Some Complexity Issues in Complex Analysis

(*Extended Abstract for CCA 96*)

Arthur W. Chou*

June 1, 1996

1 Introduction and Summary

We propose a general framework to study the complexity of several fundamental problems in computational complex analysis, based on the bit-operation (Turing) model used in recursive analysis and the complexity theory of real functions of Ko & Friedman [1982] and Ko [1991]. Some simple results were direct consequences of their counterparts in real analysis. For examples, It follows from the theory of analytic functions of a real variable (Ko [1991]) that a polynomial-time computable analytic function of a complex variable has a polynomial-time computable power series expansion, and that its derivative and contour integral are polynomial-time computable locally (that is, within the radius of convergence).

But the fundamental issues, such as the representation of two dimensional sets on the complex plane and the precise notion of polynomial-time "computable" regions remained elusive. In this work we clarify these issues and answer some related questions. It is a summary of a forthcoming paper Chou & Ko [1995] and other unpublished results. Our main goals are, first, to understand the complexity of some complex sets such as fractals (see Ko [1995a] [1995b]); and secondly, to analyze the complexity of conformal mappings and obtain tight complexity bounds for the Riemann Mapping Theorem. So far these goals have only been partially achieved.

We start with formulating the concept of polynomial-time representation of bounded subsets of the plane $\mathbf{R}^2$. 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 Turing 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 admitting errors only on a set $E \subseteq \mathbf{R}^2$ of measure less than 2^{-n}. A subset $S \subseteq \mathbf{R}^2$ is *polynomial-time recognizable* if there is a Turing 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 distance 2^{-n} from the *boundary* of S. We also explore the relation between these two notions. It turns out that it has a close connection to the well-known open question that whether the polynomial-time probabilistic computation (on integers) is strictly stronger than the polynomial-time deterministic computation (on integers).

For conformal mappings in complex analysis, the basic object of study is a bounded, simply connected region; that is, a bounded, connected open set with no hole (or, equivalently, whose

*Department of Mathematics and Computer Science, Clark University, Worcester, MA 01610; email: achou@vax.clarku.edu.

complement is connected). For such sets, the boundary curve is a natural representation. But what are the relationship between this boundary representation and the above ones? This is the critical *membership problem*: must a simply connected region with a polynomial-time computable boundary be polynomial-time recognizable? This question is closely related to the *winding number problem* of computing the winding number of a curve around a point. Here we can 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 can be constructed in such a way that its winding number problem encodes a discrete $\#P$-complete problem.

For the membership problem for simply connected regions with polynomial-time computable boundary It follows immediately that it is also solvable in polynomial time using a function oracle $f \in \#P$. Moreover, we show that a simply connected region is not necessarily polynomial-time recognizable, unless the weak one-way functions do not exist (more precisely, $P = UP$). It remains an open question whether the gap between the upper bound $P^{\#P}$ and the lower bound UP can be narrowed.

In addition to the membership problem, we also investigate other important issues 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 task is required in many computations in complex analysis. It 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 $P = NP$.

We then investigate the complexity of the *Riemann Mapping theorem* (the uniformization theorem): every bounded, simply connected region is conformally equivalent to the unit disk. Let us call the function that realizes this conformal equivalence the *Riemann mapping* (which is unique under some normalizing conditions). We define the notion of "computable" functions from a region to the complex plane $\mathbf{R}^2$, using oracle Turing machines making bounded errors (similar to the recursively approximable functions in Ko [1991]), and show that, given a bounded simply connected region with a polynomial-time computable boundary, the Riemann mapping is computable in exponential space; that is, with a space bound of $O(2^{p(n)})$ where $p(n)$ is a polynomial in "n" and "n" is the output precision of the function values. Whether this upper bound can be lowered and the question on the lower bound remain open.

Remarks on notation and convention.

We write $\ell(w)$ to denote the length of a character string w, and use $\langle w_1, w_2\rangle$ to denote the pairing function on w_1 and w_2.

The set of dyadic rationals is denoted by $\mathbf{D} = \{m/2^n : m \in \mathbf{Z}, n \in \mathbf{N}\}$. We let $\mathbf{D}_n$ denote the class of dyadic rationals with at most n bits in the fractional part of its binary representation.

For real numbers x and y, we write $\langle x, y\rangle$ to signify a point in the $\mathbf{R}^2$ plane. For a real or complex number x, $|x|$ denotes its absolute value. 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. The complement of a set S is denoted by S^c, and the cardinality of a finite set A is written as $\|A\|$.

In this paper, all subsets of $\mathbf{R}^2$ we will consider are *bounded* subsets of $\mathbf{R}^2$. A subset S of plane $\mathbf{R}^2$ is called a *region* if it is nonempty, open and connected. A region is *simply connected* if it does not have "holes," or, equivalently, its complement is connected.

2 Polynomial-Time Recognizable Sets

For the definitions and basic facts about the computational model for real functions, we refer to Ko [1991].

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 completely correctly because, within a finite number of moves, it does not have the ability to distinguish between two close but distinct points in $\mathbf{R}^2$. Thus, for a nontrivial theory, we must allow the machine M to make errors and require that the errors are under control. In this section, we present two 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 usual *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.

Definition 2.1 *(a) A set $S \subseteq \mathbf{R}^2$ is* polynomial-time approximable *(or, simply* P-approximable*) if there exist an oracle Turing machine M and a polynomial p such that for any oracles (ϕ, ψ) representing a point $\mathbf{z} = \langle x, y \rangle \in \mathbf{R}^2$ (i.e., ϕ and ψ binary converge to x and y, respectively), and for any input n, M outputs either 0 or 1 in $p(n)$ moves such that the following set $E_n(M)$ has size $\mu^*(E_n(M)) \le 2^{-n}$:*

$$E_n(M) = \{\mathbf{z} \in \mathbf{R}^2 : \textit{there exists } (\phi, \psi) \textit{ representing } \mathbf{z} \textit{ such that } M^{\phi,\psi}(n) \neq \chi_S(\mathbf{z})\}.$$

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

In other words, a set S is P-approximable if we can approximately determine the membership of S with the runtime polynomially dependent on the error size. Although the notion of P-approximable set appears interesting from the measure-theoretic point of view, it does not provide strong control over where the errors may occur. That is, two sets S_1 and S_2 are essentially regarded as the same set here if they differ by at most a set of measure 0. While a set of measure 0 is negligible from the measure-theoretic point of view, it could be an important factor in other computational problems, for instance, the problem of determining the complexity of isolated zeros of a function.

On the other hand, the notion of P-recognizability controls the errors by requiring that errors can only occur close to the boundary and that the runtime of the machine M be polynomially dependent on the distance between the given point and the boundary. The following example shows that this approach is natural for simple sets. We say a rectangle $[a, b] \times [c, d]$ is *degenerate* if $a = b$ or $c = d$.

Example 2.2 *A non-degenerate rectangle $[a, b] \times [c, d]$ is P-recognizable iff all real numbers a, b, c, d are polynomial-time computable.*

The exact relation between P-approximable sets and P-recognizable sets is not a simple matter. For instance, the above example does not appear to hold for P-approximable sets. We first consider the question of whether P-recognizable sets must be P-approximable. Intuitively, if a simply connected region S with the boundary curve Γ_S is P-recognizable and if Γ_S is *rectifiable* (i.e., Γ_S has a finite length), then the error area of a P-recognizer M for S must be small, and therefore S is P-approximable. On the other hand, if Γ_S is not rectifiable, then the error area of M could be very large. Ko [1995a] has constructed a P-recognizable, simply connected region S with a simple closed boundary such that S has a non-recursive measure. Because the measure of a recursively approximable set is a recursive real number, this shows that the region is not even a recursively approximable set.

We summarize these observations in the following theorem.

Theorem 2.3 *(a) If S is a simply connected, P-recognizable region with a rectifiable boundary, then S is also P-approximable.*

(b) There exists a simply connected, P-recognizable region S which is not P-approximable.

In the other direction, the question of whether a P-approximable set is 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 classes.

Definition 2.4 *A relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is* polynomial-length related *if there exists a polynomial p such that $R(s, t)$ implies $\ell(t) = p(\ell(s))$. Two sets $A, B \subseteq \{0,1\}^*$ are called a BP-*pair *if there exists a polynomial-time predicate R that is polynomial-length related with respect to polynomial p such that*

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

The string t can be viewed as a "witness" or "certificate" for the string s to satisfy the predicate R. A string s is in A if and only if there is greater than or equal to 3/4 probability to randomly choose a witness t to satisfy the polynomial-time predicate R. Similarly for the set B.

The above definition of BP-pairs is the 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 $B \subseteq C^c$.

In the following we are going to show that if all BP-pairs are P-separable then all P-approximable sets are also P-recognizable. Conversely, if $BPP \neq P$, then P-approximable sets are not necessarily P-recognizable. It is easy to see that the condition of all BP-pairs being P-separable implies that $BPP = P$. It is not clear whether the converse holds. The best we know is that if $FP = \#P$ then all BP-pairs are P-separable.

Theorem 2.5 *In the following, (a) ⇒ (b) ⇒ (c).*

(a) All BP-pairs (A, B) are P-separable.

(b) All P-approximable sets are P-recognizable.

(c) $BPP = P$.

Sketch of Proof.

(a) ⇒ (b): Let S be a P-approximable set and M_1 be an oracle machine that P-approximates S, with runtime $p(n)$ and error set $E_n(M_1)$ for input "n". Our goal is to construct a polynomial-time oracle Turing machine M such that it only makes errors close to the boundary of S.

Assume that $\mathbf{z}$ has rational coordinates and that the distance of $\mathbf{z}$ to the boundary Γ_S is greater than 2^{-n}. The key observation is this. Within the neighborhood of small radius $2^{-(n+1)}$ around the point $\mathbf{z}$ and among the dyadic rational points $\mathbf{e}$ with long enough, say, $p(2n+4)$ bits in the fractional part (the grid points in the neighborhood), because of the fact that the size of the error set of M_1 is exponentially small, we can find more than (3/4)th of them to be "good points" on which M_1 produces the correct answer. Therefore, these good points are the witnesses to $\mathbf{z}$ for the polynomial-time predicate $M_1^{\mathbf{e}}(2n+4) = 1$. This way the set of such $\mathbf{z}$'s inside of S together with the set of such $\mathbf{z}$'s outside of of S form a *BP*-pair. (Notice that none of these sets is necessarily in *BPP* because we don't know what happen to those $\mathbf{z}$'s that are close to the boundary.)

Now if *BP*-pairs are P-separable, we can correctly decide whether these points away from the boundary are in S or not in polynomial time; hence the set S is polynomial-time recognizable.

(b) ⇒ (c):

Assume that $BPP \neq P$ and let $A \subseteq \{0,1\}^*$ be a set in $BPP - P$. We will construct a set $S \subseteq [0,1]^2$ that is P-approximable but not P-recognizable. The idea is to embed this (discrete) set A into a subset $S \subseteq [0,1]^2$ in such a way that (1) S is P-approximable and (2) if S is P-recognizable, then A can be show to be in P.

We can embed all strings in $\{0,1\}^*$ into the interval $[0,1]$ on the x-axis in an obvious way: starting from the origin from left to right, for $n = 1, 2, 3, ...$, we successively embed all strings of length n in the lexicographic order into the next subinterval of length 2^{-n} as (the leftmost points of) segments of length 2^{-2n}. More precisely, for each string w of length n, let i_w be the integer less than 2^n whose n-bit binary expansion (including leading zeros) is w. Then, let $a_n = 1 - 2^{-(n-1)}$, and $x_w = a_n + i_w \cdot 2^{-2n}$, which is the representative of the string w on the x-axis. We note that the interval $[0,1)$ is partitioned into $\{[x_w, x_w + 2^{-2\ell(w)}) : \ell(w) \geq 1\}$.

We let $S_w = [x_w, x_w + 2^{-2n}] \times [0,1]$, and define $S = \bigcup_{w \in A, \ell(w) \geq 1} S_w$. This set S is the embedding of the *BPP* set A into $[0,1]^2$. We claim that S is the desired set.

First, to see that S is not P-recognizable, we note that $\mathbf{z}_w = \langle x_w + 2^{-(2n+1)}, 1/2\rangle$ has a distance $2^{-(2n+1)}$ away from Γ_S. Thus, if S were P-recognizable, then we could correctly determine whether $\mathbf{z}_w \in S$ in time $p(n)$ for some polynomial p (by simulating the machine that P-recognizes S with the standard oracles of $\mathbf{z}_w$ and with the input $2n+2$). This in turn would allow us to determine whether $w \in A$ in time $p(n)$, contradicting to the assumption that $A \notin P$.

It is not hard to show that this S is P-approximable. We refer to Chou & Ko [1995] for details. □

3 Winding Numbers and The Membership Problem

In this section, we consider bounded, simply connected regions S in $\mathbf{R}^2$ represented by its polynomial-time computable boundary 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 2. This problem is closely related to the computation of 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 the winding number can be formally defined as follows: Let $arg(\mathbf{z})$ denote the arguments of $\mathbf{z} \in \mathbf{R}^2$ if $\mathbf{z} \neq \langle 0, 0\rangle$; that is, arg is a multi-valued function from $\mathbf{R}^2 - \{\langle 0,0\rangle\}$ 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 $arg(f(t) - \mathbf{z}_0)$. It is easy to see that any two continuous argument functions differ by a multiple of 2π. The winding number of Γ with respect to $\mathbf{z}_0$ is defined to be

$$wind_\Gamma(\mathbf{z}_0) = \frac{1}{2\pi}\Big(h_{\mathbf{z}_0}(1) - h_{\mathbf{z}_0}(0)\Big)$$

for *any* continuous argument function $h_{\mathbf{z}_0}$. Equivalently, we may also define the winding number in the form of integrals over the curve Γ:

$$wind_\Gamma(\mathbf{z}_0) = \frac{1}{2\pi i}\int_\Gamma \frac{1}{\mathbf{z} - \mathbf{z}_0}\, d\mathbf{z}.$$

Each of these definitions provides a natural method to compute the winding number.

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 3.1 *(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 $f : [0, 1] \to \mathbf{R}^2$ that computes a closed curve Γ, a P-computable (discrete) function $\phi : \{0, 1\}^* \to \mathbf{D}\times\mathbf{D}$ and a polynomial p such that

i) $\delta(\phi(w),\Gamma) \geq 2^{-p(\ell(w))}$ *for all* $w \in \{0,1\}^*$*, and*

ii) the winding number of the curve Γ *around the point* $\phi(w)$ *is equal to* $G(w)$*.*

That is, any #P problem can be reduced be a winding number problem.

Corollary 3.2 *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 3.1). (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, $p(n)$, on $[0,1]$, we can partition $[0,1]$ into $m = 2^{p(n)}$ subintervals $[\alpha_0,\alpha_1]$, $[\alpha_1,\alpha_2]$, …, $[\alpha_{m-1},\alpha_m]$ 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

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

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

(b) We use the following well-known characterization of $\#P$ (see, for instance, Ko [1991]): If $G \in \#P$, then 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 $B_w = \{u : \ell(u) = q(\ell(w)), \langle w,u\rangle \in A\}$. We call the u such that $\langle w,u\rangle \in A$ the witness of w.

We can represent strings w on the interval $[0,1]$ the same way as in the second part of the proof of Theorem 2.5. That is, 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 $x_w + 2^{-2n} = x_u$, and that $[0,1) = \bigcup_{w,\ell(w)\geq 1}[x_w, x_w + 2^{-2\ell(w)})$

It suffices to construct a closed curve Γ defined on this interval $[0,1]$ and a sequence of points $\phi(w)$ on the plane, one for each w, such that

(1) Γ and the sequence $\{\phi(w)\}$ are polynomial-time computable,

(2) $\delta(\phi(w),\Gamma) \geq 2^{-p(\ell(w))}$ for all $w \in \{0,1\}^*$, and

(3) On the interval representing w, $[x_w, x_w + 2^{-2\ell(w)}]$, the curve Γ winds around the point $\phi(w)$ the number of times equal to $G(w)$ and encloses no other points in $\{\phi(w)\}$.

To achieve (3), we further divide the interval $[x_w, x_w + 2^{-2\ell(w)}]$ into $2^{q(\ell(w))}$ subintervals to represent the possible witnesses u ($\ell(u) = q(\ell(w))$). If u is a witness, then we make Γ wind ground $\phi(w)$ once (in the correct orientation) throughout the subinterval representing u. Otherwise we simply let Γ make a small incremental horizontal move to the right and contribute nothing to the winding number.

It is easy to define the centers of revolution, $\phi(w)$, and to connect together the pieces of curve Γ constructed above. So is to prove that they are polynomial-time computable. □

For the membership problem for simply connected regions, we can apply the proof of Theorem 3.1(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.

Corollary 3.3 *Let $f : [0,1] \to [0,1]^2$ 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 3.4 *In the following, (a) $\Rightarrow$ (b) $\Rightarrow$ (c).*
(a) $FP = \#P$.
(b) Every simply connected region S with a P-computable boundary is P-recognizable.
(c) $P = UP$.

4 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 $dist_\Gamma(\mathbf{z}) = \delta(\mathbf{z}, \Gamma)$ is also P-computable? We observe that this problem is close to the minimization problem (Ko & Friedman [1982]; Friedman [1984]), and it turns out they do have the same complexity bounds.

Theorem 4.1 *The following are equivalent:*
(a) $P = NP$.
(b) For any P-computable curve Γ, the function $dist_\Gamma$ is also P-computable.

5 Conformal Mappings and the Riemann Mapping Theorem

Conformal mappings are analytic functions defined on a region in the complex plane. They have many applications in physics and engineering, such as fluid dynamics and aerodynamics. Especially important is the so called

Riemann Mapping Theorem Let S be a simply connected region which is not the entire complex plane, and let a be a point in S. Then there exists a unique one-to-one conformal mapping f from S onto the unit disk $\mathbf{D}(\mathbf{0}, 1) = \{w : |w| < 1\}$ satisfying the conditions

$$f(a) = 0, \quad f'(a) > 0.$$

We call this conformal mapping f the *Riemann mapping*. We are interested in analyzing the computational complexity of this Riemann mapping. But first we have to define an appropriate computational model for mappings from region to region on the complex plane $\mathbf{C}$. Following the same line of thought as in the formulation of P-recognizable set (which is the computational model for the characteristic function of a set), we have to let the machine make errors when the point is close to the boundary. The following formulation is an extension of the notion of "recursively approximable functions" in Ko [1986] (see also Ko[1991]), which is based on recursive measure theory.

Recall that the set of dyadic rationals is denoted by $\mathbf{D} = \{m/2^n : m \in \mathbf{Z}, n \in \mathbf{N}\}$, and $\mathbf{D}_n = \{m/2^n : m \in \mathbf{Z}\}$ is the class of dyadic rationals with at most n bits in the fractional part of its binary representation.

Definition 5.1 *Let S be a bounded region with a polynomial-time computable boundary Γ_S. A function*

$$f : S \to \mathbf{C}$$

is called computable over S *if there exist an oracle Turing machine M such that*

(1) for any oracles (ϕ, ψ) representing a point $\mathbf{z} = \langle x, y\rangle \in \mathbf{C}$ and for any input n, $M^{\phi,\psi}(n)$ halts and outputs either # or some dyadic rational $(d_1, d_2) \in \mathbf{D}_n \times \mathbf{D}_n$, and

(2) for any n, the error set $E_n(M) = \{\mathbf{z} \in \mathbf{C}$: there exists (ϕ, ψ) representing $\mathbf{z}$ such that $|M^{\phi,\psi}(n) - f(\mathbf{z})| \geq 2^{-n}$, or $M^{\phi,\psi}(n) = \#$ but $\mathbf{z} \in S$, or $M^{\phi,\psi}(n) = d$ but $\mathbf{z} \notin S$ $\}$ is contained in a neighborhood of width 2^{-n} around the boundary; i.e.,

$$E_n(M) \subseteq \{\mathbf{z} : \delta(\mathbf{z}, \Gamma_S) \leq 2^{-n}\}.$$

We can add time or space bound to the machine M to obtain resource-bounded computational complexity for the function f in the above definition.

Notice that under this definition the region S plays a major role in the computational complexity of the function f. If the region S is not P-recognizable, then many simple functions, such as $f(\mathbf{z}) = \mathbf{z}^2$, cease to be "computable over S". This only shows that we choose an improper domain of definition for the function. On the other hand, if we consider the function $f(\mathbf{z}) = \sqrt{\mathbf{z}}$, its complexity indeed depends on the region in a crucial way: we have to make a cut (maybe curved) from the origin through the complement of the region to infinity in order to make this function well defined on S.

Modern proofs of the Riemann mapping theorem usually involve a "compactness" argument combined with the theory of normal families, which is non-constructive. Interestingly, Paul Koebe (1882 - 1945) in 1908 already gave a classical and constructive proof of this theorem for bounded regions. In his method, the region S is supposed to be interior to the unit circle, and the boundary of S is gradually pushed out to the circle with the aid of a sequence of elementary transformations involving only the extraction of square roots. This simple idea has reappeared in later proofs; in particular, in Bishop and Bridges [1985]. It has also been polished in numerical computation of conformal mappings, and bears the name "osculation algorithm". We refer to Henrici [1986] for a detailed discussion of this algorithm and its variations.

We analyze the osculation algorithm in the above model of computation and derive an upper bound for the complexity of the Riemann mapping.

We say that a function

$$f : S \to \mathbf{C}$$

is *exponential-space computable over* S if f is computable over S and the oracle Turing machine in the definition 5.1 requires no more than $O(2^{p(n)})$ tape cells to operate on, for some polynomial p.

Theorem 5.2 *Let S be a simply connected region contained in the unit disk $\mathbf{D}(\mathbf{0}, 1)$, $\mathbf{0} \in S$, and the boundary is polynomial-time computable. Let f be the Riemann mapping from S onto $\mathbf{D}(\mathbf{0}, 1)$. Then f is exponential-space computable over S.*

We are currently investigating the possibility of lowering this upper bound. As for the lower bound for the Riemann mapping, little is known, and it seems to be a difficult problem.

References

Chou, A. W. and Ko, K. [1995], Computational complexity of two-dimensional regions, *SIAM J. Comput.* (to appear)

Bishop, E. and Bridges, D [1985], *Constructive Analysis*, Springer-Verlag, Berlin.

Friedman, H. [1984], On the computational complexity of maximization and integration, *Advances in Math.* **53**, 80–98.

Henrici, P. [1986], *Applied and Computational Complex Analysis*, John Wiley & Sons, New York.

Ko, K. [1986], Approximation to measurable functions and its relation to probabilistic computation, *Annals of Pure and Applied Logic* **30**, 173–200.

Ko, K. [1991], *Complexity Theory of Real Functions*, Birkhäuser, Boston.

Ko, K. [1995a], A polynomial-time computable curve whose interior has a nonrecursive measure, *Theoret. Comput. Sci.* (to appear).

Ko, K. [1995b], On the computability of fractal dimensions and Hausdorff measure, preprint.

Ko, K. and Friedman, H. [1982], Computational complexity of real functions, *Theoret. Comput. Sci.* **20**, 323–352.