf{z} in \mathbf{R}^2 such that for any $r > 0$, the neighborhood $N(\mathbf{z}; r)$ intersects both S and its complement S^c . For a simply connected region S , Γ_S is its boundary. In the following, an oracle Turing machine M is said to run in *polynomial time* if for all oracles ϕ, ψ and all inputs n , $M^{\phi, \psi}(n)$ halts in $p(n)$ moves for some polynomial p (cf. Definition 2.1(b)).

Definition 3.1 (a) A set $S \subseteq \mathbf{R}^2$ is polynomial-time approximable (or, simply P-approximable) if there exists a polynomial-time oracle Turing machine M such that for any input n , the error set $E_n(M)$ has size $\mu^*(E_n(M)) \leq 2^{-n}$, where $E_n(M)$ is the set of all

$\mathbf{z} \in \mathbf{R}^2$ having a Cauchy function representation (ϕ, ψ) such that $M^{\phi, \psi}(n) \neq \chi_S(\mathbf{z})$. (When the machine M is understood, we write E_n for $E_n(M)$.)

(b) A set $S \subseteq \mathbf{R}^2$ is polynomial-time recognizable (or, simply P-recognizable) if there exists a polynomial-time oracle Turing machine M such that $M^{\phi, \psi}(n) = \chi_S(\mathbf{z})$ whenever (ϕ, ψ) represents a point \mathbf{z} whose distance to Γ_S is $> 2^{-n}$; i.e., $E_n(M) \subseteq \{\mathbf{z} : \delta(\mathbf{z}, \Gamma_S) \leq 2^{-n}\}$.

In the above, we used two different error control mechanisms to define polynomial-time solvable sets. The first, the notion of P-approximability, controls the error size, and is a direct generalization of one-dimensional polynomial-time measurable sets of Ko [1986]. The second, the notion of P-recognizability, controls the errors to occur only close to the boundary.

The above notions of P-approximability and P-recognizability allow two-sided errors to occur in the computation. We can also extend them to have only one-sided errors; that is, machine M must recognize \mathbf{z} if $\mathbf{z} \in S$. The one-sided error requirement is useful when we are concerned with simple sets such as sets of measure 0. (It is clear that a set S of measure 0 is trivially P-approximable and P-recognizable.)

Definition 3.2 A set $S \subseteq \mathbf{R}^2$ is strongly P-approximable (strongly P-recognizable) if S is P-approximable (P-recognizable, respectively) by an oracle machine M such that $M^{\phi, \psi}(n) = 1$ whenever (ϕ, ψ) represents a point x in S (i.e., $E_n(M) \subseteq S^c$).

Strongly P-recognizable sets are a natural characterization of the zeros of polynomial-time computable functions which have a polynomial inverse modulus at zeros. We show it in Section 4.

The relationship between the above four notions of polynomial-time computability is nontrivial. We first consider the relation between P-approximable sets and P-recognizable sets. Recall that a curve is called *rectifiable* if it has a finite length.

Theorem 3.3 (a) There exists a P-recognizable set S that is not P-approximable.

(b) If S is a simply connected, P-recognizable set that has a rectifiable boundary curve, then S is also P-approximable.

Conversely, the question of whether a P-approximable set is always P-recognizable depends on the relation between discrete probabilistic computation and deterministic computation. To make this precise, we need to extend the probabilistic complexity class BPP to more general complexity classes.

Definition 3.4 For any polynomial-time predicate R such that $R(s, t) \Rightarrow \ell(t) = p(\ell(s))$ for some fixed polynomial p , define

$$R_1 = \{s \in \{0, 1\}^* : \text{there exist } \geq (3/4)2^{p(\ell(s))} \\ \text{strings } t \text{ such that } R(s, t)\},$$

$$R_0 = \{s \in \{0, 1\}^* : \text{there exist } \geq (3/4)2^{p(\ell(s))} \\ \text{strings } t \text{ such that } \neg R(s, t)\}.$$

Two sets $A, B \subseteq \{0, 1\}^*$ are called a *BP-pair* if there exists a polynomial-time predicate R such that $A = R_1$ and $B = R_0$.

The above definition of *BP-pair* is a generalization of the complexity class *BPP*; namely, a set A is in *BPP* iff (A, A^c) is a *BP-pair*. Let A and B be two disjoint subsets of $\{0, 1\}^*$. We say A and B are *P-separable* if there exists a set $C \in P$ such that $A \subseteq C$ and $B \subseteq C^c$. It is easy to see that if all *BP-pairs* are *P-separable* then $BPP = P$. It is not clear whether the converse holds. Obviously, if $FP = \#P$ then all *BP-pairs* are *P-separable*, but some weaker sufficient condition is expected.

Theorem 3.5 In the following, (a) \Rightarrow (b) \Rightarrow (c).

- (a) All *BP-pairs* (A, B) are *P-separable*.
- (b) All *P-approximable sets* are *P-recognizable*.
- (c) $BPP = P$.

Other relations between these four classes of polynomial-time solvable sets are shown as follows. (We let *PA* stand for *P-approximable sets*, *PR* for *P-recognizable sets*, and let *S* stand for “strongly.”)

Theorem 3.6 (a) $SPA \stackrel{c}{\neq} PA$.

- (b) $SPR \stackrel{c}{\neq} PR$.
- (c) $SPA \not\subseteq SPR$.
- (d) $SPR \not\subseteq SPA$.

4 Zeros of Polynomial-Time Computable Functions

It is known that the class of recursively closed sets characterizes precisely the sets of the zeros of *P-computable* real functions from $[0, 1]$ to \mathbf{R} [Nerode and Huang, 1985; Ko, 1991]. In this section, we show an analogous result for *P-computable* real functions which have a polynomial inverse modulus of continuity at zeros; that is, the sets of zeros of such functions from $[0, 1]^2$ to \mathbf{R} can be characterized precisely as strongly *P-recognizable*, closed sets. First, let us define the notion of the inverse modulus of continuity.

Definition 4.1 A function $f : [0, 1]^2 \rightarrow \mathbf{R}$ has a polynomial inverse modulus at zeros if there exists a polynomial p such that for each zero \mathbf{z}_0 of f , there is a

constant n_0 such that for all $n \geq n_0$, $|\mathbf{z} - \mathbf{z}_0| > 2^{-n}$ implies $|f(\mathbf{z})| > 2^{-p(n)}$. The function p is called an inverse modulus function at zeros.

The condition that f has a polynomial inverse modulus at zeros means that if we know that $f(\mathbf{z})$ is close to 0 then \mathbf{z} must actually close to some zero \mathbf{z}_0 . Thus, the conventional zero-testing of $|f(\mathbf{z})| \leq \epsilon$ works for the function f . Note that if f is not known to have a polynomial inverse modulus at zeros, then it may have zeros of high complexity [Ko and Friedman, 1982]. Therefore, this condition is necessary if we are interested in zeros of reasonably low complexity.

Theorem 4.2 A closed set $S \subseteq [0, 1]^2$ is strongly *P-recognizable* iff there exists a *P-computable* function $f : [0, 1]^2 \rightarrow \mathbf{R}$ that has a polynomial inverse modulus at zeros such that S is exactly the set of zeros of f .

A particularly interesting class of closed, strongly *P-recognizable* sets is the class of strongly *P-recognizable* singleton sets $\{\mathbf{z}\}$; that is, the class of isolated zeros of *P-computable* functions that have polynomial inverse moduli at zeros. In the following, we show that such isolated roots are *P-computable* if $P = NP$.

Definition 4.3 (a) A real number x is *NP-computable* if there exist a nondeterministic Turing machine M and a polynomial p such that for each input n ,

- 1) at least one computation path of $M(n)$ halts in time $p(n)$, and
- 2) if a computation path of $M(n)$ halts with output $d \in \mathbf{D}$, then $|d - x| \leq 2^{-n}$.

(b) A point $\mathbf{z} = \langle x, y \rangle \in \mathbf{R}^2$ is *NP-computable* if both x and y are *NP-computable* real numbers.

A simple characterization of *NP-computable* real numbers is that they are precisely those real numbers which possess left cuts and right cuts in *NP*, i.e., real numbers which are both left *NP* and right *NP* (see Section 6 for the definition). It is not clear whether such a real number must have a left cut in $NP \cap coNP$, since its left *NP* cuts and right *NP* cuts may not match.

Theorem 4.4 If $\{\mathbf{z}\}$ is strongly *P-recognizable*, then \mathbf{z} is *NP-computable*.

It is not known whether the converse of the above theorem holds. That is, we do not know whether a singleton strongly *P-recognizable* set $\{\mathbf{z}\}$ must have a *P-computable* point \mathbf{z} . From the study of zeros of one-dimensional *P-computable* reals (see Chapter 4 of Ko [1991]), however, we have some partial answer to it. In the following, recall that a tally set is a set over a singleton alphabet $\{0\}$. Also recall that *UP* is the

class of sets accepted by unambiguous nondeterministic machines in polynomial time; that is, for any input w , there is at most one accepting computation for w . The relation between the class UP and the class P is related to the existence of one-way functions. In particular, the condition below that $UP \cap coUP$ contains a tally set not in P is equivalent to the following condition: there exists a one-one, polynomially-honest, polynomial-time computable function $\phi : \{0,1\}^* \rightarrow \{0,1\}^*$ such that $\{0\}^* \subseteq Range(\phi)$ and that ϕ^{-1} is not P -computable on $\{0\}^*$.

Corollary 4.5 *In the following, (a) \Rightarrow (b) \Rightarrow (c).*

(a) *All tally sets in Δ_2^P are in P .*

(b) *Every strongly P -recognizable singleton set $\{\mathbf{z}\}$ must consist of a P -computable point \mathbf{z} .*

(c) *All tally set in $UP \cap coUP$ are in P .*

5 Winding Numbers and The Membership Problem

In this section, we consider bounded, simply connected regions S in \mathbf{R}^2 that have P -computable boundary curves Γ_S . In particular, we study the complexity of the membership problem of such sets S in terms of the notion of P -recognizability introduced in Section 3. We approach this problem by first studying a more general problem of counting the winding numbers. The winding number problem is, informally, the problem of computing the number of times a P -computable closed curve winds around a given point (presumably not on the curve). For a simple, closed curve, the winding number determines whether a point is in the interior or the exterior of the curve. Thus, the upper bound for the winding number problem is also an upper bound for the membership problem with respect to the boundary representation.

The notion of winding numbers can be formally defined as follows: Let $arg(\mathbf{z})$ denote the arguments of $\mathbf{z} \in \mathbf{R}^2$ if $\mathbf{z} \neq \mathbf{0}$; that is, arg is a multi-valued function from $\mathbf{R}^2 - \{\mathbf{0}\}$ to \mathbf{R} such that if $\mathbf{z} = \langle x, y \rangle$ then $x = |\mathbf{z}| \cos(arg(\mathbf{z}))$ and $y = |\mathbf{z}| \sin(arg(\mathbf{z}))$. Let Γ be a closed curve with a representation f ; that is, f is a continuous function from $[0, 1]$ to \mathbf{R}^2 such that $f(0) = f(1)$, and Γ is the range of f . For any point $\mathbf{z}_0 \notin \Gamma$, a *continuous argument function* $h_{\mathbf{z}_0}$ is a continuous function such that $h_{\mathbf{z}_0}(t)$ is a value of $arg(f(t) - \mathbf{z}_0)$. The winding number of \mathbf{z}_0 with respect to Γ is defined to be $(h_{\mathbf{z}_0}(1) - h_{\mathbf{z}_0}(0))/2\pi$ for *any* continuous argument function $h_{\mathbf{z}_0}$.

Note that the winding number of a given point \mathbf{z} with respect to a curve Γ , regarded as a function of \mathbf{z} , has discontinuities on the curve Γ . Thus, similar to polynomial-time computable subsets of \mathbf{R}^2 , any notion of computability for winding numbers must allow

errors to occur. Our computational model for winding numbers is similar to the model for P -recognizable sets: it allows the errors but only when the input point is close to the curve, where discontinuities may occur. More formally, let Γ be a closed curve. We say an oracle Turing machine M *computes the winding number* of Γ , if for all oracles (ϕ, ψ) that represent some \mathbf{z} in \mathbf{R}^2 , and for all inputs n , $M^{\phi, \psi}(n)$ outputs the winding number of \mathbf{z} with respect to Γ whenever $\delta(\mathbf{z}, \Gamma) > 2^{-n}$. We say the winding number of a closed curve Γ is *P -computable* if there exists such an oracle machine that computes the winding number and always halts in $p(n)$ moves on input n , where p is a polynomial.

The complexity of winding numbers is to be characterized by the counting class $\#P$.

Theorem 5.1 (a) *For any continuous closed curve Γ that has a polynomial-time representation f , there exists an oracle machine that computes the winding number of Γ in polynomial time, using a function G in $\#P$ as the oracle.*

(b) *For any function $G \in \#P$, there exist a P -computable function $f : [0, 1] \rightarrow \mathbf{R}^2$ that represents a closed curve Γ , and a P -computable (discrete) function $\phi : \{0, 1\}^* \rightarrow \mathbf{D} \times \mathbf{D}$ such that*

- 1) $\delta(\phi(w), \Gamma) \geq 2^{-p(\ell(w))}$ for some polynomial p , and
- 2) *the winding number of $\phi(w)$ with respect to the curve Γ is equal to $G(w)$.*

Corollary 5.2 *The following are equivalent:*

(a) $FP = \#P$.

(b) *For every P -computable closed curve Γ , the winding number problem with respect to Γ is solvable in polynomial time.*

Sketch of Proof (Theorem 5.1). (a) Let \mathbf{z}_0 be a point not in Γ and $\delta(\mathbf{z}_0, \Gamma) > 2^{-n}$. Assume that $[\alpha, \beta]$ is a subinterval of $[0, 1]$ such that $|f(t_1) - f(t_2)| < \delta(\mathbf{z}_0, \Gamma)$ for all $t_1, t_2 \in [\alpha, \beta]$. Then it is easy to verify that the *argument increase* $h_{\mathbf{z}_0}(\beta) - h_{\mathbf{z}_0}(\alpha)$ is P -computable (as a function of (α, β)). Since f has a polynomial modulus on $[0, 1]$, we may partition $[0, 1]$ into at most $m = 2^{p(n)}$ subintervals $[\alpha_0, \alpha_1], [\alpha_1, \alpha_2], \dots, [\alpha_{m-1}, \alpha_m]$, where p is a polynomial, such that $|f(t_1) - f(t_2)| \leq 2^{-n} < \delta(\mathbf{z}_0, \Gamma)$ as long as t_1 and t_2 belong to the same subinterval. Now, it is clear that the winding number of \mathbf{z}_0 with respect to Γ , or the total argument increase $h_{\mathbf{z}_0}(1) - h_{\mathbf{z}_0}(0)$ divided by 2π , is equal to the sum

$$\sum_{i=0}^{m-1} \frac{1}{2\pi} (h_{\mathbf{z}_0}(\alpha_{i+1}) - h_{\mathbf{z}_0}(\alpha_i)),$$

which is computable in polynomial time relative to a function $G \in \#P$, since the argument increase for each subinterval is polynomial-time computable.

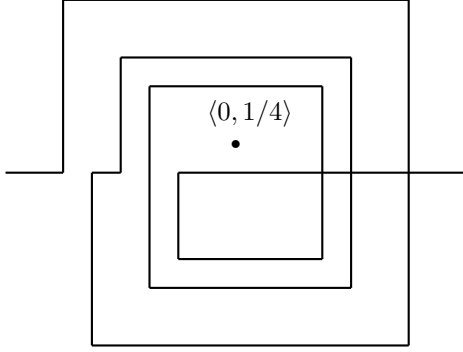


Figure 1: The arc Γ_B for set $B = \{00, 10, 11\}$.

(b) We first describe a basic construction that will be used later. For any integer n and any set $B \subseteq \{0, 1\}^n$, we define an arc Γ_B that is represented by a function $g_B : [0, 1] \rightarrow \mathbf{R}^2$. For each integer k , $0 \leq k \leq 2^n - 1$, we let u_k denote the n -bit binary representation of k .

(1) g_B is linear on $[0, 1/4]$: $g_B(0) = \langle -2, 0 \rangle$ and $g_B(1/4) = \langle -1 - 2 \cdot 2^{-n}, 0 \rangle$.

(2) For each k such that $0 \leq k \leq 2^n - 1$, let $t_k = 1/4 + k \cdot 2^{-(n+1)}$. If $u_k \notin B$, then g_B is linear on $[t_k, t_{k+1}]$: $g_B(t_k) = \langle -1 + (k-2) \cdot 2^{-n}, 0 \rangle$ and $g_B(t_{k+1}) = \langle -1 + (k-1) \cdot 2^{-n}, 0 \rangle$.

(3) For each k such that $0 \leq k \leq 2^n - 1$, if $u_k \in B$, then g_B is piecewise linear on $[t_k, t_{k+1}]$: it divides $[t_k, t_{k+1}]$ into 5 subintervals of equal length and maps them to the 5 consecutive line segments defined by the following points:

$$\begin{aligned} &\langle -1 + (k-2) \cdot 2^{-n}, 0 \rangle, \\ &\langle -1 + (k-2) \cdot 2^{-n}, 1 - (k-2) \cdot 2^{-n} \rangle, \\ &\langle 1 - (k-2) \cdot 2^{-n}, 1 - (k-2) \cdot 2^{-n} \rangle, \\ &\langle 1 - (k-2) \cdot 2^{-n}, -1 + (k-2) \cdot 2^{-n} \rangle, \\ &\langle -1 + (k-1) \cdot 2^{-n}, -1 + (k-2) \cdot 2^{-n} \rangle, \\ &\langle -1 + (k-1) \cdot 2^{-n}, 0 \rangle. \end{aligned}$$

(4) g_B is piecewise linear on $[3/4, 1]$: $g_B(3/4) = \langle -2 \cdot 2^{-n}, 0 \rangle$ and $g_B(1) = \langle 2, 0 \rangle$.

In Figure 1, we show the arc g_B for the set $B = \{00, 10, 11\}$, where $n = 2$. It is easy to see that if we connect the two endpoints of the arc g_B from below to form a closed curve (called the *extended curve* g_B), then the winding number of point $\langle 0, 2^{-n} \rangle$ with respect to this extended curve is equal to the size of B .

Now we describe the function f . For convenience, we will define f on $[0, 2]$ instead of $[0, 1]$. It is clear we can easily transform it to a function on $[0, 1]$ if necessary. Since $G \in \#P$, there exist a set $A \in P$ and a polynomial q such that for all $w \in \{0, 1\}^*$,

$\ell(w) = n$, $G(w) = \|B_w\|$, where $B_w = \{u : \ell(u) = q(\ell(w)), \langle u, w \rangle \in A\}$. For any string w of length n , let i_w be the integer whose n -bit binary representation is w . Let $a_n = 1 - 2^{-(n+1)}$ and $x_w = a_n + i_w \cdot 2^{-2n}$. Note that if u is the lexicographic successor of w , then $x_w + 2^{-2n} = x_u$.

For each $w \in \{0, 1\}^*$, we define the function f on the subinterval $[x_w, x_w + 2^{-2n}]$, where $\ell(w) = n$, to be a linear transformation of g_{B_w} on $[0, 1]$. Let $g_1, g_2 : [0, 1] \rightarrow \mathbf{R}$ be such that $g_{B_w}(t) = \langle g_1(t), g_2(t) \rangle$. Then, f on $[x_w, x_w + 2^{-2n}]$ can be defined as follows:

$$\begin{aligned} f(t) = &\langle 2^{-(2n+2)} \cdot g_1(2^{2n}(t - x_w)) + x_w + 2^{-(2n+1)}, \\ &2^{-(2n+2)} \cdot g_2(2^{2n}(t - x_w)) \rangle. \end{aligned}$$

The above defined f on $[0, 1]$. Now, we define f on $[1, 2]$ to be piecewise linear mapping the interval $[1, 2]$ to three line segments connecting the following 4 points: $\langle 1, 0 \rangle$, $\langle 1, -1 \rangle$, $\langle 0, -1 \rangle$, and $\langle 0, 0 \rangle$.

It is clear that f is continuous on $[0, 1]$ and it defines a closed curve Γ . It is also easy to see that if we define $\phi(w) = \langle x_w + 2^{-(2n+q(n)+2)}, 2^{-(2n+q(n)+2)} \rangle$, where $n = \ell(w)$, then ϕ is P-computable and $\delta(\phi(w), \Gamma) = 2^{-(2n+q(n)+2)}$. Furthermore, the winding number of $\phi(w)$ with respect to Γ is equal to that of $\langle 0, 2^{-q(n)} \rangle$ with respect to the extended curve defined by g_{B_w} , which is equal to $G(w)$.

Finally, we show that f is P-computable. This follows from the following two observations: First, the functions g_{B_w} are uniformly P-computable, and f is a linear transformation of these functions. Second, f has a polynomial modulus of continuity on $[0, 1]$ because of the parameter $2^{-(2n+2)}$ in the linear transformation definition for f . We omit the details. \square

Next, for the membership problem for simply connected regions, we apply the proof of Theorem 5.1(a) to give a slightly tighter upper bound than $\#P$. That is, we need only one bit from a function in $\#P$ to help us to determine the membership of a given point. This observation is closely related to the recently studied complexity class *MidBitP* [Regan, Schwentick, 1991; Green et al. 1992].

Corollary 5.3 *Let $f : [0, 1] \rightarrow [0, 1]^2$ be a P-computable function defining a simple closed curve Γ . Then, the interior S of the curve is P-recognizable with respect to an oracle $G \in \#P$. In addition, the oracle machine M that P-recognizes S needs only to ask the oracle for one bit of a value of G .*

For the lower bound of the membership problem, we can only prove a weaker bound in terms of the complexity class *UP*.

Theorem 5.4 *In the following, (a) \Rightarrow (b) \Rightarrow (c).*
(a) $FP = \#P$.

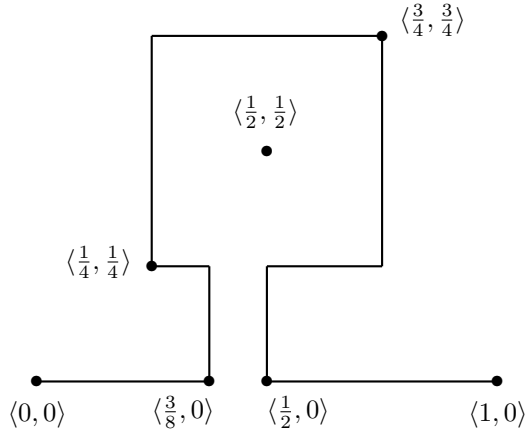


Figure 2: The function g_B for set $B = \{011\}$.

(b) Every simply connected region S with a P -computable boundary is P -recognizable.

(c) $P = UP$.

Sketch of Proof. (a) \Rightarrow (b) is an immediate corollary of Theorem 5.1. We give a sketch of the proof for (b) \Rightarrow (c), which follows the same idea as Theorem 5.1(b). First, we describe a basic function $g_B : [0, 1] \rightarrow \mathbf{R}^2$ for each $B \subseteq \{0, 1\}^n$ that is either a singleton or empty. First, if B is empty, then we let g_B linearly maps $[0, 1]$ to the line segment from $\langle 0, 0 \rangle$ to $\langle 1, 0 \rangle$. Next, if B is a singleton of a single element u , then we divide $[0, 1]$ into 2^n subintervals, each corresponding to a string in $\{0, 1\}^n$, and let g_B maps all subintervals not corresponding to u like g_B for the empty set B , and let g_B consist of 7 line segments that enclose the point $\langle 1/2, 1/2 \rangle$ below the curve of g_B . For instance, if $n = 3$ and $B = \{011\}$, then the curve defined by g_B is shown in Figure 2.

Let $A \in UP$ and $B \in P$ such that for each $w \in \{0, 1\}^*$,

$$\begin{aligned} w \in A &\iff (\exists u, \ell(u) = p(\ell(w))) \langle w, u \rangle \in B \\ &\iff (\exists \text{ unique } u, \ell(u) = p(\ell(w))) \langle w, u \rangle \in B, \end{aligned}$$

where p is a polynomial. For each w , we let $B_w = \{u : \ell(u) = p(\ell(w)), \langle w, u \rangle \in B\}$. Then, B_w is either a singleton or empty.

As in Theorem 5.1, we define the function f on each interval $[x_w, x_w + 2^{-2n}]$ as a linear transformation of g_{B_w} : for each $t \in [x_w, x_w + 2^{-2n}]$, $\ell(w) = n$,

$$\begin{aligned} f(t) &= \langle 2^{-2n} \cdot g_1(2^{2n}(t - x_w)) + x_w, \\ &\quad 2^{-2n} \cdot g_2(2^{2n}(t - x_w)) \rangle, \end{aligned}$$

where $\langle g_1(t), g_2(t) \rangle = g_{B_w}(t)$. Then, we extend f to connect the endpoints $\langle 1, 0 \rangle$ to $\langle 0, 0 \rangle$ by a simple curve that does not intersect the images of g_{B_w} .

It is not hard to prove that f is P -computable. In addition, if we let, for each w , $\phi(w)$ be the image of $\langle 1/2, 1/2 \rangle$ under the linear transformation used to define f , then we can see from the transformation that $\delta(\phi(w), \Gamma) \geq 2^{-2n}$. Also, $w \in A$ iff $\phi(w)$ is in the interior of the curve Γ . This completes the proof. \square

6 The Distance Between a Point and a Curve

Computing the distance between a point \mathbf{z} and a curve Γ is one of the fundamental problems in computational complex analysis. In addition, many computational tasks work only when a point \mathbf{z} is bounded away from the curve Γ ; e.g., the winding number and the membership problem we discussed above. What is the complexity of this problem? Or, if Γ is P -computable, does it follow that the function $\text{dist}_\Gamma(\mathbf{z}) = \delta(\mathbf{z}, \Gamma)$ is also P -computable? We observe that this problem is close to the minimization problem [Ko, 1982; Friedman, 1984], and it turns out they do have the same complexity bounds.

Theorem 6.1 *The following are equivalent:*

(a) $P = NP$.

(b) For any P -computable curve Γ , the function dist_Γ is also P -computable.

Next, we consider the problem of determining the distance value between a fixed point and a P -computable curve. That is, if Γ is a P -computable curve and \mathbf{z}_0 is a fixed point, is the value $\delta(\mathbf{z}_0, \Gamma)$ always a P -computable real number?

For the maximization problem, it is known that the set of maximum values for P -computable functions from $[0, 1]$ to \mathbf{R} is exactly the set of *left NP real numbers* [Ko, 1982], where a real number x is left NP if there exists a Cauchy function ϕ binary converging to x such that the set $L_\phi = \{d \in \mathbf{D}_n : d \leq \phi(n)\}$ is in NP .⁵ This notion of left NP real numbers can also be applied to characterize the complexity of distance values. We say a real number x is *right NP* if $-x$ is left NP. In the following, we let $\mathbf{0}$ denote the origin $\mathbf{0} = \langle 0, 0 \rangle$.

Theorem 6.2 *A real number x is right NP iff there exists a P -computable curve Γ such that $x = \delta(\mathbf{0}, \Gamma)$.*

The complexity of left NP real numbers has been studied in Ko [1991]. This study yields the following bounds for the distance values. Recall that a tally set is a set of strings over a singleton alphabet $\{0\}$.

⁵Note that for the same dyadic real number d , it is possible that when d is represented by a string of precision n then it is in L_ϕ , while when it is represented by a string of precision $n + 1$, it is not in L_ϕ . Thus, L_ϕ is really a set of the string representations of dyadic numbers, rather than a subset of \mathbf{D} .

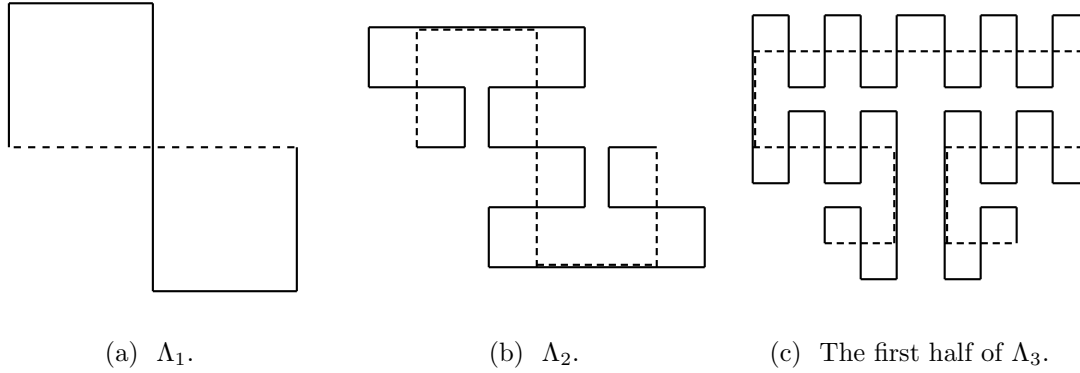


Figure 3: The functions Λ_n , $n = 1, 2, 3$, where the dot lines indicate the function Λ_{n-1} .

Corollary 6.3 *In the following, (a) \Rightarrow (b) \Rightarrow (c).*

(a) *All tally sets in Δ_2^P are in P .*

(b) *For every P -computable function $f : [0, 1] \rightarrow \mathbf{R}^2$ that represents a curve Γ , the distance $\delta(\mathbf{0}, \Gamma)$ is a P -computable real number.*

(c) *All tally sets in NP are in P .*

It is well known that condition (c) above is equivalent to the condition that the class $NEXP$ of nondeterministic exponential-time ($2^{n^{O(1)}}$) computable sets collapses to the class EXP of deterministic exponential-time computable sets.

7 The Area of a Region

In this section, we consider the problem of computing the area of a region, given either a P -approximator M or a boundary Γ as the representation. Assume that a simply connected region S is P -approximable; then the area can be computed by the straightforward sampling method, which can be done in polynomial time relative to an oracle in $\#P$. Indeed, Ko [1986] has shown that the computation of the measure of a P -approximable one-dimensional set $S \subseteq [0, 1]$ can be done in polynomial time iff $FP_1 = \#P_1$, where FP_1 (and $\#P_1$) denotes the class of functions ϕ in FP (and $\#P$, respectively) whose inputs are strings over a singleton alphabet $\{0\}$. A similar proof applies to two-dimensional P -approximable sets.

Theorem 7.1 *The following are equivalent:*

(a) $FP_1 = \#P_1$.

(b) *For any P -approximable set $S \subseteq [0, 1]^2$, the measure of set S is P -computable.*

Corollary 7.2 *The following are equivalent:*

(a) $FP_1 = \#P_1$.

(b) *For any simply connected region S that has a P -computable, rectifiable boundary, the measure of S is P -computable.*

It is interesting to see that the above rectifiability condition on the boundary of S is necessary. Indeed, the following result shows that the measure of a region with a polynomial-time computable curve which is not rectifiable is not even necessarily computable. Thus, the rectifiability, in addition to the P -computability, of the boundary curve is a critical condition on a region.

Theorem 7.3 *There exists a polynomial-time computable, simple closed curve Γ whose interior S is a simply connected region with a nonrecursive measure.*

Sketch of Proof. The idea of the proof is as follows: Let Γ' be the boundary of the square $[0, 1] \times [-1, 0]$, in the clockwise orientation. The curve Γ is to be obtained from Γ' by substituting a simple curve Γ_n for a line segment $L_n \subset [0, 1] \times \{0\}$ in Γ' , where L_n is of length $O(2^{-n})$ and has a distance $O(2^{-n})$ to line segments L_{n-1} and L_{n+1} . Pick a recursively enumerable but nonrecursive set K . For each $m \notin K$, the curve Γ_m is of infinite length but is symmetric with respect to the line $y = 0$. Thus, the substitution of Γ_m for a line segment of Γ' does not change the measure of the interior of Γ' . For each $n \in K$, the curve Γ_n is of finite length but it contributes to the interior of Γ an extra area of size $(4/5) \cdot 2^{-2n}$. Thus, the measure of the interior of Γ becomes

$$1 + \frac{4}{5} \cdot \sum_{n \in K} 2^{-2n},$$

which is easy to see to be a nonrecursive real number.

The main difficulty of the construction is to make curves Γ_m for $m \notin K$ and curves Γ_n for $n \in K$ to be of the similar shape and yet of a significant difference in

the measures “below” the curves. (They must be of the similar shape, for otherwise a recursive procedure may be devised to decide whether $n \in K$ by observing the difference between these two types of curves.) To solve this problem, we construct a sequence $\{\Lambda_n\}$ of curves which have the following properties:

- 1) The curves $\{\Lambda_n\}$ are uniformly P-computable,
- 2) The curves $\{\Lambda_n\}$ converges to a limit curve Λ in a polynomial speed; that is, if functions g_n compute Λ_n and function g computes Λ , then $|g_{p(n)}(t) - g(t)| \leq 2^{-n}$ for some polynomial p .
- 3) The curve Λ_n consists of $\Theta(5^n)$ line segments, each of length $\Theta(5^{-n/2})$. (This implies that the lengths of the curves Λ_n grow in an exponential rate.)
- 4) The curves Λ_n are symmetric with respect to the line $y = 0$.

The formal definition for curves Λ_n is too complicated to be included here. The reader can get a rough idea of the structure of these curves from Figure 3. (The resulting limit curve Λ is similar to the curve called the Lebesgue-Osgood monster in Mandelbrot [1983].) From this figure and condition (3) above, we observe that each curve Λ_n has $\Theta(5^n)$ many open rectangles of the shape \square , \sqcup , \square or \sqsupset , each of size $\Theta(5^{-n})$.

Next, for each n , we modify Λ_n into a curve Λ'_n by eliminating half of the open rectangles of the shape \square , \sqcup , \square or \sqsupset in Λ_n so that the curve Λ'_n is no longer symmetric to the line $y = 0$. Rather, the area “below” the curve Λ'_n is greater than the area “below” the curve Λ_n by the size $4/5$. (Note that this size difference is a constant, independent of n . This can be done because Λ_n has $\Theta(5^n)$ many open rectangles and each is of size $\Theta(5^{-n})$.)

Now, we can complete the construction of the curve Γ . Let M be a Turing machine accepting the set K . If $m \notin K$, then we let Γ_m be the image of the curve Λ under a linear transformation that shrinks Λ by a factor of 2^{-m} . If $n \in K$, assume that M accepts n in $T(n)$ moves. Then, let Γ_n be the image of $\Lambda'_{T(n)}$ under a linear transformation that shrinks $\Lambda'_{T(n)}$ by a factor of 2^{-n} . It is ready to verify that these curves Γ_n indeed satisfy our needs. In particular, we can prove that the curve Γ is P-computable: For each integer n and each error bound 2^{-k} , we simulate machine M on input n for k moves. If M does not accept n within k moves, then we simply assume that $n \notin K$ and outputs an approximation of the linear transformation of the curve Λ ; otherwise, if M accepts n in k moves, then $k \geq T(n)$, and we can output the linear transformation of the curve $\Lambda'_{T(n)}$ within $q(k)$ moves for some polynomial q . In the case $n \in K$ but $T(n) > k$, we observe that $\Lambda'_{T(n)}$ and Λ differ by an error $\leq 2^{-T(n)}$, and so the

output has a small error. \square

Remark. The above set S can be proved to be P-recognizable. On the other hand, it follows from Theorem 7.1 that it is not P-approximable. Indeed, it is not even recursively approximable [Ko, 1991]. Thus this set S is an example for Theorem 3.3(a).

8 The Length of a Curve

In this section we consider the question of measuring the length of a polynomial-time computable, one-to-one curve. The problem of measuring the length of a curve is called *the coastline problem* in Mandelbrot [1983]. Since the length of the curve may actually be infinite, the problem is considered as a difficult problem. In the following, we show that even when the curve has a finite length, the length may still be a nonrecursive real number. Thus, it justifies, in our computational model, that the coastline problem is indeed difficult even if the coastline is assumed to be of finite length.

Theorem 8.1 *There exists a polynomial-time computable, one-to-one function $f : [0, 1] \rightarrow \mathbf{R}^2$ that defines a rectifiable curve Γ such that the length of Γ is a nonrecursive real number.*

Sketch of Proof. The construction of the curve Γ is based on the same idea as in Theorem 7.3. Here, we need an additional property of the curves Λ_n constructed in Theorem 7.3: the lengths $\text{Leng}(\Lambda_n)$ of Λ_n are uniformly P-computable in the sense that an approximation value for $\text{Leng}(\Lambda_n)$ with error $\leq 2^{-k}$ can be computed in time $p(n+k)$ for some polynomial p . We let $l_n = \text{Leng}(\Lambda_n)$. Note that $l_n = \Theta(5^{n/2})$.

Again, let K be a recursively enumerable but non-recursive set. Let $T(n)$ be a function such that a Turing machine M accepts $n \in K$ in $T(n)$ moves ($T(n) = \infty$ if $n \notin K$). First, we have a basic line segment Γ' from $\langle 0, 0 \rangle$ to $\langle 1, 0 \rangle$. Then, for each $n \in K$, we substitute a curve Γ_n for a line segment L_n in Γ . Each L_n is of length λ_n and has a distance $O(2^{-n})$ to L_{n-1} and L_{n+1} , where $\lambda_n = 2^{-2n} \cdot (l_{T(n)} - 1)^{-1}$. The curve Γ_n is the image of the curve $\Lambda_{T(n)}$ under a linear transformation that shrinks the curve $\Lambda_{T(n)}$ by the factor of λ_n . Thus, the line segment L_n of length λ_n in Γ' is replaced by the curve Γ_n which is of length $\lambda_n \cdot l_{T(n)}$. As far as the length of the curve Γ is concerned, the net effect of this particular substitution is to increase the length of Γ' by $\lambda_n \cdot l_{T(n)} - \lambda_n = 2^{-2n}$. Thus, the total length of Γ is $1 + \sum_{n \in K} 2^{-2n}$, which is nonrecursive.

To see that the curve Γ is P-computable, we consider the following algorithm: for each integer n and error bound 2^{-k} , if the Turing machine M does not accept n in k moves, then we output the line segment

L_n . Otherwise, we know that $k \geq T(n)$, and we can compute $\Lambda_{T(n)}$ and $l_{T(n)}$, and hence the curve Γ_n , in time polynomial in k . When $n \in K$ but $T(n) > k$, we notice that the curve Γ_n and the line segment L_n differ by at most $\lambda_n = O(5^{-T(n)/2})$, and so the error of our approximation is bounded by 2^{-k} . \square

References

- Blum, L., Shub, M. and Smale, S. [1989], On a theory of computation and complexity over the real numbers; NP completeness, recursive functions and universal machines, *Bullet. Amer. Math. Soc. (new series)* **21**, 1–46.
- Friedman, H. [1984], On the computational complexity of maximization and integration, *Advances in Math.* **53**, 80–98.
- Green, F., Köbler, J. and Torán, J. [1992], The power of the middle bit, *Proceedings, 7th Structure in Complexity Theory Conference*, 111–117.
- Ko, K. [1982], The maximum value problem and NP real numbers, *J. Comput. System Sci.* **24**, 15–31.
- Ko, K. [1986], Approximation to measurable functions and its relation to probabilistic computation, *Annals of Pure and Applied Logic* **30**, 173–200.
- Ko, K. [1989], Computational complexity of roots of real functions, *Proceedings, 30th IEEE Symposium on Foundations of Computer Science*, 204–209.
- Ko, K. [1991], *Complexity Theory of Real Functions*, Birkhäuser, Boston.
- Ko, K. and Friedman, H. [1982], Computational complexity of real functions, *Theoret. Comput. Sci.* **20**, 323–352.
- Lovász, L. [1986], *An Algorithmic Theory of Numbers, Graphs and Convexity*, SIAM, Philadelphia.
- Mandelbrot, B.B. [1983], *The Fractal Geometry of Nature*, W.H. Freeman, New York.
- Nerode, A. and Hwang, W. [1985], Applications of pure recursion theory in recursive analysis, *Acta Mathematica Sinica (in Chinese)* **28**, 625–636.
- Papadimitriou, C.H. and Tsitsiklis, J. [1986], Intractable problems in control theory, *SIAM J. Contr. Optim.*, **24**, 639–654.
- Pour-El, M. and Richards, I. [1989], *Computability in Analysis and Physics*, Springer-Verlag, Berlin.
- Regan, K. and Schwentick, T. [1991], On the power of one bit of a $\#P$ function, preprint.
- Schönhage, A. [1982], The fundamental theorem of algebra in terms of computational complexity, preprint, Mathematisches Institut der Universität Tübingen, West Germany.
- Traub, J. F., Wasilkowski, G. W. and Woźniakowski, H. [1988] *Information-Based Complexity*, Academic Press, New York.