Towards the Complexity of Riemann Mappings (Extended Abstract)

Robert Rettinger[1]

Department of Mathematics and Computer Science
University of Hagen, Germany

Abstract. We show that under reasonable assumptions there exist Riemann mappings which are as hard as tally ♯-P even in the non-uniform case. More precisely, we show that under a widely accepted conjecture from numerical mathematics there exist single domains with simple, i.e. polynomial time computable, smooth boundary whose Riemann mapping is polynomial time computable if and only if tally ♯-P equals P. Additionally, we give similar results without any assumptions using tally UP instead of ♯-P and show that Riemann mappings of domains with polynomial time computable analytic boundaries are polynomial time computable.

1 Introduction

In this paper we will prove lower bounds on the complexity of Riemann mappings, i.e. conformal mappings of a simply connected domain onto $\mathbb{D}$. Though the existence of such mappings is well known, computability results or even complexity results were unknown for a long time. Despite the fact that constructive proof methods were known for the problem (see [Hen86])) before, a characterization of those domains which have computable Riemann mappings was proven not before [Her99]. In a recent paper, Binder, Braverman and Yampolsky [BBY07] gave sharp bounds on the complexity of the corresponing functor, i.e. the functor which maps domains to their Riemann mappings: This functor is ♯-P complete. (Actually the authors showed that this functor is ♯-P hard and belongs to PSPACE. Using similar techniques, however, even a sharp upper bound of ♯-P can be proven (see [Ret08a]).)

Using the proof techniques of [BBY07] it is not hard to show that this functor remains ♯-P complete even if we restrict the class of domains to those domains which have analytic boundaries. On the other hand, the Riemann mapping of any domain with polynomial time computable analytic boundary can be computed in polynomial time as we will show in Section 4. This underlines that hardness of the functor does not necessarily imply hardness of the mappings themselves and raises the question on the complexity of Riemann mappings in general. In Section 5 we will prove, however, that even the complexity of a single Riemann mapping can be as hard as tally ♯-P under reasonable assumptions. Furthermore we will give a new proof on the (uniform) lower bound of Riemann mappings.

Andrej Bauer, Peter Hertling, Ker-I Ko (Eds.)
6th Int'l Conf. on Computability and Complexity in Analysis, 2009, pp. 209-220
http://drops.dagstuhl.de/opus/volltexte/2009/2272

Our proofs in the non-uniform case will heavily depend on this proof. Besides, this new proof might be of some interest itself as we will use only potential theoretic techniques. Some basic notations of complex analysis, complexity and Type-2 theory are given in the following section.

2 Preliminaries

We denote the set of natural, integer, rational, dyadic, real and complex numbers by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Y}$, $\mathbb{R}$ and $\mathbb{C}$, respectively. Here, a dyadic number is a number of the form $i/2^j$ with i and j integers. As we quite often use the symbol i as an index, we denote the imaginary unit $\sqrt{-1}$ by $\hat{\imath}$ instead. The imaginary and real part of a complex number z are denoted by $\Im(z)$ and $\Re(z)$, respectively. We identify $\mathbb{C}$ and $\mathbb{R} \times \mathbb{R}$ in the usual sense and denote the distance between two numbers z, z' by $d(z, z') = |z - z'|$ and the (Hausdorff)-distance between two sets M, N by $d(M, N) = \sup\{d(z', N), d(z'', M) | z' \in M, z'' \in N\}$, where $d(z, M) = d(M, z) = \inf\{d(z, z') | z' \in M\}$. Furthermore let $\mathbb{D}_\varepsilon(z_0)$ denote the open disc of radius ε with center z_0. To simplify notation we use $\mathbb{D}_\varepsilon := \mathbb{D}_\varepsilon(0)$ and $\mathbb{D} = \mathbb{D}_1$.

For an open subset G of $\mathbb{C}$, a function $f : G \to \mathbb{C}$ is called holomorphic iff its complex derivative f' exists throughout G. A holomorphic function f is called conformal on A iff $|f'(z)| > 0$ for all $z \in A$. If f is conformal throughout its domain we simply say that f is conformal.

Beside functions we allow also multi-functions, denoted by $f :\subseteq M \rightrightarrows N$ and $f : M \rightrightarrows N$ for partial and total multi-function, respectively. We will use both notations $f(x) = y$ and $f(x) \ni y$ to denote that y belongs to the image of x under a multi-function f. Furthermore, for a function (or multi-function) $f :\subseteq G \to G'$ and $H \subseteq G$ we denote the restriction of f to H by $f|_H$.

Before turning to Type-2 objects, we will recall some notions of discrete complexity theory. For more details see e.g. [DK00] or [Sip97]. We denote by FP and P the class of polynomial time computable functions $f : \Sigma^* \to \Sigma^*$ and polynomial time decidable languages $L \subseteq \Sigma^*$, respectively. (Σ denotes here and later on a finite alphabet.) Restricting the alphabet Σ to a single symbol, say 0, leads to tally functions and sets. The corresponding classes will be denoted by the subscript 1, i.e. FP_1, P_1, etc.

Beside we will also need the classes ♯P and UP. ♯P denotes the class of functions $h : \Sigma^* \to \Sigma^*$ so that there exists some $L \in \mathrm{P}$ and polynomial p with $h(u) = |\{v \in \Sigma^* \mid |v| = p(|u|) \wedge (u, v) \in L\}|$ for all $u \in \Sigma^*$. UP denotes the class of languages L so that there exist $\hat{L} \in \mathrm{P}$ and polynomials p with $L = \{v \in \Sigma^* | \exists! u. |u| \leq p(|v|) \wedge (u, v) \in \hat{L}\}$ and $\Sigma^* \setminus L = \{v \in \Sigma^* | \forall u. (u, v) \notin \hat{L}\}$, where $\exists! u$ denotes as usual the fact that there exists exactly one u.

The usual notation of separating complexity classes (or classes in general) is to simply ask for a language, which belongs to the first but not to the second class. Another notion used in literature is separation almost everywhere which can be expressed by the related notion of immune languages (see [DK00]). We will need in this paper a stronger separation notion than the usual one provides. On the other hand we do not need the full power of almost everywhere separation. We will therefore introduce next a kind of separation, which lies between the usual separation and almost everywhere separation. We define this for function classes over the alphabet $\{0\}$ only.

Definition 1. *Let $\Sigma = \{0\}$. Then a function $f : \Sigma^* \to \Sigma^*$ is called selectively separable by a function $s : \mathbb{N} \to \mathbb{N}$ from a class K iff for every $g \in K$, $g : \Sigma^* \to \Sigma^*$ there exists some $i \in \mathbb{N}$ so that $g(0^{s(i)}) \neq f(0^{s(i)})$.*

Furthermore we say that we can separate two classes K_1 and K_2 selectively ($K_1 \neq_{sel} K_2$) iff for every strictly monotone time constructible function $s : \mathbb{N} \to \mathbb{N}$ there exists a function $f : \Sigma^* \to \Sigma^*$ in K_1 which is selectively separable by s from K_2 or vice versa.

Next, let Σ^{**} denote the set $(\Sigma^*)^{\Sigma^*}$, i.e. the set of total functions $f : \Sigma^* \to \Sigma^*$. We fix some standard tuple function $\langle\cdot\rangle$ on $(\Sigma^{**})^n$ mapping products to Σ^{**}.

To give a natural notion of complexity we extend the Type-2-Turing machine model by allowing some kind of indirect access to the input tapes. Formally we realize this by a new definition of representations and the usage of oracle machines, where oracles are elements of Σ^{**}, i.e. functions rather than languages. Queries to the oracle are here answered by the function value of the string on the oracle tape. An oracle Turing machine M computes a function $f_M :\subseteq \Sigma^{**} \to \Sigma^{**}$ in the following sense: $f_M(\alpha)$ is defined to be β iff for each $w \in \Sigma^*$ the machine M together with the oracle α outputs $\beta(w)$. For fixed α we can define the time complexity as usually. We denote the class of such functions of polynomial time complexity (independently of α) by FP_*. In a similar way even relative computations with respect to some oracles can be defined. Details can be found in [Ret08a].

To introduce complexity on more general Type-2 objects we fix a set of standard representations, i.e. surjective functions $\nu :\subseteq \Sigma^* \to M$ or $\nu :\subseteq \Sigma^{**} \to M$ onto the represented set M, next. A (multi)function $g :\subseteq M \to N$ is then called polynomially time computable if there exists a polynomially time computable realization, i.e. a function $f :\subseteq A \to B$ for some $A, B \in \Sigma^{**}$ so that $f \circ \nu_M = \nu_N \circ g$ on $\mathrm{dom}(\nu_M)$ where ν_M and ν_N denote the standard representation of M, N, respectively. We will denote the corresponding complexity class again by FP_*.

Dyadics will be given by their dual representation, denoting the decimal point by ., i.e. $\nu_{\mathbb{Y}}(w.v) = \nu_{\mathrm{dual}}(w) + \nu_{\mathrm{dual}}(v) \cdot 2^{-|v|}$ and $\nu_{\mathbb{Y}}(-w.v) = -(\nu_{\mathrm{dual}}(w) + \nu_{\mathrm{dual}}(v) \cdot 2^{-|v|})$ for $w, v \in \{0,1\}^*$, $w[0] \neq 0$, where ν_{dual} denotes the dual notation of natural numbers. Complex dyadics are represented by pairs of dyadics: $\nu_{\mathbb{Y}[\hat{\imath}]}(\langle d_0, d_1\rangle) = \nu_{\mathbb{Y}}(d_0) + \hat{\imath}\nu_{\mathbb{Y}}(d_1)$ for all $d_0, d_1 \in \mathrm{dom}(\nu_{\mathbb{Y}})$.

A real number x is represented as a sequence of dyadics, which converges fast to x, i.e. $\nu_{\mathbb{R}}(f) = x \Leftrightarrow \forall w \in \Sigma^*. |\nu_{\mathbb{Y}}(f(w)) - x| < 2^{-|w|}$ for all $f \in \Sigma^{**}$. Finally, by identifying $\mathbb{C}$ and $\mathbb{R} \times \mathbb{R}$, we get our standard representation of $\mathbb{C}$ by $\nu_{\mathbb{C}} = \nu_{\mathbb{R}\times\mathbb{R}}$.

Now let G, G' be subsets of $\mathbb{C}$. Then the standard representation $\nu_{\mathcal{A}} :\subseteq \Sigma^{**} \to \mathcal{A}$ of a subclass $\mathcal{A}$ of $\mathrm{Cont}(G, G') = \{g : G \to G' \mid g \text{ continuous}\}$ is defined by

$$g \in \nu_{\mathcal{A}}(f) \Leftrightarrow \forall z \in \nu^{-1}_{\mathbb{Y}[\hat{\iota}]}(G).\forall n \in \mathrm{dom}(\nu_{\mathbb{N}}).|f(\langle n, z\rangle) - g(\nu_{\mathbb{Y}}(z))| < 2^{-n}$$

for all $g \in \mathcal{A}$, $f \in \Sigma^{**}$. The main point of this representation is that we can evaluate functions. For domains there are several different representations. We will use the following representation based on the distance to the boundary.

The representation $\nu^{<}_{\subseteq\mathbb{C}} :\subseteq \Sigma^{**} \rightrightarrows (2^{\mathbb{C}}\backslash\{\mathbb{C}\})$ is defined via a modified distance function. For $f : \mathrm{dom}(\nu_{\mathbb{N}}) \times \mathrm{dom}(\nu_{\mathbb{Y}[\hat{\iota}]}) \to \mathrm{dom}(\nu_{\mathbb{R}})$ let $\nu^{<}_{\subseteq\mathbb{C}}(f) = A \subseteq \mathbb{C}$ iff $3/4 \cdot d(\nu_{\mathbb{Y}[\hat{\iota}]}(z)) - 2^{-\nu_{\mathbb{N}}(n)} < |\nu_{\mathbb{R}}(f(n,z))| < d(\nu_{\mathbb{Y}[\hat{\iota}]}(z))$ for all $n \in \mathrm{dom}(\nu_{\mathbb{N}})$ and all $z \in \nu^{-1}_{\mathbb{Y}[\hat{\iota}]}(A)$, where $d(z) := \inf_{z' \in \partial G} |z - z'|$.

Let A be a represented set. Then we say that a function $f :\subseteq A \to \mathbb{R}^+$ belongs to $\sharp\mathrm{P}_*$, iff there exists a polynomial p and a polynomial time computable function $g :\subseteq A \times \Sigma^* \to \mathbb{R}^+$, so that $f(a) = \sum_{w \in \Sigma^{p(n)}} g(a, w)$ for all $a \in \mathrm{dom}(f)$ (as usually n denotes the length of the input of finite length).

3 Riemann mappings

In this section we will summarize some central results on Riemann mappings.

Theorem 1. *Let G be a bounded simply connected domain. Then for every $z \in G$ and $\phi \in [0; 2\pi]$ there exists a unique conformal mapping $f^{z,\phi}_G : G \to \mathbb{D}$ so that $f^{z,\phi}_G(z) = 0$ and the argument of $(f^{z,\phi}_G)'(z)$ is ϕ.*

We will denote these Riemann mappings usually in the way of the above theorem where we omit ϕ and/or z if $\phi = 0$ and/or $z = 0$, or if these parameters are uniquely determined by the context. If G is a Jordan domain, the Riemann mapping continues topologically onto the boundary (see [Pom92] for details). If the boundary γ of G is even analytic, the Riemann mapping continues even holomorphically, which can be easily seen by the reflection principle.

To simplify things, we will restrict ourselves in the sequel to the class of simply connected domains which are contained in the disk $\mathbb{D}_{4/5}$ and contain the disk $\mathbb{D}_{3/5}$. This class of simply connected domains will be denoted by $\mathbb{G}$ in the sequel. For more general classes of simply connected domains the ideas given below can be easily adapted as long as the domains are bounded. This can for example be achieved by the usual square root transformation (see e.g. [Hen86]). Alternatively, the osculation method can be used to reduce the domain. This method converges fast as long as the domain is far away from the unit disc (with respect to the Hausdorff distance). Furthermore we will compute the Riemann mapping on a fixed compact subset of its domain. We can then get the full Riemann mapping by continuation (see e.g. [Ret08b]).

Theorem 2 ([BBY07],see also [Ret08a]). *There exists a function $F_{conf} :\subseteq \mathbb{G} \times \mathbb{D}_{1/2} \to \mathbb{D}$, $F \in FP^{\sharp P_*}_*$, mapping each simply connected domain $G \in \mathbb{G}$ and point $z \in \mathbb{D}_{1/2}$ to $f_G(z)$.*

If we restrict the above function to boundaries, which can be computed in time bounded by a fixed polynomial, then the Riemann mapping can be computed by polynomially time bounded machines with access to a $\sharp P$-oracle.

The proof of the above theorem shows that slight changes in the shape of the domain G will only slightly change the Riemann mapping. We will use this fact e.g. to give a polynomial upper bound for the Riemann mapping for analytic boundaries in Section 4 below.

Corollary 1. *There exists a polynomial p so that for all $G, G' \in \mathbb{G}$ we have: The Riemann mappings f and g of G and G', respectively, determined by $f(0) = g(0) = 0$ and $f'(0) > 0$, $g'(0) > 0$ differ by at most 2^{-n} on $z \in \mathbb{D}_{1/2}$, i.e. $|f(z) - g(z)| \leq 2^{-n}$, if the Hausdorff distance of G and G' is at most $2^{-p(n)}$.*

4 Analytic Boundaries

In this section we will show that for any simply connected Jordan domain G with analytic, polynomial time computable boundary, the Riemann mapping from G is always computable in polynomial time. To prove this we will use a technique based on the Bergman kernel function and orthonormal polynomials.

For given $G \in \mathbb{G}$ and $i \in \mathbb{N}$ let in the sequel p_i denote the i-th orthonormal polynomial, determined by the sequence 1, z, z^2, ... and the Gram-Schmidt algorithm, using the inner product $\langle \cdot, \cdot \rangle$ defined by

$$\langle f, g \rangle = \int \int_G f(z)\overline{g(z)}dxdy$$

for all $f, g \in L^2(G, \mathbb{C})$, where $L^2(G, \mathbb{C})$ denotes the space of square integrable complex functions on G (see e.g. [Gai87]).

Lemma 1. *Let $G \in \mathbb{G}$ be a Jordan domain with its boundary given by a polynomial time computable conformal mapping $\delta : U \to \mathbb{D}$ of an open neighborhood U of $\partial\mathbb{D}$. Then the sequence p_0, p_1 ,... of orthonormal polynomials is computable in polynomial time.*

Notice that orthonormal polynomials can be computed efficiently even in other cases, e.g. in the case of Schwarz-Christoffel mappings. However, it is not known, if the polynomials in this case can be used to compute the Riemann mapping efficiently.

Once we have these orthonormal polynomials for a domain $G \in \mathbb{G}$, we can build a fast algorithm to compute the Riemann mapping upon a well known relation of the Riemann mapping and the Bergmann kernel $K : G \times G \to \mathbb{R}$ (see e.g. [Neh52]).

Theorem 3. *Let $G \in \mathbb{G}$ be a Jordan domain with its boundary given by a polynomial time computable conformal mapping $\delta : U \to \mathbb{D}$ of an open neighborhood U of $\partial\mathbb{D}$. Then f_G is computable in polynomial time, where f_G denotes the uniquely determined Riemann mapping with $f_G(0) = 0$ and $f'_G(0) > 0$.*

Proof: We have the well know relation between the Bergmann kernel function and f_G

$$f'_G(z) = \sqrt{\frac{\pi}{K(0,0)}} \cdot K(z,0).$$

Furthermore the Bergmann kernel function can be expressed by means of the orthonormal polynomials p_0, p_1, ... of G via $K(z,0) = \sum_{j=0}^{\infty} \overline{p_j(0)} \cdot p_j(z)$, where the convergence is uniformly on any compact subset of G. Approximating the Bergmann kernel function by K_n $(n \in \mathbb{N})$, where $K_n(z,0) = \sum_{j=0}^{n-1} \overline{p_j(0)} \cdot p_j(z)$ gives us the the Bieberach polynomials q_i $(i \in \mathbb{N})$, determined by $q_i(z) = \sum_{j=0}^{n} \frac{\int_0^z K_{i-1}(z,\zeta)d\zeta}{\sqrt{K_{i-1}(0,0)}}$ for $i \in \mathbb{N}$.

By Lemma 1, the Bieberach polynomials can be computed in polynomial time. Notice, as the sequence $K_i(0,0)$ converge to $K(0,0) \neq 0$, the $K_i(0,0)$ are bounded away from 0 by a constant for all but finitely many i's. Furthermore we know that for analytic boundaries there exists $M > 0$ and $q \in (0;1)$ so that $|f_G(z) - q_i(z)| < M \cdot q^i$ for all z in say $\mathbb{D}_{1/2}$ and all $i \in \mathbb{N}$ (see [Gai87]). Thus we can compute f_G in polynomial time on $\mathbb{D}_{1/2}$. As continuation of holomorphic mappings can be done in polynomial time (see e.g. [Mül93]) and the fact that f_G can be continued to a whole neighborhood of G by the reflection principle, proves that f_G can be computed in polynomial time throughout G.

□

5 Towards lower bounds

In this section we will first give a new proof for the lower bound on Riemann mappings in the uniform case first shown in [BBY07]. Afterwards we will turn to the non-uniform case. Our proof of the following theorem will use only basic ideas of potential theory.

Theorem 4 ([BBY07]). *If $F : \mathbb{G} \to \mathbb{D}$ with $F(G) = f'_G(0)$ is computable in polynomial time, then every function in $\sharp P_*$ is computable in polynomial time. Especially we have that if F is computable in polynomial time then $\sharp P = FP$.*

Even for restrictions of F to those domains $G \in \mathbb{G}$ whose boundaries are analytic or polygons, this result holds, i.e. if this restrictions are computable in polynomial time then $\sharp P_* = FP_*$.

Proof: The second statement follows from the first one by suitable approximations of general domains by the restricted ones using Corollary 1 above.

To prove the first statement, notice that $f'_G(0) = e^{-u}$, where u is the solution of the Dirichlet problem with boundary values $z \mapsto \log_e(|z|)$ for $z \in \partial G$. We will thus code the behavior of a Turing machine M into such a boundary value problem. In contrast to the construction in [BBY07] we will use the slit map rather than the crescent map, which simplifies things further. Nevertheless, using the ideas below, even the construction of the domains in [BBY07] could be used to prove the above result with potential theoretic ideas only.

Our construction will be based on the slit map (see [Hen86], Chapter 16). Let therefore, for given $\rho \in (0;1)$, $S(\rho)$ denote the straight line from -1 to $-\rho$ and furthermore, by D_ρ the set D_ρ. Then a conformal mapping $h_\rho : D_\rho \to \mathbb{D}$ with $h_\rho(0) = 0$ is given by $h_\rho(z) = (s_\rho(z) - 1 + z)/(s_\rho(z) + 1 - z)$ for all $z \in \mathbb{D} \setminus S_\rho$, where we use the abbreviation $s_\rho(z) = \sqrt{(1+\rho(z))(1+\frac{1}{\rho}z)}$. Furthermore for $h'_\rho(0)$ we have $h'_\rho(0) = \frac{(1+\rho)^2}{4\rho}$ (see [Hen74]).

The main point of giving this map explicitly is that we can easily compute $|h'_\rho(0)|$ and thus $\log|h'_\rho(0)|$.

Claim. There exists a mapping $h : (1/2;1) \to \mathbb{R}$ with $h(\rho) = h'_\rho(0)$ for all $\rho \in (1/2;1)$, which is computable in time $O(n^2)$. Furthermore there exist constants $c_0, c_1, c_2 > 0$, so that $c_0 \cdot (1-\rho)^2 < h'_\rho(0) - 1 < c_1 \cdot (1-\rho)^2$ and $|\log_e(h'_\rho(0)| > c_2 \cdot (1-\rho)^2$ for all $\rho > 3/4$.

We assume now that F is polynomial time computable and $L \in \sharp P_* \setminus FP_*$, $L : A \to \mathbb{R}^+$ for some represented space A. In a first step we will reduce L to a problem in $\sharp P$. Let M be a polynomial time computable Turing machine and q be a polynomial, so that on every input $a \in A$, $n \in \mathbb{N}$ and $w \in \Sigma^*$ with $|w| = q(n)$, M stops in exactly $q(n)$ steps, outputs $o_M(a,n,w) \in \mathbb{Y}^+$ and fulfills

$$|L(a) - \sum_{w \in \Sigma^{q(n)}} o_M(a,n,w)| \leq 2^{-n}.$$

For given input $a \in A$ and a precision 2^{-n}, we are thus asked to compute $L(a)$ up to this precision. As we have to add up at most $2^{q(n)}$ values, we have to compute each of the elements of the above sum up to precision $2^{-(n+q(n))}$ only. This can be done by an addition of $2^{q(n)}$ integers of at most $n + q(n)$ bits each with an appropriate shift afterwards. As this shift is polynomial time computable, we can, by a standard manipulation of M, give a Turing machine N and a polynomial p with the following properties:

1. N stops on input $a \in A$, $n \in \mathbb{N}$ and every $w \in \Sigma^{p(n)}$ in at most $p(n)$ steps with output $o_N(a,n,w) \in \{0,1\}$ and
2. $L(a)$ can be computed from $\hat{L}(a,n) = \sum_{w \in \Sigma^{p(n)}} o_N(a,n,w)$ in polynomial time for every $a \in A$ and $n \in \mathbb{N}$.

Let some $a \in A$, $n \in \mathbb{N}$ with $n > 2$ be given. We construct, using the slit map above, some $G_{a,n} \in \mathbb{G}$ so that for $f_{G_{a,n}}$ with $f_{G_{a,n}}(z) = F(G_{a,n}, z)$ we have

$$\hat{L}(a,n) = \lfloor \log_e(f'_{G_{a,n}}(0))/\log_e(h'_{1-2^{-m}}(0)) \rfloor,$$

where m is polynomially bounded in n and will be chosen later on. Thus, if F is polynomial time computable, clearly L is polynomial time computable too. We will give here a slightly more general result than necessary by introducing an additional parameter ε. We will need this general result in the proof of Theorems 5 and 6 later on. For given $v \in \Sigma^{p(n)}$ and $\varepsilon \in (0;2\pi)$ let $\phi_v^\varepsilon := \varepsilon 2^{-(p(n))} \cdot$

$(\nu_{\text{dual}}(1v) - 2^{p(n)})$. The values ϕ_v^ε of all such v are in the interval $[0;\varepsilon]$ and for different $u, v \in \Sigma^{p(n)}$ we have $|\phi_v - \phi_u| \geq \varepsilon \cdot 2^{-p(n)}$. Now let

$$G_{a,n}^\varepsilon = \bigcup_{\substack{v \in \Sigma^{p(n)} \\ o_N(a,n,v)=1}} e^{-i\phi_v^\varepsilon} \cdot D_{1-2^{-m_\varepsilon}}.$$

Notice that for fixed $\varepsilon \in \mathbb{Y} \cap (0;1)$, the function $H : A \times \mathbb{N} \to \mathbb{G}$ with $H(a,n) = G_{a,n}^\varepsilon$ for all $a \in A$, $n \in \mathbb{N}$, is polynomial time computable. By the above discussion it remains to show that we can compute $\hat{L}(a,n)$ efficiently from $f'_{G_{a,n}^\varepsilon}(0)$, because then L can be computed in polynomial time in contradiction to our assumption.

Claim. Let $\varepsilon \in (0;2\pi)$, $n \in \mathbb{N}$ be given and $m_\varepsilon(n) = \lceil 4 \cdot p(n) + \log(1/\varepsilon) + \log(c_2) \rceil + 15$. Then $\hat{L}(a,n) = \lfloor \log_e(f'_{G_{a,n}^\varepsilon}(0)) / \log_e(h'_{1-2^{-m_\varepsilon(n)}}(0)) \rfloor$ for all $n \in \mathbb{N}$.

As we consider a fixed $n \in \mathbb{N}$ in the sequel we will write m_ε instead of $m_\varepsilon(n)$, for short. The main work in showing the above equation is, to bound the cross terms introduced to the Riemann mapping when combining the different slit maps. To this end we will use the relation between the Brownian motion and potentials as already considered in the last section. Let therefore, for given $G \in \mathbb{G}$, $z \in G$ and $Z \subseteq \partial G$, $p_G(Z|z)$ denote the probability to end up in Z when we start in z. To be more precise, let for $z \in G$, $B_G^t(z)$ denote the Brownian motion process, which starts in z. Furthermore let T be the first time $B_G^t(z)$ hits the boundary ∂G. Then for given continuous or piecewise constant and bounded values $v(x) \in \mathbb{R}$ (for boundary points $x \in \partial G$) we know that $f : G \to \mathbb{R}$, $f(z) = E(v(B_G^T(z)))$, is the unique solution to the corresponding Dirichlet problem (where $E(X)$ denotes the expectation of the random variable X). Furthermore $p_G(Z|z)$ is the expectation $p_G(Z|z) = E(\chi_Z(B_G^T(z)))$. To simplify things we will in addition use the notation $p_G(Z|z \rightsquigarrow Z')$ meaning the probability to end up in $Z \subseteq \partial G$, starting in $z \in G$ and visiting at least once a point in $Z' \subseteq G$.

A main tool in bounding the probabilities is the Poisson formula

$$u(z) = \frac{1}{2\pi} \cdot \int_{\partial\mathbb{D}} v(y) \cdot \frac{1 - |z|^2}{|z-y|^2} dy$$

which gives an explicit solution to the Dirichlet problem if $G = \mathbb{D}$. Unfortunately, however, G_w^ε is likely to be not $\mathbb{D}$ (unless $\hat{L}(n,a) = 0$). By the following result we can nevertheless use Poissons formula, where we use the abbreviation $\partial_{m_\varepsilon} := \partial\mathbb{D} \cap \mathbb{D}_{2^{-m_\varepsilon}}(-1)$:

Claim. For all $z \in S_{1-2^{-m_\varepsilon}}$ we have

$$p_{\mathbb{D}}(\partial_{m_\varepsilon}|z) = \frac{1}{2\pi} \cdot \int_{\partial\mathbb{D}} \chi_{\partial_{m_\varepsilon}}(y) \cdot \frac{1 - |z|^2}{|z-y|^2} dy > \frac{3}{4}.$$

Let $u : G^\varepsilon_{a,n} \to \mathbb{R}$ be the solution of the Dirichlet problem with boundary values $v(x) = \log(|x|)$ for $x \in \partial G^\varepsilon_{a,n}$, especially we have $f'_{G^\varepsilon_{a,n}}(0) = e^{-u(0)}$. We will bound the difference of $u(0)$ and $\hat{L}(n,a) \cdot |\log_e(h'_{1-2^{-m_\varepsilon}}(0))|$ accordingly. Notice that $\hat{L}(n,a)$ is the number of slits in $G^\varepsilon_{a,n}$. Each slit, say at angle $\phi = \phi^\varepsilon_v$, taken alone, adds a value $\log_e(h'_{1-2^{-m_\varepsilon}}(0))$ to u_0. However, not every path in the Brownian motion, which ends at the slit $e^{\hat{\imath}\phi} \cdot S_{2^{-m_\varepsilon}}$ on $(e^{\hat{\imath}\phi} \cdot D_{2^{-m_\varepsilon}})$, will also end there on $G^\varepsilon_{a,n}$, because it might hit another slit in between. (As $\log_e(1) = 0$ only the hits of slits are counted.) To simplify things we will use the abbreviation $S_\phi := e^{\hat{\imath}\phi} \cdot S_{2^{-m_\varepsilon}}$ and $D_\phi = \mathbb{D} \setminus S_\phi$ in the sequel.

As $p_{G^\varepsilon_{a,n}}(Z|z \rightsquigarrow Z') \le p_{G^\varepsilon_{a,n}}(Z'|z) \cdot \sup_{z' \in Z'} p_{G^\varepsilon_w}(Z|z')$ for all $z, z' \in G^\varepsilon_{a,n}$ and $Z, Z' \subseteq G^\varepsilon_{a,n}$, we can bound the difference $|u(0) - \hat{L}(n,a) \cdot \log_e(h'_{1-2^{-m_\varepsilon}}(0))|$ by the sum of the probabilities to miss a slit $S_{\phi^\varepsilon_v}$ in $G^\varepsilon_{a,n}$, because of hitting a slit $S_{\phi^\varepsilon_u}$ first. For given $z \in S_{\phi^\varepsilon_u}$ we have

$$p_{G^\varepsilon_{a,n}}(S_{\phi^\varepsilon_v}|z) \le \frac{4}{3}\frac{1}{2\pi} \cdot \int_{\partial\mathbb{D}} \chi_{\partial_{m_\varepsilon}}(y) \cdot \frac{1-|z|^2}{|z-y|^2} dy.$$

As $1 - |z|^2 \le 2^{-m_\varepsilon+1} - 2^{-2m_\varepsilon} \le 2^{-m_\varepsilon+1}$ and $|z-x|^2 \ge ((1/\pi) \cdot (\varepsilon \cdot 2^{-p(n)} - 2 \cdot 2^{-m_\varepsilon}))^2 \ge 2^{-2p(n)+3}$ for all $z \in S_{\phi^\varepsilon_u}$ and $x \in \partial_{m_\varepsilon}$, we get

$$p_{G^\varepsilon_{a,n}}(S_{\phi^\varepsilon_v}|z) \le \frac{4}{3} \cdot (2 \cdot \pi \cdot 2^{-m_\varepsilon+1}) \cdot (2^{-m_\varepsilon+1}/(\varepsilon \cdot 2^{-(2p(n)+3)})).$$

Furthermore we have $p_{G^\varepsilon_w}(S_{\phi_u}|0) \le \frac{4}{3} \cdot 2\pi \cdot 2^{-m_\varepsilon+1}$ thus giving

$$|u(0) - \hat{L}(n,w) \cdot \log_e(h'_{1-2^{-m_\varepsilon}}(0))| \le \frac{2^{2p(n)} \cdot 2^{-2(m_\varepsilon-5)} \cdot 2^{-m_\varepsilon+1}}{(\varepsilon \cdot 2^{-2(p(n)+3)})}.$$

Notice that by the above cross-terms the probability to hit a slit is decreased, i.e. $u(0) \ge \hat{L}(n,a) \cdot \log_e(h'_{1-2^{-m_\varepsilon}}(0))$. As $|\log_e(h'_{1-2^{-m_\varepsilon}}(0))| > c_2 \cdot 2^{-2 \cdot m_\varepsilon}$ we get

$$\hat{L}(n,a) + 1/2 \ge u(0)/\log_e(h'_{1-2^{-m_\varepsilon}}(0)) \ge \hat{L}(n,a)$$

which proves the theorem. □

The previous theorem states that we cannot compute the Riemann mappings for all G in a uniform way. As shown in Section 4, this does not mean that the Riemann mapping for each $G \in \mathbb{G}$ cannot be computed in polynomial time. This raises the question, wether there exists a single domain G in $\mathbb{G}$, which is polynomial time computable, but the Riemann mapping f_G of G is not polynomial time computable under reasonable assumptions. We restrict ourselves to computing this map on a small neighborhood of 0 and we will answer this question affirmative under the following conditions:

1. if $UP_1 \neq_{sel} FP_1$ or
2. if $\sharp P_1 \neq_{sel} FP_1$ and in addition Conjecture 1 on the existence of Schwarz-Christoffel mappings holds.

Notice that any such result is involved with tally classes, i.e. classes of languages in $\{0\}^*$ rather than languages over alphabets with more symbols. This stems from the fact that we can compute the Riemann mapping on any compact subset of G, say to precision 2^{-n}, by asking a single question to a $\sharp P$ oracle. (Actually we need a polynomial number of such queries. However, these can be coded into a single query of a modified oracle.)

We will start to prove the existence of the domain G under the first condition.

Theorem 5. *If $UP_1 \neq_{sel} FP_1$ then there exists a polynomial time computable domain $G \in \mathbb{G}$, so that f_G is not polynomial time computable.*

Proof: Let L be a function in UP_1, which is selectively separated from FP_1. L is obviously a function in $\sharp P$ with values in $\{0,1\}$. Thus we can use all the notations of the proof of Theorem 4 also here. Notice, however that we start already with some function in $\sharp P$ and thus we do not have to reduce to such a function first. So we use L instead of $\hat{L}$ here and furthermore the parameter a used in the proof of Theorem 4 does not appear here. Especially, let N be a Turing machine and p be a polynomial with the following properties:

1. N stops on input w and every $v \in \Sigma^{p(|w|)}$ in at most $p(|w|)$ steps with output $o_N(w,v) \in \{0,1\}$ and
2. $L(w) = \sum_{v \in \Sigma^{p(|w|)}} o_N(w,v)$

for every $w \in \Sigma^*$.

The main idea of the proof is as follows. Using the techniques of the proof of Theorem 4 above, we construct a domain $G \in \mathbb{G}$ in steps i, where we determine in each step a domain G_i, a conformal mapping $f_i : G_i \to \mathbb{D}$ and a natural number n_i so that (1) we can compute $L(0^{n_i})$ from $f'_{G_i}(0)$ in polynomial time and (2) G_i differs from G_{i+1} (in the Hausdorff metric) by at most 2^{-n_i+1}.

Thus, by defining the n_i large enough, we can ensure that there exists a $G \in \mathbb{G}$ with $d_H(G_i, G) \leq 2^{-n_i+2}$ and the difference of $f'_G(0)$ and $f'_{G_i}(0)$ is small enough, so that we can still compute $L(0^{n_i})$ from $f'_G(0)$ in polynomial time. To this end we have simply to ensure that $n_{i+1} > q(n_i)$, where the polynomial is given by Corollary 1. Once we have constructed G_i, f_i and n_i with this property we proceed in step $i+1$ as follows: First we find some n'_{i+1} so that we can compute f_i in polynomial time for all inputs of length at least n'_{i+1}. Then we choose n_{i+1} to be the maximum of $q(n_i)$ and n'_{i+1}. Following the idea of the proof of Theorem 4 we can compute a domain $G^1_{n_{i+1}}$ so that $L(0^{n_{i+1}})$ can be computed from $f'_{G^1_{n_{i+1}}}(0)$ in polynomial time. If we finally fix G_{i+1} to be $G_{i+1} = f_i^{-1}(G^1_{n_{i+1}})$ and $f_{i+1} = f_{G^1_{n_{i+1}}} \circ f_i$, we can still compute the value $L(0^{n_{i+1}}$ from $f'_{i+1}(0)$ in polynomial time: Simply divide $f'_{i+1}(0)$ by $f'_i(0)$ to get $f'_{G^1_{n_{i+1}}}(0)$. As $f'_i(0)$ can be computed in polynomial time by choice of n'_{i+1}, we are done. Notice that we can define n'_{i+1} because f_i is a composition of Riemann mappings $f_{G'}$ for slit-maps G' as $L(0^{n_i}) \in \{0,1\}$ for all i.

□

Using the ideas of the previous proof, we can also show the existence of G in the second case. Before giving this result we need to specify the conjecture on the existence of efficient algorithms for Schwarz-Christoffel mappings.

We will consider polygons given by the list of their vertices, which we assume to be complex dyadics. Furthermore we restrict ourselves to polygons which are the boundary of some domain in $\mathbb{G}$. Let Polygon be the set of the polygons restricted in such a way. Furthermore we introduce a standard representation $\nu_{Polygon} :\subseteq \Sigma^* \to$ Polygon by simply taking $\langle d_1, ..., d_n \rangle$ to be a $\nu_{Polygon}$-name for a polygon γ, iff the d_i's are names of the complex dyadic vertices of γ in counter clockwise order. Finally we will not distinguish between polygons γ and the corresponding domains with boundary γ, which we denote by $I(\gamma)$.

By the well known Schwarz-Christoffel formula (see e.g. [DT02]), f_γ is determined by

$$f_\gamma^{-1}(z) = C \cdot \int_0^z (1 - x/z_k)^{\alpha_k - 1} dx$$

where z_k are the images $f_\gamma(w_i)$ of the vertices w_i of γ, $\alpha_k \pi$ are the interior angles of γ and C is a positive real number. We can compute the integral above quite efficiently once we know C and z_1,...,z_k. The determination of these parameters is called the parameter problem of the Schwarz-Christoffel mapping. The usual way to solve this problem in numerics is to consider the non-linear system composed of the side-length conditions and a transformation to get an unconstrained system, i.e. to get rid of the condition on the ordering of the vertices and images of the vertices. Then this system of equations is solved by well known methods. There exist however examples, where this leads to local solutions which are not solutions for the Schwarz-Christoffel parameter problem. We do not know wether these methods are applicable to our problem. Notice however that in contrast to the general parameter problem, the polygons used in the proof below, can be chosen up to some degree, thus probably simplifying the problem.

There are other methods to solve the parameter problem, for example by deriving conditions on the so called cross ratios (see [DV98]). This seemingly leads to equations, which might be solvable efficiently in general.

Unfortunately, however, there does not exist an analysis of these methods, which can be translated to the rigorous definition of complexity we need. Thus we will give here the result we need, and which is claimed in a much stronger sense in numerical analysis, as a conjecture.

Conjecture 1. There exists a polynomial p and a computable function F_{SC} : Polygon $\times \mathbb{D} \to \mathbb{N} \times \mathbb{D}$ so that for each $\gamma \in$ Polygon there exist n_γ so that $F_{SC}(\gamma, z) = (n_\gamma, f_\gamma^{-1}(z))$ for all $z \in \mathbb{D}$, and F_{SC} is computable in time $O(n_\gamma \cdot p)$.

Here f_γ denotes the Riemann mapping with $f_\gamma(0) = 0$ and $f'_\gamma(0) > 0$.

Using a similar proof technique as in the first case we can show the existence of the domain G also in the second case:

Theorem 6. *There exists a polynomial time computable (Jordan) domain $G \in \mathbb{G}$, so that f_G is not polynomial time computable if $\sharp P_1 \neq_{sel} FP_1$ and Conjecture 1 holds.*

6 Remarks

We have proved lower bounds on the complexity of Riemann mappings even in the computational case. As shown, a proof that the parameter problem of Schwarz Christoffel mappings is polynomial time computable, which is undoubtedly interesting on its own, would improve upon the bound we have given. Another interesting question is, wether it is possible to prove such a result for more general separation assumptions than the selective separation we have used.

Finally, a more general connection between orhtonormal polynomials and f_G for domains with non-analytic boundaries would be interesting. (Such results exist, but the corresponding speed of convergence for the Bieberach polynomials is too slow to be reasonable applicable, see e.g. [Gai87].)

For domains with analytic boundaries, a polynomial time algorithm for the Riemann mapping can be also deduced differently, by different relation of the Riemann mapping and orthonormal polynomials via a theorem by Carlemann (see [Gai87] for more details).

References

[BBY07] I. Binder, M. Braverman, and M. Yampolsky, *On computational complexity of Riemann mapping*, Arkiv for Matematik (2007), to appear.

[DK00] D.-Z. Du and K.-I Ko, *Theory of computational complexity*, Wiley-Interscience Series in Discrete Mathematics and Optimization, 2000.

[DT02] T. A. Driscoll and L. N. Trefethen, *Schwarz–Christoffel mapping*, Cambridge Monographs on Applied and Computational Mathematics, vol. 8, Cambridge University Press, Cambridge, UK, 2002.

[DV98] T. A. Driscoll and S. A. Vavasis, *Numerical conformal mapping using cross-ratios and Delaunay triangulation*, SIAM Journal on Scientific and Statistical Computing **19** (1998).

[Gai87] D. Gaier, *Lectures on complex approximation*, Birkhäuser, 1987.

[Hen74] P. Henrici, *Applied and complex analysis. Vol. 1*, Pure and Applied Mathematics, John Wiley & Sons, New York, 1974.

[Hen86] ———, *Applied and computational complex analysis. Vol. 3*, Pure and Applied Mathematics, John Wiley & Sons, New York, 1986.

[Her99] P. Hertling, *An effective Riemann Mapping Theorem*, Theoretical Computer Science **219** (1999), 225–265.

[Mül93] N. Th. Müller, *Polynomial time computation of Taylor series*, JAIIO - Panel, Part 2, 1993, Buenos Aires, 1993, pp. 259–281.

[Neh52] Z. Nehari, *Conformal mapping*, Dover Publications, New York, 1952.

[Pom92] Ch. Pommerenke, *Boundary behaviour of conformal maps*, Springer-Verlag, 1992.

[Ret08a] R. Rettinger, *Computability and complexity aspects of univariate complex analysis – habilitation thesis*, 2008.

[Ret08b] ———, *On the continuation of holomorphic functions*, CCA, 2008.

[Sip97] M. Sipser, *Introduction to the theory of computation*, PWS Publishing, 1997.