On effective analytic continuation

Joris van der Hoeven

Mathématiques, Bâtiment 425
CNRS, Université Paris-Sud
91405 Orsay Cedex
France

Email: joris@texmacs.org
Web: http://www.math.u-psud.fr/~vdhoeven

Until now, the area of symbolic computation has mainly focused on the manipulation of algebraic expressions. It would be interesting to apply a similar spirit of "exact computations" to the field of mathematical analysis.

One important step for such a project is the ability to compute with computable complex numbers and computable analytic functions. Such computations include effective analytic continuation, the exploration of Riemann surfaces and the study of singularities. This paper aims at providing some first contributions in this direction, both from a theoretical point of view (such as precise definitions of computable Riemann surfaces and computable analytic functions) and a practical one (how to compute bounds and analytic continuations in a reasonably efficient way).

We started to implement some of the algorithms in the Mmxlib library. However, during the implementation, it became apparent that further study was necessary, giving rise to the present paper.

Keywords: analytic continuation, Riemann surface, algorithm, differential equation, convolution equation, relaxed power series, error bound

A.M.S. subject classification: 03F60, 30-04, 30B40, 30F99

1. Introduction

Although the field of symbolic computation has given rise to several softwares for mathematically correct computations with algebraic expressions, similar tools for analytic computations are still somewhat inexistent.

Of course, a large amount of software for numerical analysis does exist, but the user generally has to make several error estimates by hand in order to guarantee the applicability of the method being used. There are also several systems for interval arithmetic, but the vast majority of them works only for fixed precisions. Finally, several systems have been developed for certified arbitrary precision computations with polynomial systems. However, such systems cannot cope with transcendental functions or differential equations.

The first central concept of a systematic theory for certified computational analysis is the notion of a *computable real number*. Such a number $x \in \mathbb{R}$ is given by an *approximation algorithm* which takes $\varepsilon \in \mathbb{R}^{\mathrm{dig}} = \mathbb{Z}\ 2^{\mathbb{Z}}$ with $\varepsilon > 0$ on input and which produces an *ε-approximation* $\tilde{x} \in \mathbb{R}^{\mathrm{dig}}$ for x with $|\tilde{x} - x| < \varepsilon$. One defines computable complex numbers in a similar way.

The theory of computable real numbers and functions goes back to Turing [Tur36] and has been developed further from a theoretical point of view [Grz55, Alb80, BB85, Wei00]. It should be noticed that computable real and complex numbers are a bit tricky to manipulate: although they easily be added, multiplied, etc., there exists no test for deciding

whether a computable real number is identically zero. Nevertheless, possibly incomplete zero-tests do exist for interesting subclasses of the real numbers [Ric97, MP00, vdH01b]. In section 2.5, we will also introduce the concept of semi-computable real numbers, which may be useful if a zero-test is really needed.

The subject of computable real numbers also raises several practical and complexity issues. At the ground level, one usually implements a library for the evaluation of basic operations $+$, $-$, $\times$, etc. and special functions $\exp$, $\log$, $\sin$, etc. Using fast multiplication methods like the FFT [KO63, CT65, SS71], this raises the question of how to do this in an asymptotically efficient way [Bre76a, Bre76b, CC90, Kar91, vdH99a, vdH01a, vdH05b]. At an intermediate level, one needs a software interface for certified operations with arbitrary precision numbers. Several implementations exist [FHL+05, GPR03, Mül00, vdH99b, vdH06b], which are mostly based on correct rounding or interval arithmetic [Moo66, AH83, Neu90, JKDW01, BBH01, Bla02]. At the top level, one may finally provide a data type for real numbers [MM96, Mül00, Lam06, O'C05, vdH06a, vdH06b]. Given the real number result of a complex computation, an interesting question is to globally optimize the cost of determining a given number of digits of the result, by automatically adjusting the precisions for intermediate computations [vdH06a, vdH06b].

The next major challenge for computational analysis is the efficient resolution of more complicated problems, like differential or functional equations. In our opinion, it is important to consider this problem in the complex domain. There are several reasons for this:

- Most explicitly stated problems admit analytic (or semi-analytic) solutions.
- The locations of the singularities of the solutions in the complex plane give important information on the optimal step-size for numerical algorithms.
- The behaviour of the solutions near singularities gives important information on the nature of these solutions.
- Analytic functions are very rigid in the sense that they are entirely determined by their power series expansion at a point, using the process of analytic continuation.

This paper aims at providing a basic theoretical framework for computations with computable analytic functions and effective analytic continuation. When possible, our study is oriented to efficiency and concrete implementability.

The history of analytic continuation of solutions to complex dynamical systems goes back to the 19-th century [BB56]. Although interval arithmetic and Taylor models have widely been used for certified numeric integration of dynamical systems [Moo66, Loh88, MB96, Loh01, MB04], most implementations currently use a fixed precision [Ber98]. Some early work on effective analytic continuation in the multiple precision context was done in [CC90, vdH99a, vdH01a, vdH05b]; see also [vdH07] for some applications. Of course, fast arithmetic on formal power series [BK75, BK78, vdH02b] is an important ingredient from the practical point of view. Again, the manipulation of computable analytic functions is very tricky. For instance, even for convergent local solutions to algebraic differential equations with rational coefficients and initial conditions, there exists no general algorithm for determining the radius of convergence [DL89]. Of course, one also inherits the zero-test problem from computable complex numbers.

Let us detail the structure and the main results of this paper. In section 2, we start by recalling some basic definitions and results from the theory of computable real numbers. In particular, we recall the concepts of left computable and right computable real numbers, which correspond to computable lower resp. upper bounds of real numbers.

In section 3, we introduce the concept of a computable Riemann surface. In a similar way as computable real numbers are approximated by "digital numbers" in $\mathbb{Z}\,2^{\mathbb{Z}}$, we will approximate computable Riemann surfaces by so called "digital Riemann surfaces", which are easier to manipulate from an effective point of view. For instance, in section 3.2, we will see how to identify two branches in a digital Riemann surface. However, from a conceptual point of view, it is not always convenient to see Riemann surfaces as limits of sequences of digital approximations. In sections 3.4 and 3.5, we will therefore discuss two equivalent ways to represent computable Riemann surfaces. Notice that all Riemann surfaces in this paper are above $\mathbb{C}$.

The next section 4 deals with constructions of several kinds of computable Riemann surfaces. We start with the definition of computable coverings (which can be thought of as morphisms of computable Riemann surfaces) and the construction of the limit of a sequence of coverings. We proceed with the definition of disjoint unions, covering products, quotients and joins at a point. For instance, if $\mathcal{R}_f$ and $\mathcal{R}_g$ are the Riemann surfaces of two analytic functions f resp. g, then $f+g$ and fg are defined on the covering product $\mathcal{R}_f \bowtie \mathcal{R}_g$ of $\mathcal{R}_f$ and $\mathcal{R}_g$. In section 4.4, we consider Riemann surfaces which admit a distinguished point, the root. This allows for the definition of a smallest "organic" Riemann surface which contains a prescribed set of "broken line paths". Universal covering spaces and so called convolution products of rooted Riemann surfaces are special cases of organic Riemann surfaces.

In section 5, we come to the main subject of computable analytic functions. In [vdH05a], a first definition was proposed. Roughly speaking, the idea was to see a computable analytic function as an instance f of an abstract data type $\mathbb{A}^{\mathrm{lcom}}$, with methods for computing

- The coefficients of f.
- A lower bound $\underline{r}_f$ for the radius of convergence of f.
- An upper bound $\llbracket f \rrbracket_\rho$ for $|f|$ on any disk of radius $\rho < \underline{r}_f$.
- The analytic continuation $f_{+\delta} \in \mathbb{A}^{\mathrm{lcom}}$ of f from 0 to δ, with $|\delta| < \underline{r}_f$.

This point of view is very natural from a computational point of view if we want to solve a differential or more general functional equation, since it is often possible to locally solve such equations. However, the computed bounds are usually not sharp, so we need some additional global conditions in order to ensure that analytic continuation can be carried out effectively at all points where the solutions are defined.

Now the more systematic theory of computable Riemann surfaces of this paper makes it possible to directly define the concept of a computable analytic function on a given computable Riemann surface. Although this makes definitions easier, one still has to show how to construct the Riemann surface of a computable analytic function. Using the results from section 4, we will do this for many classical operations, like $+$, $-$, $\times$, ∂, $\int$, $\exp$, $\log$, $\circ$, algebraic and differential equations, convolution products, etc. Especially in the case of convolution products, the global knowledge of an underlying Riemann surface is very important. What is more, we will show that it is possible to construct the Riemann surfaces incrementally, on explicit demand by the user. Also, whereas all underlying Riemann surfaces from [vdH05a] were simply connected, the present theory enables us to identify certain branches where the function takes identical values. Nevertheless, the local approach from [vdH05a] remains useful, because any "locally computable analytic function" induces a natural "globally computable analytic function" (see theorem 5.7).

During the implementation of some of the algorithms from [vdH05a] in our MMXLIB library, it turned out that bad bounds r_f and $\llbracket f \rrbracket_\rho$ could lead to extremely inefficient algorithms. Therefore, it is essential to have algorithms for the efficient computation of accurate bounds. In section 6, we will study this problem in a systematic way. Our *leitmotiv* is to work with truncated power series expansions at an order n with a bound for the remainder. On the one hand, we will study how such expansions and bounds can be computed efficiently and accurately (sections 6.3 and 6.4). On the other hand, we will show how to use them for computing the absolute value of the smallest zero of an analytic function (section 6.1) and for computing extremal values on a compact disk (section 6.2). Several of the ideas behind our algorithms already occur in the literature about Taylor models and polynomial root finding. However, the context is a bit different, so our exposition may have some interest for its own sake.

For the sake of simplicity, we have limited ourselves to the study of univariate analytic functions. It should be possible to generalize to the multivariate case along the same lines. The main extra difficulty we foresee is integration, because it requires an automatic algorithm for the deformation of paths. Nevertheless, in sections 4.8 and 5.5, we study convolution products, and a similar approach might be used for integration. Some of the algorithms in this paper have been implemented in the MMXLIB library. However, our implementation is still quite unstable and work is in progress to include the ideas from the present paper.

2. Computable real and complex numbers

2.1. Computable functions and relations on effective sets

We assume that the reader is familiar with basic notions of the theory of Turing machines. We recall that a Turing machine T computes a function $f_T\colon \mathbb{N} \to \mathbb{N} \cup \{\mathbf{fail}\}$, where $f_T(n) = \mathbf{fail}$ if the Turing machine does not halt on the input n. A function $f\colon \mathbb{N} \to \mathbb{N}$ is said to be *computable* if $f = f_T$ for some Turing machine T. A subset A of $\mathbb{N}$ is said to be *recursively enumerable*, or shortly *enumerable*, if $A = \varnothing$ or if there exists a Turing machine T with $A = \operatorname{im} f_T$. We say that A is *computable* if both A and $\mathbb{N} \setminus A$ are enumerable. Denoting by $\mathbb{T}$ the set of Turing machines, there exists a bijection $\chi\colon \mathbb{N} \rightharpoonup \mathbb{T}$, whose inverse encodes each Turing machine by a unique natural number.

More generally, an *encoding* (or *effective representation*) of a set A is a partial surjective function $\chi = \chi_A\colon \mathbb{N} \rightharpoonup A$, which is not necessarily injective. In that case, we call A (or more precisely the pair (A, χ)) an *effective* set. If $\operatorname{dom} \chi_A$ is computable or enumerable, then we call A an *abstract computable* resp. *enumerable* set. For instance, the set of Turing machines which halt on all inputs is an effective set, but not an abstract computable set, because of the halting problem. If A and B are effective sets, then so is $A \times B$, for the encoding $\chi_{A\times B}(\varphi(i,j)) = (\chi_A(i), \chi_B(j))$, where $\varphi\colon \mathbb{N}^2 \to \mathbb{N}; (i,j) \mapsto (i+j)^2 + i$. By induction, A^n is an effective set for each $n \in \mathbb{N}$. Many other classical sets, like finite sequences or trees over an effective set admit straightforward encodings, which will not be detailed in what follows.

A function $f\colon A \to B$ between two effective sets A and B is said to be *computable* if there exists a Turing machine $T \in \mathbb{T}$ such that $f(\chi_A(i)) = \chi_B(f_T(i))$ for all $i \in \operatorname{dom} \chi_A$. In that case, each n with $T = \chi_{\mathbb{T}}(n)$ provides an encoding for f, and we denote by $\mathcal{F}^{\mathrm{com}}(A, B)$ the effective set of all computable functions from A to B. A partial function $f\colon A \rightharpoonup B$ is said to be *computable* if there exists a Turing machine $T \in \mathbb{T}$ with $f(\chi_A(i)) = \chi_B(f_T(i))$ for all $i \in \chi_A^{-1}(\operatorname{dom} f)$.

Sometimes, it is convenient to allow for generalized encodings $\chi^E = \chi^E_A : E \to A$, where E is another encoded set. Indeed, in that case, the composition $\chi_A = \chi^E_A \circ \chi_E$ yields an encoding in the usual sense. For instance, $\chi_{\mathcal{F}^{\mathrm{com}}(A,B)} = \chi^{\mathbb{T}} \circ \chi_{\mathbb{T}}$, where $\chi^{\mathbb{T}}$ encodes each function $f \in \mathcal{F}^{\mathrm{com}}(A, B)$ by the Turing machine which computes it. Given $a = \chi^E_A(c)$, we will write $c = \check{a}$ and $a = \hat{c}$. To each object $a \in A$, given by its encoding $a = \chi_A(n)$ with $n \in \mathbb{N}$, we may naturally associate its representation $\check{a} = \chi_E(n)$ in E. However, this association does not lead to a mapping $\check{}: A \to E$, since we do not necessarily have $\chi_A(m) = \chi_A(n) \Rightarrow \chi_E(m) = \chi_E(n)$. In particular, in order to implement a computable function $f : A \to B$ via a computable function $\check{f} : E \to B$, using $f(a) = \check{f}(\check{a})$, one has to make sure that $\check{f}(c_1) = \check{f}(c_2)$ whenever $\hat{c}_1 = \hat{c}_2$.

An n-ary relation $\mathcal{R} \subseteq A^n$ on an effective set A is said to be *computable*, if there exists a computable subset $\check{\mathcal{R}}$ of $\mathbb{N}$, with $\chi^{-1}_{A^n}(\mathcal{R}) = \check{\mathcal{R}} \cap \operatorname{dom} \chi_{A^n}$. Equivalently, we may require the existence of a computable function $\check{\mathcal{R}} : A^n \to \{0, 1\}$ with $\mathcal{R}(a) \Leftrightarrow \check{\mathcal{R}}(a) = 1$ for all $a \in A^n$. Similarly, $\mathcal{R} \subseteq A^n$ is *enumerable*, if there exists an enumerable subset $\check{\mathcal{R}}$ of $\mathbb{N}$, with $\chi^{-1}_{A_n}(\mathcal{R}) = \check{\mathcal{R}} \cap \operatorname{dom} \chi_{A^n}$. This is equivalent to the existence of a computable function $\check{\mathcal{R}} : A^n \to \{0, 1\}^{\mathrm{ult}}$ with $\mathcal{R}(a) \Leftrightarrow \check{\mathcal{R}}(a) = 1$ for all $a \in A^n$. Here $\{0, 1\}^{\mathrm{ult}}$ denotes the set of increasing computable functions $f : \mathbb{N} \to \{0, 1\}$, divided by the equivalence relation $\sim$ with $f \sim g \Leftrightarrow \lim_{n\to\infty} f_n = \lim_{n\to\infty} g_n$. Notice that $\{0, 1\}^{\mathrm{ult}}$ and $\{0, 1\}$ are equal as sets, but *not* as effective sets. A computable function $\check{\mathcal{R}} : A^n \to \{0, 1\}^{\mathrm{ult}}$ will be called an *ultimate test*. Notice that the equality relation on an effective set A is not necessarily computable or enumerable, even if A is an abstract computable set.

Since a subset $B \subseteq A$ is also a unary relation on A, the above definition in particular yields the notions of computable and enumerable subsets of A. We also define B to be a *sequentially enumerable* subset of A if $B = \varnothing$ or if there exists a computable function $\check{B} : \mathbb{N} \to A$ with $B = \operatorname{im} \check{B}$. Similarly, we say that B is *sequentially computable* if both B and $A \setminus B$ are sequentially enumerable. If B is sequentially enumerable and A admits a computable equality test, then B is enumerable. If B is enumerable and A is an abstract enumerable set, then B is sequentially enumerable. If B is sequentially computable, then A is an abstract enumerable set.

There are several other interesting notions which deserve further study, but which will not be used in what follows. For instance, we may define a subset B of an effective set A to be *pseudo-computable*, if there exists a computable function $\check{B} : A \to \mathbb{N}^{\mathrm{ult}}$ with $B = \{x \in A : \check{B}(x) = +\infty\}$, where $\mathbb{N}^{\mathrm{ult}}$ is defined similarly as $\{0, 1\}^{\mathrm{ult}}$. For instance, given a Turing machine $T \in \mathbb{T}$, the set $\{x \in \mathbb{N} : f_T(x) \neq \mathbf{fail}\}$ is a pseudo-computable subset of $\mathbb{N}$.

2.2. Computable real numbers

Let $\mathbb{R}^{\mathrm{dig}} = \mathbb{Z}\, 2^{\mathbb{Z}}$ be the set of *digital* or *dyadic* numbers. Given an ordered ring R, we denote $R^{>} = \{x \in R : x > 0\}$, $R^{\geqslant} = \{x \in R : x \geqslant 0\}$, etc. Given $x \in \mathbb{R}$ and $\varepsilon \in \mathbb{R}^{\mathrm{dig},>}$, we say that $x' \in \mathbb{R}^{\mathrm{dig}}$ is an *ε-approximation* of x if $|x' - x| < \varepsilon$. An *approximator* for x is a computable function $\check{x} : \mathcal{F}^{\mathrm{com}}(\mathbb{R}^{\mathrm{dig},>}, \mathbb{R}^{\mathrm{dig}})$ which sends $\varepsilon \in \mathbb{R}^{\mathrm{dig},>}$ to an ε-approximation of x. If x admits such an approximator, then we call x a *computable real number* and encode x by $\check{x}$. We denote by $\mathbb{R}^{\mathrm{app}}$ the set of approximators and by $\mathbb{R}^{\mathrm{com}} \subseteq \mathbb{R}$ the effective set of computable real numbers. Given $i, j \in \mathbb{N}$, both the problems of testing whether $\chi_{\mathbb{R}^{\mathrm{app}}}(i) = \chi_{\mathbb{R}^{\mathrm{app}}}(j)$ resp. $\chi_{\mathbb{R}^{\mathrm{com}}}(i) = \chi_{R^{\mathrm{com}}}(j)$ are undecidable.

The usual topologies on $\mathbb{R}$ and $\mathbb{R}^n$ naturally induce topologies on $\mathbb{R}^{\mathrm{com}}$ and $(\mathbb{R}^{\mathrm{com}})^n$. Given an open subset Ω of $(\mathbb{R}^{\mathrm{com}})^n$, an element of $\mathcal{F}^{\mathrm{com}}(\Omega, \mathbb{R}^{\mathrm{com}})$ is called a *computable real function*. Notice that such a function admits a natural encoding by an element $\check{f} \in \mathcal{F}^{\mathrm{com}}(\check{\Omega}, \mathbb{R}^{\mathrm{app}})$, where $\check{\Omega} = \{\check{x} \in (\mathbb{R}^{\mathrm{app}})^n : \hat{\check{x}} = (\hat{\check{x}}_1, \ldots, \hat{\check{x}}_n) \in \Omega\}$. Many classical

functions like $+$, $-$, $\times$, exp , log , max, min are easily seen to be computable. It can be shown (see [Grz55, Grz57, Wei00] and theorem 2.3 below) that a computable real function is necessarily continuous. Consequently, the step and stair functions are not computable. Intuitively speaking, this stems from the fact that the sign function cannot be computed effectively for computable real numbers.

It is convenient to express part of the semantics of computations with computable real numbers by providing a signature for the available operations. For instance, the class $\mathbb{R}^{\text{app}}$ comes with two main functions

$$\begin{array}{rrcl} \mathsf{approx}: & \mathbb{R}^{\text{app}} \times \mathbb{R}^{\text{dig},>} & \to & \mathbb{R}^{\text{dig}} \\ \chi: & \mathbb{R}^{\text{app}} & \to & \mathbb{R}^{\text{com}} \end{array}$$

Similarly, the class $\mathbb{R}^{\text{com}}$ provides operations

$$\begin{array}{rrcl} \iota: & \mathbb{Q} & \to & \mathbb{R}^{\text{com}} \\ +,-,\times: & \mathbb{R}^{\text{com}} \times \mathbb{R}^{\text{com}} & \to & \mathbb{R}^{\text{com}} \\ /: & \mathbb{R}^{\text{com}} \times \mathbb{R}^{\text{com},\neq} & \to & \mathbb{R}^{\text{com}} \\ \min, \max: & \mathbb{R}^{\text{com}} \times \mathbb{R}^{\text{com}} & \to & \mathbb{R}^{\text{com}} \\ \exp, \sin, \cos: & \mathbb{R}^{\text{com}} & \to & \mathbb{R}^{\text{com}} \\ \log: & \mathbb{R}^{\text{com},>} & \to & \mathbb{R}^{\text{com}} \\ \vdots & & & \end{array}$$

However, we take care *not* to provide functions for comparisons.

2.3. Left and right computable real numbers

There exist many equivalent definitions for computable real numbers and several alternative encodings [Wei00, Chapter 4]. A particularly interesting alternative encoding is to define an *approximator* (or two-sided approximator) of $x \in \mathbb{R}$ to be a computable function $\mathbb{N} \to (\mathbb{R}^{\text{dig}})^2; k \mapsto x_k = (\underline{x}_k, \overline{x}_k)$ with

$$\underline{x}_1 \leqslant \underline{x}_2 \leqslant \cdots \leqslant x \leqslant \cdots \leqslant \overline{x}_2 \leqslant \overline{x}_1$$

and $\lim_{k\to\infty} \underline{x}_k = \lim_{k\to\infty} \overline{x}_k = x$. This definition admits two variants: a *left approximator* (resp. *right approximator*) of $x \in \mathbb{R}$ is a computable increasing (resp. decreasing) function $\mathbb{N} \to \mathbb{R}^{\text{dig}}; k \mapsto x_k$, with $x = \lim_{k\to\infty} x_k$. A real number is said to be *left computable* (resp. *right computable*) if it admits a left (resp. right) approximator.

Intuitively speaking, a left (resp. right) computable real number corresponds to a computable lower (resp. upper) bound. Indeed, in what follows, it will frequently occur that we can compute sharper and sharper lower or upper bounds for certain real numbers, without being able to compute an optimal bound. We denote by $\mathbb{R}^{\text{lapp}}$, $\mathbb{R}^{\text{rapp}}$, $\mathbb{R}^{\text{lcom}}$ and $\mathbb{R}^{\text{rcom}}$ the left and right analogues of $\mathbb{R}^{\text{app}}$ and $\mathbb{R}^{\text{com}}$.

Remark 2.1. The above definitions of left, right and two-sided approximators naturally extend to the case of sequences in the set $\overline{\mathbb{R}^{\text{dig}}} = \{-\infty\} \cup \mathbb{R}^{\text{dig}} \cup \{+\infty\}$ of extended digital numbers. This leads to natural counterparts $\overline{\mathbb{R}}^{\text{app}}$, $\overline{\mathbb{R}}^{\text{com}}$, $\overline{\mathbb{R}}^{\text{lapp}}$, etc.

Remark 2.2. For actual implementations, it is a good idea to let the index k of approximators $k \mapsto x_k$ correspond to the estimated cost of the computation of x_k (see also [vdH06b]). We also notice that left, right and two-sided approximators can be implemented by a common class `real` with a method `approximate`, which returns a bounding interval $\boldsymbol{x}_k \ni x$ as a function of k. In the case of left (resp. right) approximators, we would have $\boldsymbol{x}_k = [x_k, +\infty]$ (resp. $x_k = [-\infty, x_k]$).

Let Ω be an open subset of $\mathbb{R}^n$ or $(\mathbb{R}^{\mathrm{com}})^n$. A function $f: \Omega \to \mathbb{R}$ is said to be *lower continuous* (resp. *upper continuous*), if for every $x \in \Omega$ and every $y' < f(x)$ (resp. $y' > f(x)$), there exists a neighbourhood V of x, such that $y' < f(x')$ (resp. $y' > f(x')$) for all $x' \in V$. We have [Grz55, Grz57, Wei00]:

THEOREM 2.3. *Let Ω be an open subset of $(\mathbb{R}^{\mathrm{com}})^n$. Then*

a) Any $f \in \mathcal{F}^{\mathrm{com}}(\Omega, \mathbb{R}^{\mathrm{com}})$ is continuous.

b) Any $f \in \mathcal{F}^{\mathrm{com}}(\Omega, \mathbb{R}^{\mathrm{lcom}})$ is lower continuous.

c) Any $f \in \mathcal{F}^{\mathrm{com}}(\Omega, \mathbb{R}^{\mathrm{rcom}})$ is upper continuous.

Proof. We will prove (b); the other two assertions are proved in a similar way. The function f admits an encoding $\check{f} \in \mathcal{F}^{\mathrm{com}}(\check{\Omega}, \mathbb{R}^{\mathrm{lcom}})$. Let $x \in \Omega$ with approximator

$$\check{x}: k \mapsto x_k = ((\underline{x}_{k,1}, \overline{x}_{k,1}), ..., (\underline{x}_{k,n}, \overline{x}_{k,n})).$$

Let $\check{y}: k \mapsto y_k$ be a left approximator for $y = f(x)$. Given $y' < y$, there exists a $q \in \mathbb{N}$ with $y_q > y'$. Now the computation of $y_0, ..., y_q$ by $\check{f}$ only depends on $x_0, ..., x_p$ for some finite $p \in \mathbb{N}$. Increasing p if necessary, we may assume without loss of generality that

$$V = R_{x_p} = \{(v_1, ..., v_n) \in (\mathbb{R}^{\mathrm{com}})^n : \underline{x}_{p,1} < v_1 < \overline{x}_{p,1}, ..., \underline{x}_{p,n} < v_n < \overline{x}_{p,n}\} \subseteq \Omega.$$

Let $x' \in V$, with approximator $\check{x}': k \mapsto x'_k$. For a certain $p' \geqslant p$, we have $R_{x'_{p'}} \subseteq V$. Now consider the alternative approximator $\check{x}'': k \mapsto x''_k$ of x' with $x''_k = x_k$ for $k \leqslant p$ and $x''_k = x'_{k+p'-p}$ otherwise. Then, by construction, $\check{y}'' = \check{f}(\check{x}'') : k \mapsto y''_k$ satisfies $y''_0 = y_0$, ..., $y''_q = y_q$. We conclude that $f(x') = \lim_{k \to \infty} y''_k \geqslant y_q > y'$. □

The "lower step function" σ, defined by $\sigma(x) = 0$ if $x < 0$ and $\sigma(x) = 1$ otherwise, is lower computable in the sense that $\sigma \in \mathcal{F}^{\mathrm{com}}(\mathbb{R}^{\mathrm{com}}, \mathbb{R}^{\mathrm{lcom}})$. Indeed, given $\check{x}: n \mapsto (\underline{x}_n, \overline{x}_n)$, we may take $\check{y} = \check{\sigma}(\check{x}): n \mapsto \sigma(\underline{x}_n)$. Similarly, the function $x \mapsto \lfloor x \rfloor$ is lower computable, while $x \mapsto \lceil x \rceil$ is upper computable. In particular, this shows that $\mathcal{F}^{\mathrm{com}}(\mathbb{R}^{\mathrm{com}}, \mathbb{R}^{\mathrm{lcom}}) \supsetneq \mathcal{F}^{\mathrm{com}}(\mathbb{R}^{\mathrm{com}}, \mathbb{R}^{\mathrm{com}}) \subsetneq \mathcal{F}^{\mathrm{com}}(\mathbb{R}^{\mathrm{com}}, \mathbb{R}^{\mathrm{rcom}})$. Besides the projections

$$\begin{aligned} \mathsf{left}: &\ \mathbb{R}^{\mathrm{com}} \to \mathbb{R}^{\mathrm{lcom}} \\ \mathsf{right}: &\ \mathbb{R}^{\mathrm{com}} \to \mathbb{R}^{\mathrm{rcom}} \end{aligned}$$

typical lower computable functions on $\mathbb{R}^{\mathrm{lcom}}$ are:

$$\begin{aligned} +: &\ \mathbb{R}^{\mathrm{lcom}} \times \mathbb{R}^{\mathrm{lcom}} \to \mathbb{R}^{\mathrm{lcom}} \\ \times: &\ \mathbb{R}^{\mathrm{lcom},\geqslant} \times \mathbb{R}^{\mathrm{lcom},\geqslant} \to \mathbb{R}^{\mathrm{lcom},\geqslant} \\ \min, \max: &\ \mathbb{R}^{\mathrm{lcom}} \times \mathbb{R}^{\mathrm{lcom}} \to \mathbb{R}^{\mathrm{lcom}} \\ \exp: &\ \mathbb{R}^{\mathrm{lcom}} \to \mathbb{R}^{\mathrm{lcom}} \\ \log: &\ \mathbb{R}^{\mathrm{lcom},>} \to \mathbb{R}^{\mathrm{lcom}} \\ \sigma, \lfloor \cdot \rfloor: &\ \mathbb{R}^{\mathrm{lcom}} \to \mathbb{R}^{\mathrm{lcom}} \end{aligned}$$

Here the dot in $\lfloor \cdot \rfloor$ indicates the argument of the function $x \mapsto \lfloor x \rfloor$. Left computable numbers are turned into right computable numbers and *vice versa* by the following operations:

$$\begin{aligned} -: &\ \mathbb{R}^{\mathrm{lcom}} \to \mathbb{R}^{\mathrm{rcom}} \\ 1/\cdot: &\ \mathbb{R}^{\mathrm{lcom},>} \to \mathbb{R}^{\mathrm{rcom},>} \end{aligned}$$

More generally, increasing computable real functions induce both increasing lower and upper computable real functions, while decreasing computable real functions turn left computable real numbers into right computable real numbers and *vice versa*.

2.4. Computable complex numbers

The complexification $\mathbb{C}^{\mathrm{com}} = \mathbb{R}^{\mathrm{com}}[\mathrm{i}] = \mathbb{R}^{\mathrm{com}} \oplus \mathbb{R}^{\mathrm{com}}\,\mathrm{i}$ of $\mathbb{R}^{\mathrm{com}}$ provides a natural definition for the set of *computable complex numbers*. Typical operations on $\mathbb{C}^{\mathrm{com}}$ include

$$\begin{array}{rrcl}
\mathsf{complex,\,polar:} & \mathbb{R}^{\mathrm{com}} \times \mathbb{R}^{\mathrm{com}} & \to & \mathbb{C}^{\mathrm{com}} \\
\Re, \Im: & \mathbb{C}^{\mathrm{com}} & \to & \mathbb{R}^{\mathrm{com}} \\
\mathrm{abs}: & \mathbb{C}^{\mathrm{com}} & \to & \mathbb{R}^{\mathrm{com},\geqslant} \\
\mathrm{arg}: & \mathbb{C}^{\mathrm{com}} \setminus \mathbb{R}^{\mathrm{com},\leqslant} & \to & (-\pi, \pi)^{\mathrm{com}} \\
+, -, \times: & \mathbb{C}^{\mathrm{com}} \times \mathbb{C}^{\mathrm{com}} & \to & \mathbb{C}^{\mathrm{com}} \\
/: & \mathbb{C}^{\mathrm{com}} \times \mathbb{C}^{\mathrm{com},\neq} & \to & \mathbb{C}^{\mathrm{com}} \\
\exp, \sin, \cos: & \mathbb{C}^{\mathrm{com}} & \to & \mathbb{C}^{\mathrm{com}} \\
\log: & \mathbb{C}^{\mathrm{com}} \setminus \mathbb{R}^{\mathrm{com},\leqslant} & \to & \mathbb{R}^{\mathrm{com},>} + (-\pi, \pi)^{\mathrm{com}}\,\mathrm{i}
\end{array}$$

The complexification $\mathbb{C}^{\mathrm{app}} = \mathbb{R}^{\mathrm{app}}[\mathrm{i}]$ of $\mathbb{R}^{\mathrm{app}}$ also provides a natural encoding for $\mathbb{C}^{\mathrm{com}}$ and, setting $\mathbb{C}^{\mathrm{dig}} = \mathbb{R}^{\mathrm{dig}}[\mathrm{i}]$, the approximation function for numbers in $\mathbb{R}^{\mathrm{app}}$ extends to

$$\mathsf{approx:}\ \ \mathbb{C}^{\mathrm{app}} \times \mathbb{R}^{\mathrm{dig},>} \ \to\ \mathbb{C}^{\mathrm{dig}}$$

Clearly, functions like arg, $\log$, $\sqrt{\cdot}$, etc. can only be defined on simply connected subsets of $\mathbb{C}$. On the other hand, $\mathbb{C}^{\mathrm{com}}$ is *effectively algebraically closed* in the sense that there exists an algorithm which takes a polynomial $P \in \mathbb{C}^{\mathrm{com}}[z]^{\neq}$ of degree d on input and which returns its set of d roots in $\mathbb{C}^{\mathrm{com}}$.

2.5. Semi-computable numbers

For many applications, the absence of computable comparisons for computable real or complex numbers can be a big problem. One solution to this problem is to systematically consider all possible answers of zero tests or sign computations and to use these answers as hypotheses during subsequent tests. For instance, if we assume that $x \geqslant 2$, then a subsequent test $x^2 - x \geqslant 1$ should return **true**.

The above approach can be formalized as follows. A *system of real constraints* is a pair $(x, \epsilon) = ((x_1, ..., x_l), (\epsilon_1, ..., \epsilon_l))$ with $x_i \in \mathbb{R}^{\mathrm{com}}$ and $\epsilon_i \in \{-1, 0, 1\}$ for $i = 1, ..., l$. We say that (x, ϵ) is *satisfied* if $\operatorname{sign} x_i = \epsilon_i$ for $i = 1, ..., l$. We denote by $\mathbf{\Sigma}$ the set of systems of real constraints. A *semi-computable real number* is encoded by a computable function $\check{x}: S_{\check{x}} \to \mathbb{R}^{\mathrm{com}}$, where $S_{\check{x}}$ is a finite subset of $\mathbf{\Sigma}$ such that at least one element of $S_{\check{x}}$ is satisfied and $\check{x}(\Sigma) = \check{x}(\Sigma')$ whenever both Σ and Σ' are satisfied. We denote by $\mathbb{R}^{\mathrm{scom}}$ the set of semi-computable real numbers. A *semi-computable function* is a function $f: \mathbb{R}^{\mathrm{com}} \to \mathbb{R}^{\mathrm{scom}}$. Such a function naturally induces a function $F: \mathbb{R}^{\mathrm{scom}} \to \mathbb{R}^{\mathrm{scom}}$. Indeed, given $x \in \mathbb{R}^{\mathrm{scom}}$, encoded by $\check{x}: S_{\check{x}} \to \mathbb{R}^{\mathrm{com}}$, we may take $S_{\check{F}(\check{x})} = \bigcup_{\Sigma \in S_{\check{x}}} S_{\check{f}(\check{x}(\Sigma))}$ and $\check{F}(\check{x})(\Sigma') = \check{f}(\check{x}(\Sigma))(\Sigma')$, whenever $\Sigma' \in S_{\check{f}(\check{x}(\Sigma))}$.

Example 2.4. The step function $f: x \mapsto \lfloor x \rfloor$ is semi-computable. Indeed, given $x \in \mathbb{R}^{\mathrm{com}}$, we first compute an ε-approximation $\tilde{x} \in \mathbb{R}^{\mathrm{dig}}$ of x with $\varepsilon \ll 1$ (e.g. $\varepsilon = 2^{-32}$) and $n = \lfloor \tilde{x} + \varepsilon \rfloor$. If $\lfloor \tilde{x} - \varepsilon \rfloor = n$, then we let

$$S_{\check{f}(x)} = \{\Sigma\} = \{((), ())\}$$

and take $\check{f}(x): S_{\check{f}(x)} \to \mathbb{R}^{\mathrm{com}}; \Sigma \to n$. Otherwise, we let

$$S_{\check{f}(x)} = \{\Sigma_{-1}, \Sigma_0, \Sigma_1\} = \{((x-n),(-1)), ((x-n),(0)), ((x-n),(1))\}$$

and take $\check{f}(x): S_{\check{f}(x)} \to \mathbb{R}^{\mathrm{com}}$ with $\check{f}(x)(\Sigma_{-1}) = n-1$ and $\check{f}(x)(\Sigma_0) = \check{f}(x)(\Sigma_1) = n$.

From a practical point of view, computations with semi-computable numbers can be implemented using non-deterministic evaluation and we point to the similarity with the computation with parameterized expressions [vdH97, Chapter 8]. Each branch of the non-deterministic computation process comes with a system $\Sigma = ((x_1, ..., x_l), (\epsilon_1, ..., \epsilon_l))$ of real constraints in $\mathbb{\Sigma}$. A constraint checker is used in order to eliminate branches for which Σ is contradictory.

In many applications, the numbers $x_1, ...x_l$ belong to a polynomial algebra $\mathbb{Q}[y_1, ..., y_n]$ and one may use classical algorithms from real algebraic geometry to check the consistency of Σ [BPR03]. Modulo further progress in automatic proofs of identities [Ric92, Zei90, vdH02a], we hope that more and more powerful constraint checkers will be constructed for increasingly general classes of constants (like algebraic exp-log expressions in $y_1, ..., y_n$). This would allow for the automatic elimination of a large number of inconsistent branches. Notice also that it is recommended to spend a roughly equivalent time in trying to prove and disprove constraints. Of course, proving $x > 0$ is easy, since it suffices to find a non zero digit of x.

As in the case of computations with parameterized expressions, many algorithms for computable real numbers naturally generalize to semi-computable real numbers. This is due to the fact that all numbers involved often belong to a fixed polynomial algebra $\mathbb{Q}[y_1, ..., y_n]$, in which the Noetherianity of this algebra may be used in termination proofs. We refer to [vdH97] for examples.

Remark 2.5. In our definition of systems of real constraints, we have considered sign conditions on computable real numbers. The same construction may be applied to more general types of constraints, like $x_i \in \Omega_{k_i}$, for a certain number $\Omega_1, \Omega_2, ...$ of fixed subsets of the real numbers. However, we have not yet found any practical use for such a generalization.

3. Computable Riemann surfaces

A classical *Riemann surface* (above $\mathbb{C}$) is a topological space $\mathcal{R}$, together with a projection $\pi: \mathcal{R} \to \mathbb{C}$, so that every $x \in \mathcal{R}$ admits a neighbourhood V for which $\pi|_V$ is a homeomorphism of V on an open ball of $\mathbb{C}$. A Riemann surface $\bar{\mathcal{R}} = \mathcal{R} \amalg \partial\mathcal{R}$ with border $\partial\bar{\mathcal{R}} = \partial\mathcal{R}$ is defined similarly, except that each $x \in \partial\bar{\mathcal{R}}$ now admits a neighbourhood V for which $\pi_{|V}$ is a homeomorphism of V on a subset of $\mathbb{C}$ which is homeomorphic to $\{z \in \mathbb{C}: \Re z \geqslant 0\}$. A classical *covering* is a local homeomorphism $\varphi: \mathcal{R} \to \mathcal{S}$ between two Riemann surfaces, which commutes with the projections, i.e. $\pi_{\mathcal{S}} \circ \varphi = \pi_{\mathcal{R}}$. Throughout this paper, coverings are *not* required to be surjective.

3.1. Digital Riemann surfaces

An *encoding of a digital Riemann surface* is a tuple $\check{\mathcal{R}} = (\lambda, A, \pi, \boxplus)$, where A is a finite set of *nodes*, $\lambda \in 2^{\mathbb{Z}}$ a *scale*, $\pi: A \to \mathbb{Z}[\mathrm{i}]$ a *projection* and $\boxplus \subseteq A^2$ a symmetric *adjacency relation*, such that

DR1. If $a \boxplus b$, then $\pi(a) - \pi(b) \in \{1, -1, \mathrm{i}, -\mathrm{i}\}$.

DR2. If $a \boxplus b$ and $a \boxplus b'$ are such that $\pi(b) = \pi(b')$, then $b = b'$.

DR3. Let $a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}$ be such that $\pi(a_{\delta,\varepsilon}) = \pi(a_{0,0}) + \delta + \varepsilon\,\mathrm{i}$ for $\delta, \varepsilon \in \{0,1\}$ and such that three relations among $a_{0,0}\boxplus a_{0,1}$, $a_{0,0}\boxplus a_{1,0}$, $a_{0,1}\boxplus a_{1,1}$ and $a_{1,0}\boxplus a_{1,1}$ hold. Then the fourth relation holds as well.

The conditions **DR2** and **DR3** are illustrated in figure 3.1 below. In the case when a, b, $c, d \in A$ with pairwise distinct projections $\pi(a), \pi(b), \pi(c)$ and $\pi(d)$ satisfy $a\boxplus b\boxplus c\boxplus d\boxplus a$, then we will also write ${}^{a}_{d}\boxplus^{b}_{c}$. Notice that ${}^{a}_{d}\boxplus^{b}_{c} \Leftrightarrow {}^{a}_{b}\boxplus^{d}_{c} \Leftrightarrow {}^{d}_{c}\boxplus^{a}_{b}$.

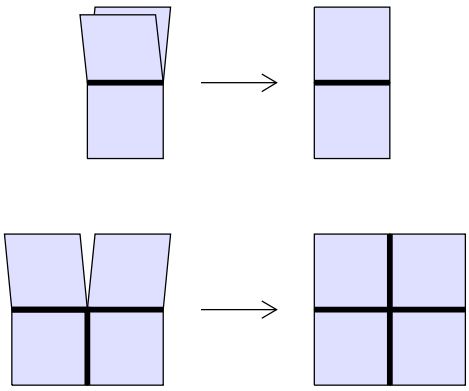

Figure 3.1. Illustration of the axioms **DR2** (top) and **DR3** (bottom) for digital Riemann surfaces. When regarding the left hand sides as digital Riemann pastings, the right hand sides also correspond to their normalizations.

Let us show how to associate a Riemann surface $\mathcal{R}$ in the classical sense to an encoding $\breve{\mathcal{R}} = (\lambda,\ A,\ \pi,\ \boxplus)$ as above. To each $z \in \mathbb{Z}[\mathrm{i}]$, we may associate a compact square $\bar{\mathcal{Q}}_{z,\lambda}$ by

$$\bar{\mathcal{Q}}_{z,\lambda} = \lambda\,(z + [0,1] + [0,1]\,\mathrm{i}).$$

We now consider the topological space

$$\breve{\mathcal{R}} = \coprod_{a \in A} \breve{\mathcal{Q}}_a,$$

where $\breve{\mathcal{Q}}_a$ is a copy of $\bar{\mathcal{Q}}_{\pi(a),\lambda}$ for each $a \in A$. Whenever $a\boxplus b$, the squares $\bar{\mathcal{Q}}_{\pi(a),\lambda}$ and $\bar{\mathcal{Q}}_{\pi(b),\lambda}$ admit a common edge in $\mathbb{C}$. Gluing the corresponding copies $\breve{\mathcal{Q}}_a$ and $\breve{\mathcal{Q}}_b$ together in $\breve{\mathcal{R}}$ according to this edge determines a new topological space

$$\bar{\mathcal{R}} = \breve{\mathcal{R}}/\!\sim.$$

The space $\bar{\mathcal{R}}$ is a Riemann surface with a border, whose projection on $\mathbb{C}$ is naturally determined by the projections of the $\breve{\mathcal{Q}}_a$ on $\mathbb{C}$. Indeed, **DR2** (resp. **DR3**) implies that points on the edges (resp. vertices) of the $\breve{\mathcal{Q}}_a/\!\sim$ are either in the interior of $\bar{\mathcal{R}}$ or on its border. The interior $\mathcal{R}$ of $\bar{\mathcal{R}}$, endowed with its natural projection on $\mathbb{C}$, is a Riemann surface in the classical sense; we call it the *digital Riemann surface* associated to $\breve{\mathcal{R}}$. It will be convenient to write $\bar{\mathcal{Q}}_a = \breve{\mathcal{Q}}_a/\!\sim$ and denote the interior of $\bar{\mathcal{Q}}_a$ by $\mathcal{Q}_a$. More generally, if $B \subseteq A$, then we write $\bar{\mathcal{Q}}_B = \bigcup_{a \in B} \bar{\mathcal{Q}}_a$ and $\mathcal{Q}_B$ for its interior.

Example 3.1. If the mapping $\pi\colon A \to \mathbb{Z}[\mathrm{i}]$ is injective, then the induced projection $\pi\colon \mathcal{R} \to \mathbb{C}$ is a homeomorphism on its image. In that case, we will identify $\mathcal{R}$ with the subset $\pi(\mathcal{R})$ of $\mathbb{C}$, and call $\mathcal{R}$ a *digital subset* of $\mathbb{C}$. Conversely, any $\lambda \in 2^{\mathbb{Z}}$ together with a finite subset $A \subseteq \mathbb{Z}[\mathrm{i}]$ determines a natural digital subset $\mathcal{R} \subseteq \mathbb{C}$, encoded by $\breve{\mathcal{R}} = (\lambda, A, \mathrm{Id}, \boxplus)$ with $a\boxplus b \Leftrightarrow a - b \in \{\pm 1, \pm\mathrm{i}\}$.

Example 3.2. One of the simplest examples of a digital Riemann surface $\mathcal{R}$ for which π is *not* injective is shown in figure 3.2. Formally speaking, this surface is encoded by

$$\begin{aligned}
\lambda &= 1\\
A &= \{a_{1,0}, a_{1,1}, a_{0,1}, a_{-1,1}, a_{-1,0}, a_{-1,-1}, a_{0,-1}, a_{1,-1}, b_{1,0}\}\\
\pi(a_{x,y}) &= x + y\,\mathrm{i}\\
\pi(b_{1,0}) &= 1\\
a_{x,y} \square\!\!\square a_{x',y'} &\Leftrightarrow |x'-x| + |y'-y| = 1 \wedge \{a_{x,y}, a_{x',y'}\} \neq \{a_{1,0}, a_{1,-1}\}\\
a_{x,y} \square\!\!\square b_{1,0} &\Leftrightarrow a_{x,y} = a_{1,-1}.
\end{aligned}$$

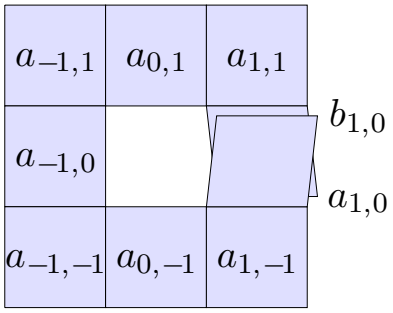

Figure 3.2. Example of a digital Riemann surfaces with non-trivial fibers.

Consider an encoding $\check{\mathcal{R}} = (\lambda, A, \pi, \square\!\!\square)$ of a digital Riemann surface $\mathcal{R}$ at scale λ. This encoding induces a natural "doubled encoding" $\check{\mathcal{R}}^{\boxplus} = (\lambda/2, A^{\boxplus}, \pi^{\boxplus}, \square\!\!\square^{\boxplus})$, by associating four nodes $a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1} \in A^{\boxplus}$ with $\pi^{\boxplus}(a_{\delta,\varepsilon}) = 2\,\pi(a) + \delta + \varepsilon\,\mathrm{i}$ to each a. Given $a_{\delta,\varepsilon}$, $a'_{\delta',\varepsilon'} \in A^{\boxplus}$, we set $a_{\delta,\varepsilon} \square\!\!\square^{\boxplus} a'_{\delta',\varepsilon'}$ if and only if $a = a'$ and $\pi^{\boxplus}(a'_{\delta',\varepsilon'}) - \pi^{\boxplus}(a_{\delta,\varepsilon}) \in \{\pm 1, \pm\mathrm{i}\}$, or $a \square\!\!\square a'$ and $\pi(a'_{\delta',\varepsilon'}) - \pi(a_{\delta,\varepsilon}) = \pi(a') - \pi(a)$. The doubled encoding $\check{\mathcal{R}}^{\boxplus}$ encodes the same digital Riemann surface $\mathcal{R}$, but at the smaller scale $\lambda/2$. By induction, it is possible to obtain encodings at any scale $\lambda/2^n$ with $n \in \mathbb{N}$.

Given a digital Riemann surface $\mathcal{R}$, the above argument shows that there exists a maximal scale $\lambda_{\max} \in 2^{\mathbb{Z}}$, such that $\mathcal{R}$ admits an encoding at scale $\lambda = \lambda_{\max}/2^n$ for every $n \in \mathbb{N}$. Inversely, the encoding $(\lambda, A, \pi, \square\!\!\square)$ of $\mathcal{R}$ at a given scale λ is essentially unique (up to bijections $A \to A'$). Indeed, given $a \in A$, the center c_a of each $\mathcal{Q}_a$ ($a \in A$) corresponds to a unique point in $\mathcal{R}$. Furthermore, given $a, b \in A$ with $\pi(a) - \pi(b) \in \{\pm 1, \pm\mathrm{i}\}$, we have $a \square\!\!\square b$ if and only if the segment $[\pi(c_a), \pi(c_b)]$ lifts to a segment $[c_a, c_b]$ on $\mathcal{R}$. If the scale λ is clear from the context, then it will be convenient to denote "the" encoding of $\mathcal{R}$ by $(\lambda, A_{\mathcal{R}}, \pi_{\mathcal{R}}, \square\!\!\square_{\mathcal{R}})$. If $\mathcal{R}$ is the result of some computation, then we will denote by $\lambda_{\mathcal{R}}$ the scale of the corresponding representation.

Remark 3.3. In practice, it is more efficient to work with a set of scaled nodes A^{sc} instead of A. Each element of A^{sc} is a pair (a, n) with $a \in A$, $n \in 2^{\mathbb{N}}$ and $\pi(a) \in \lambda\, n(\mathbb{Z} + \mathbb{Z}\,\mathrm{i})$. A scaled node corresponds to n^2 nodes $(a_{i,j})_{0 \leqslant i,j < n}$ in A with $\pi(a_{i,j}) = \pi(a) + i + j\,\mathrm{i}$ and $\begin{smallmatrix} a_{i,j+1} & \boxplus & a_{i+1,j+1} \\ a_{i,j} & & a_{i+1,j} \end{smallmatrix}$ for all $0 \leqslant i, j < n - 1$. For simplicity, we will directly work with nodes in A in what follows. Nevertheless, with some additional effort, our algorithms can be adapted to work with scaled nodes.

Let $\mathcal{R}$ be a digital Riemann surface, with a fixed encoding $\check{\mathcal{R}} = (\lambda, A, \pi, \square\!\!\square)$. We write

$$\mathcal{R}^{\mathrm{dig}} = \{\zeta \in \mathcal{R} \colon \pi(\zeta) \in \mathbb{C}^{\mathrm{dig}}\}$$

for the set of *digital points* on $\mathcal{R}$. Such a point $\zeta \in \mathcal{R}^{\mathrm{dig}}$ can be encoded by a pair $\check{\zeta} = (a, z)$ with $a \in A$ and $z = \pi(\zeta)$ such that $\zeta \in \bar{\mathcal{Q}}_a \cap \mathcal{R}$. This encoding is unique, except when ζ lies on the border of two squares. Notice that $\mathcal{R}^{\mathrm{dig}}$ is an abstract computable set. Similarly, we write

$$\mathcal{R}^{\mathrm{com}} = \{\zeta \in \mathcal{R} : \pi(\zeta) \in \mathbb{C}^{\mathrm{com}}\}$$

for the set of *computable points* on $\mathcal{R}$. Such a point $\zeta \in \mathcal{R}^{\mathrm{com}}$ can be encoded by a pair $\check{\zeta} = (a, z)$ with $a \in A$ and $z = \pi(\zeta)$, such that the distance between ζ and $\bar{\mathcal{Q}}_a$ is bounded by $\lambda/2$. Hence, we have $\zeta \in \mathcal{Q}_a$, or $\zeta \in \mathcal{Q}_{\{a,b\}}$ with $a \square b$, or $\zeta \in \mathcal{Q}_{\{a,b,c,d\}}$ with $\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}$. In particular, ζ admits a computable open neighbourhood $\mathcal{U}_\zeta$, such that $\pi_{|\mathcal{U}_\zeta}$ is a homeomorphism onto a rectangle $\pi(\mathcal{U}_\zeta)$ with corners in $\mathbb{C}^{\mathrm{dig}}$. Notice that there exists no algorithm for testing whether $\zeta \in \mathcal{Q}_a$ for given $\zeta \in \mathcal{R}^{\mathrm{com}}$ and $a \in A$.

3.2. Digital Riemann pastings

During actual computations with digital Riemann surfaces, the conditions **DR2** and **DR3** are not always met *a priori*. In that case, we need an automatic procedure to identify nodes when necessary. This will be the main objective of this section.

Consider a tuple $\check{\mathcal{R}} = (\lambda, A, \pi, \square)$ as in the previous section which only satisfies **DR1**. Then the construction of the previous section still yields a topological space $\mathcal{R}$ with a projection $\pi : \mathcal{R} \to \mathbb{C}$, even though $\mathcal{R}$ may now contain points which are not locally homeomorphic to open subsets of $\mathbb{C}$. We will call $\mathcal{R}$ a *digital Riemann pasting* with encoding $\check{\mathcal{R}}$. For instance, taking $A = \{a, b, c\}$, $\pi(a) = \pi(b) = \pi(c) + \mathrm{i}$, $a \square c$ and $b \square c$, we obtain the digital Riemann pasting shown at the upper left hand side of figure 3.1.

A *quotient structure* on $\check{\mathcal{R}}$ is a pair $(\sim_1, \square_1)$, where $\sim_1$ is an equivalence relation on A and $\square_1 \supseteq \square$ an adjacency relation on A, such that

- $a \sim_1 b \Rightarrow \pi(a) = \pi(b)$.
- $a \square_1 b \wedge a' \sim_1 a \wedge b' \sim_1 b \Rightarrow a' \square_1 b'$.

In that case, when setting $\pi_1(a/\sim_1) = \pi(a)$ for all $a \in A$, the tuple $\check{\mathcal{R}}/(\sim_1, \square_1) = (\lambda, A/\sim_1, \pi_1, \square_1)$ again encodes a digital Riemann pasting.

The intersection $(\sim_1 \cap \sim_2, \square_1 \cap \square_2)$ of two quotient structures $(\sim_1, \square_1)$ and $(\sim_2, \square_2)$ is again a quotient structure. Moreover, if both $\check{\mathcal{R}}/(\sim_1, \square_1)$ and $\check{\mathcal{R}}/(\sim_2, \square_2)$ encode digital Riemann surfaces, then so does $\check{\mathcal{R}}/(\sim_1 \cap \sim_2, \square_1 \cap \square_2)$. Consequently, $(=_A, \square)$ generates a smallest quotient structure $(\sim^*, \square^*)$ for which $\check{\mathcal{R}}^* = \check{\mathcal{R}}/(\sim^*, \square^*)$ encodes a digital Riemann surface. We call $\check{\mathcal{R}}^*$ the *normalization* of $\check{\mathcal{R}}$ and the corresponding digital Riemann surface $\mathcal{R}^*$ the normalization of $\mathcal{R}$. It can be checked that normalization commutes with the doubling operator $(\check{\mathcal{R}}^{\boxplus})^* = (\check{\mathcal{R}}^*)^{\boxplus}$, so that that this definition indeed does not depend on the chosen scale. Two examples of normalizations are shown in figure 3.1.

Example 3.4. Let $\mathcal{R}$ be a digital Riemann pasting and let $\sim$ be an equivalence relation on $\mathcal{R}$ with $\zeta \sim \xi \Rightarrow \pi(\zeta) = \pi(\xi)$. Given an encoding $\check{\mathcal{R}} = (\lambda, A, \pi, \square)$, we may define an equivalence relation $\sim'$ on A by $a \sim' b \Leftrightarrow \exists \zeta \in \bar{\mathcal{Q}}_a, \exists \xi \in \bar{\mathcal{Q}}_b, \zeta \sim \xi$. Then $(\check{\mathcal{R}}/(\sim', \square))^*$ encodes a digital Riemann surface, which we will denote by $\mathcal{R}/\sim$.

In order to compute the normalization of a digital Riemann pasting $\mathcal{R}$, it is convenient to maintain

- The map π^{-1} which associates to each $z \in \operatorname{im} \pi$ its preimage $\pi^{-1}(z) \subseteq A$.
- The map $\mu : A \times \{\pm 1, \pm \mathrm{i}\} \to \mathscr{P}(A)$ with $\mu(a, \delta) = \{b \in A : \pi(b) = \pi(a) + \delta \wedge a \square b\}$.

Given a subset $B \subseteq A$ of nodes such that $\pi(B)$ consists of a singleton z, we may then glue all nodes in B together using

- $\mu(a, -\delta) := \mu(a, -\delta) \cup \{B\} \setminus B$ for all $\delta \in \{\pm 1, \pm \mathrm{i}\}$ and $a \in \mu(b, \delta)$ for some $b \in B$.
- $\mu(B, \delta) := \bigcup_{b \in B} \mu(b, \delta)$ for all $\delta \in \{\pm 1, \pm \mathrm{i}\}$.
- $\pi^{-1}(z) := \pi^{-1}(z) \cup \{B\} \setminus B$.

In order to normalize $\mathcal{R}$, we now keep on doing the following operations until both **DR2** and **DR3** are satisfied:

- If **DR2** is not satisfied, then there exist $a \in A$ and $\delta \in \{\pm 1, \pm \mathrm{i}\}$ such that $\mu(a, \delta)$ contains more than one element. In that case, we glue all elements in $\mu(a, \delta)$ together using the above procedure.
- If **DR2** is satisfied, but not **DR3**, then there exits an $a \in A$ and a permutation $\{\delta_1, ..., \delta_4\}$ of $\{\pm 1, \pm \mathrm{i}\}$ with $\mu(\mu(\mu(a, \delta_1), \delta_2), \delta_3) = \{b\}$, but $b \notin \mu(a, -\delta_4)$. In that case, we add b to $\mu(a, -\delta_4)$ and a to $\mu(b, \delta_4)$.

The normalization procedure finishes, because the size of A strictly decreases for the first operation and the number of $(a, b) \in A^2$ strictly increases for the second operation.

3.3. Digital coverings and computable Riemann surfaces

A *digital covering* is a covering $\varphi\colon \mathcal{R}_1 \to \mathcal{R}_2$ in the classical sense between two digital Riemann surfaces. Let φ be a digital covering and let $\check{\mathcal{R}}_1 = (\lambda, A_1, \pi_1, \square_1)$ and $\check{\mathcal{R}}_2 = (\lambda, A_2, \pi_2, \square_2)$ be encodings of $\mathcal{R}_1$ and $\mathcal{R}_2$ at the same scale λ. Then φ induces a mapping $F\colon A_1 \to A_2$, which sends $a \in A_1$ to the unique $F(a) \in A_2$ with $c_{F(a)} = \varphi(c_a)$, where c_a stands for the center of $\mathcal{Q}_a$. This mapping satisfies

$$\pi_2 \circ F = \pi_1 \tag{3.1}$$

$$a \square_1 b \Rightarrow F(a) \square_2 F(b) \qquad (a, b \in A_1) \tag{3.2}$$

Inversely, given a mapping $F\colon A_1 \to A_2$ which satisfies (3.1) and (3.2), we obtain a covering in the classical sense by sending $\zeta \in \bar{\mathcal{Q}}_a \cap \mathcal{R}_1$ to the unique point $\varphi(\zeta) \in \bar{\mathcal{Q}}_{F(a)}$ with $\pi_2(F(\zeta)) = \pi_1(\zeta)$. In other words, the digital covering φ may be encoded by the triple $\check{\varphi} = (\check{\mathcal{R}}_1, \check{\mathcal{R}}_2, F)$. We will denote by $\mathbb{V}^{\mathrm{dig}}$ the set of all digital coverings.

Example 3.5. Let $\mathcal{R}$ be a digital Riemann surface encoded by $\check{\mathcal{R}} = (\lambda, A, \pi, \square)$. Consider the equivalence relation $a \sim b \Leftrightarrow \pi(a) = \pi(b)$ on A and the projection $P\colon A \to A/\sim$. Then the tuple $\check{\mathcal{R}}^{\mathrm{pl}} = (\lambda, A^{\mathrm{pl}}, \pi^{\mathrm{pl}}, \square^{\mathrm{pl}})$ with $A^{\mathrm{pl}} = A/\sim$, $\pi^{\mathrm{pl}} \circ P = \pi$ and $a \square^{\mathrm{pl}} b \Leftrightarrow \pi(a) - \pi(b) \in \{\pm 1, \pm \mathrm{i}\}$ encodes the digital complex subset $\pi(\mathcal{R})$ of $\mathbb{C}$ and $\pi\colon \mathcal{R} \to \pi(\mathcal{R})$ is a digital covering.

Example 3.6. The definition of digital coverings naturally extends to digital Riemann pastings. The normalization $\mathcal{R}^*$ of a digital Riemann pasting comes with a natural digital covering $\cdot^*\colon \mathcal{R} \to \mathcal{R}^*$; $\zeta \mapsto \zeta^*$. In particular, given an equivalence relation $\sim$ on $\mathcal{R}$ with $\zeta \sim \xi \Rightarrow \pi(\zeta) = \pi(\xi)$, we obtain a natural covering $\mathcal{R} \to \mathcal{R}/\sim$. This generalizes the previous example, by taking $\zeta \sim \xi \Leftrightarrow \pi(\zeta) = \pi(\xi)$. Moreover, any digital covering $\varphi\colon \mathcal{R} \to \mathcal{S}$ induces a digital covering $\varphi^*\colon \mathcal{R}^* \to \mathcal{S}^*$ which commutes with $\cdot^*$.

Given digital Riemann surfaces $\mathcal{R}_0, \mathcal{R}_1, ...$ and coverings $\varphi_0\colon \mathcal{R}_0 \to \mathcal{R}_1$, $\varphi_1\colon \mathcal{R}_1 \to \mathcal{R}_2, ...$, we call

$$\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \mathcal{R}_2 \xrightarrow{\varphi_2} \cdots \tag{3.3}$$

a *digital covering sequence*. Such a sequence admits a natural limit

$$\mathcal{R} = \lim \mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots = (\mathcal{R}_0 \amalg \mathcal{R}_1 \amalg \cdots)/\sim, \tag{3.4}$$

where $\sim$ is the smallest equivalence relation with $\zeta \sim \varphi_i(\zeta)$ for each $\zeta \in \mathcal{R}_i$, and $\mathcal{R}$ has the natural structure of a Riemann surface. We will write $\varphi_{i;j}$ for the composed covering $\varphi_{j-1} \circ \cdots \circ \varphi_i : \mathcal{R}_i \to \mathcal{R}_j$ and $\varphi_{i;}$ for the covering which sends $\zeta \in \mathcal{R}_i$ to $\zeta/\sim \in \mathcal{R}$. We say that the covering sequence (3.3) is *computable*, if the mapping $\check{\mathcal{R}} : \mathbb{N} \to \mathbb{S}^{\mathrm{dig}} \times \mathbb{V}^{\mathrm{dig}}; n \mapsto (\mathcal{R}_n, \varphi_n)$ is computable. In that case, we call $\mathcal{R}$ a *computable Riemann surface*. Notice that coverings are not necessarily injective. This corresponds to the idea that better knowledge of the Riemann surface of a function may lead to the detection of identical leaves.

Example 3.7. Let $\mathcal{R} \subseteq \mathbb{C}$ be an open rectangle with corners in $\mathbb{C}^{\mathrm{com}}$ or an open disk with center in $\mathbb{C}^{\mathrm{com}}$ and radius in $\mathbb{R}^{\mathrm{com},>}$. For each $n \in \mathbb{N}$, let

$$\begin{aligned} \lambda_n &= 2^{-n} \\ A_n &= \{a \in \mathbb{Z}[\mathrm{i}] : \bar{\mathcal{Q}}_{a,\lambda_n} \subseteq \mathcal{R}\}. \end{aligned}$$

By example 3.1, λ_n and A_n determine a digital subset $\mathcal{R}_n$ of $\mathbb{C}$. The limit of the sequence of embeddings $\mathcal{R}_0 \to \mathcal{R}_1 \to ...$ is a computable digital Riemann surface, which is homeomorphic to $\mathcal{R}$. More generally, the limit of a computable sequence $\mathcal{R}_0 \to \mathcal{R}_1 \to \cdots$ of embeddings of digital subsets of $\mathbb{C}$ is called a *computable open subset* of $\mathbb{C}$.

Example 3.8. The example 3.2 can be adapted to turn infinitely many times around the hole in the middle. Indeed, consider the "infinite digital Riemann surface" $\mathcal{R}_\infty$ encoded by:

$$\begin{aligned} \lambda &= 1 \\ A &= \{a_{x,y;k} : x, y \in \{-1, 0, 1\}, (x, y) \neq 0, k \in \mathbb{Z}\} \\ \pi(a_{x,y;k}) &= x + y\,\mathrm{i} \\ a_{x,y;k} \square\!\!\square a_{x',y';k'} &\Leftrightarrow |x' - x| + |y' - y| = 1 \wedge \\ &\quad (((x, y) = (1, -1) \wedge (x', y') = (1, 0) \wedge k' = k + 1) \vee \\ &\quad ((x', y') = (1, -1) \wedge (x, y) = (1, 0) \wedge k = k' + 1) \vee \\ &\quad (\{(x, y), (x', y')\} \neq \{(1, 0), (1, -1)\} \wedge k' = k)) \end{aligned}$$

Given $n \in \mathbb{N}$, the restriction $\mathcal{R}_n$ of $\mathcal{R}_\infty$ to those $a_{x,y,k}$ with $-n \leqslant k \leqslant n$ determines a digital Riemann surface $\mathcal{R}_n$ in the usual sense. The natural inclusions determine a digital covering sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$ whose limit corresponds to $\mathcal{R}_\infty$. Notice that $\mathcal{R}_\infty$ is isomorphic to the universal covering space of $\pi(\mathcal{R}_\infty)$; see also section 4.7.

Let $\mathcal{R}$ be a fixed computable Riemann surface. We denote by

$$\begin{aligned} \mathcal{R}^{\mathrm{dig}} &= \{\zeta \in \mathcal{R} : \pi(\zeta) \in \mathbb{C}^{\mathrm{dig}}\} \\ \mathcal{R}^{\mathrm{com}} &= \{\zeta \in \mathcal{R} : \pi(\zeta) \in \mathbb{C}^{\mathrm{com}}\} \end{aligned}$$

the sets of digital and computable points on $\mathcal{R}$. A digital point $\zeta \in \mathcal{R}^{\mathrm{dig}}$ (and similarly for a computable point $\zeta \in \mathcal{R}^{\mathrm{com}}$) may be encoded by a partial sequence $\check{\zeta} : n \geqslant n_\zeta \mapsto \zeta_n \in \mathcal{R}_n^{\mathrm{dig}}$ such that $\zeta_{n+1} = \varphi_n(\zeta_n)$ and $\zeta = \varphi_{n;}(\zeta_n)$ for all $n \geqslant n_\zeta$. We notice that $\mathcal{R}^{\mathrm{dig}}$ is an abstract enumerable set. We have natural computable mappings

$$\begin{aligned} \pi : \quad \mathcal{R}^{\mathrm{dig}} &\to \mathbb{C}^{\mathrm{dig}} \\ \pi : \quad \mathcal{R}^{\mathrm{com}} &\to \mathbb{C}^{\mathrm{com}} \end{aligned}$$

As in the case of digital Riemann surfaces, each $\zeta \in \mathcal{R}^{\mathrm{com}}$ admits a computable open neighbourhood $\mathcal{U}_\zeta$, such that $\pi_{|\mathcal{U}_\zeta}$ is a homeomorphism onto a rectangle $\pi(\mathcal{U}_\zeta)$ with corners in $\mathbb{C}^{\mathrm{dig}}$.

3.4. Atlas representation of a computable Riemann surface

Instead of using the *digital representation* of computable Riemann surfaces (i.e. as limits of digital covering sequences), we may also try to mimic more classical representations of Riemann surfaces. For instance, a *computable atlas representation* of a Riemann surface $\mathcal{R}$ with projection $\pi: \mathcal{R} \to \mathbb{C}$ is a tuple $\check{\mathcal{R}} = (\mathcal{A}, \mathcal{U}, \mathsf{lift}, \mathcal{V}, \sqcap)$, where

- $\mathcal{A}$ is an abstract enumerable set.
- $\mathcal{U}$ is a computable map which sends $a \in \mathcal{A}$ to a computable open subset $\mathcal{U}_a$ of $\mathbb{C}^{\mathrm{com}}$.
- $\mathsf{lift}: (\mathcal{A}, \mathbb{C}^{\mathrm{com}}) \rightharpoonup \mathcal{R}^{\mathrm{com}}$ is a computable partial map such that $\mathsf{lift}(a, \cdot): \mathcal{U}_a^{\mathrm{com}} \to \mathcal{R}^{\mathrm{com}}$ is an immersion for every $a \in \mathcal{A}$, with $\pi(\mathsf{lift}(a, z)) = z$ for all $z \in \mathcal{U}_a^{\mathrm{com}}$. Here
$$\mathcal{R}^{\mathrm{com}} = \{z \in \mathcal{R}: \pi(z) \in \mathbb{C}^{\mathrm{com}}\}.$$
- $\mathcal{V}: \mathcal{R}^{\mathrm{com}} \to \mathcal{A}$ is a computable function such that $z \in \operatorname{im} \mathsf{lift}(\mathcal{V}_z, \cdot)$ for all $z \in \mathcal{R}^{\mathrm{com}}$.
- An enumerable relation $\sqcap \subseteq \mathcal{A}^2$ with
$$a \sqcap b \Leftrightarrow \operatorname{im} \mathsf{lift}(a, \cdot) \cap \operatorname{im} \mathsf{lift}(b, \cdot) \neq \varnothing.$$

PROPOSITION 3.9. *Any computable Riemann surface admits an atlas representation.*

Proof. Let $\mathcal{R}$ be the limit of a digital covering sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$ of digital Riemann surfaces $\mathcal{R}_n$ and define

$$\begin{aligned}
\mathcal{A} &= \{(n, \{a\}): n \in \mathbb{N}, a \in A_{\mathcal{R}_n}\} \cup \\
&\quad \{(n, \{a, b\}): n \in \mathbb{N}, a, b \in A_{\mathcal{R}_n}, a \square\!\!\square b\} \cup \\
&\quad \{(n, \{a, b, c, d\}): n \in \mathbb{N}, a, b, c, d \in A_{\mathcal{R}_n}, {}_c^a\boxplus_d^b\} \\
\mathcal{U} &: (n, B) \in \mathcal{A} \mapsto \mathcal{Q}_{\pi_{\mathcal{R}_n}(B)}.
\end{aligned}$$

Given $\zeta = \varphi_{n;}(\zeta_n) \in \mathcal{R}^{\mathrm{com}}$, let $\check{\zeta}_n = (a, z) \in A_{\mathcal{R}_n} \times \mathbb{C}^{\mathrm{com}}$ be an encoding of ζ_n. We have already noticed that $z \in \mathcal{Q}_B$ for $B = \{a\}$, $B = \{a, b\}$ with $a \square\!\!\square b$ or $B = \{a, b, c, d\}$ with ${}_c^a\boxplus_d^b$. We may thus take $\mathcal{V}_\zeta = (n, B)$. Conversely, given $(n, B) \in \mathcal{A}$, the composition of $\pi^{-1}: \mathcal{Q}_{\pi_{\mathcal{R}_n}(B)} \to \mathcal{Q}_B$ and the restriction of $\varphi_{n;}$ to $\mathcal{Q}_B$ determines an immersion $\mathsf{lift}((n, B), \cdot)$ of $\mathcal{Q}_{\pi_{\mathcal{R}_n}(B)}$ into $\mathcal{R}$. Finally, given pairs (i, B), $(j, C) \in \mathcal{A}$, we may ultimately check whether $\varphi_{i;}(\mathcal{Q}_B) \cap \varphi_{j;}(\mathcal{Q}_C) \neq \varnothing$: given $n \in \mathbb{N}$, we check whether $n \geqslant \max(i, j)$ and $\varphi_{i;n}(\mathcal{Q}_B) \cap \varphi_{j;n}(\mathcal{Q}_C) \neq \varnothing$. □

PROPOSITION 3.10. *Any Riemann surface with a computable atlas representation can be given the structure of a computable Riemann surface.*

Proof. Let $\mathcal{A} = \{a_0, a_1, \ldots\}$ be an enumeration of $\mathcal{A}$ and $\{E_0, E_1, \ldots\}$ an enumeration of all pairs (i, j) with $a_i \sqcap a_j$.

Let us first assume that each $\mathcal{U}_{a_n}$ is a digital subset of $\mathbb{C}$. Consider the disjoint union $\mathcal{U}_{a_0} \amalg \cdots \amalg \mathcal{U}_{a_n}$, together with the smallest equivalence relation $\sim$ for which corresponding squares in $\mathcal{U}_{a_i}$ and $\mathcal{U}_{a_j}$ are equivalent if and only if $(i, j) \in \{E_0, \ldots, E_n\}$. Setting $\mathcal{R}_n = (\mathcal{U}_{a_0} \amalg \cdots \amalg \mathcal{U}_{a_n})/\!\sim$, we obtain a natural computable digital covering sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$. We claim that $\mathcal{R}$ is isomorphic to the limit $\tilde{\mathcal{R}}$ of this sequence.

Indeed, the construction implies natural coverings $\psi_n: \mathcal{R}_n \to \mathcal{R}$ which pass to the limit $\psi: \tilde{\mathcal{R}} \to \mathcal{R}$. Inversely, $\operatorname{im} \mathsf{lift}(a, \cdot)$ naturally immerses into $\tilde{\mathcal{R}}$, with inverse ψ. Gluing these immersions together for all $a \in \mathcal{A}$, we obtain a covering $\xi: \mathcal{R} \to \tilde{\mathcal{R}}$ with $\psi \circ \xi = \mathrm{Id}_{\mathcal{R}}$ (since every $z \in \mathcal{R}^{\mathrm{com}}$ is contained in $\operatorname{im} \mathsf{lift}(\mathcal{V}_z, \cdot)$), proving that $\mathcal{R} \cong \tilde{\mathcal{R}}$.

In the general case, each $\mathcal{U}_{a_n}$ is the computable limit of a sequence $\mathcal{R}_{n,0} \to \mathcal{R}_{n,1} \to \cdots$ of immersions. We may now construct another computable atlas representation of $\mathcal{R}$, by taking $\tilde{\mathcal{A}} = \{a_{0,0}, a_{1,0}, a_{0,1}, a_{2,0}, a_{1,1}, a_{0,2}, \ldots\}$, $\tilde{\mathcal{U}}_{a_{i,j}} = \mathcal{R}_{i,j}$, etc. We conclude by applying the above argument to this new computable atlas representation. □

Remark 3.11. From the proofs of the above propositions, it becomes clear that the class of Riemann surfaces with a computable atlas representation does not change if we require the computable open sets $\mathcal{U}_a$ to be of a prescribed type, like open rectangles with corners in $\mathbb{C}^{\mathrm{dig}}$ or open balls with centers in $\mathbb{C}^{\mathrm{com}}$ and radii in $\mathbb{R}^{\mathrm{com},>}$.

3.5. Intrinsic representation of a computable Riemann surface

Let $\mathcal{R}$ be a classical Riemann surface above $\mathbb{C}$ and denote

$$\begin{aligned} \mathcal{R}^{\mathrm{dig}} &= \{\zeta \in \mathcal{R} : \pi(\zeta) \in \mathbb{C}^{\mathrm{dig}}\}; \\ \mathcal{R}^{\mathrm{com}} &= \{\zeta \in \mathcal{R} : \pi(\zeta) \in \mathbb{C}^{\mathrm{com}}\}. \end{aligned}$$

Given $z \in \mathbb{C}$ and $\rho \in \mathbb{R}^{>} \cup \{+\infty\}$, we denote

$$\begin{aligned} \mathcal{B}_{z,\rho} &= \{z + \delta \in \mathbb{C} : |\delta| < \rho\} & \mathcal{B}_{\rho} &= \mathcal{B}_{0,\rho} \\ \bar{\mathcal{B}}_{z,\rho} &= \{z + \delta \in \mathbb{C} : |\delta| \leqslant \rho\} & \bar{\mathcal{B}}_{\rho} &= \bar{\mathcal{B}}_{0,\rho} \end{aligned} \tag{3.5}$$

Given a point $\zeta \in \mathcal{R}$, let $r_\zeta \in \mathbb{R}^{>} \cup \{+\infty\}$ be the largest radius such that there exists an open disk $\mathcal{B}_{\pi(\zeta),r} \subseteq \mathbb{C}$ for which ζ admits an open neighbourhood $\mathcal{V} \subseteq \mathcal{R}$ so that the restriction $\pi_{|\mathcal{V}}$ of π to $\mathcal{V}$ is a homeomorphism between $\mathcal{V}$ and $\mathcal{B}_{\pi(\zeta),r}$. Given $\delta \in \mathbb{C}$ with $|\delta| < r_\delta$, we denote by $\zeta + \delta$ or $\zeta_{+\delta}$ the unique point in $\mathcal{V}$ with $\pi(\zeta + \delta) = \pi(\zeta) + \delta$. In particular, the notations (3.5) naturally generalize to the case of balls $\mathcal{B}_{\zeta,\rho}$ and $\bar{\mathcal{B}}_{\zeta,\rho}$ in $\mathcal{R}$, for $\rho \leqslant r_\zeta$ (resp. $\rho < r_\zeta$).

A *computable intrinsic representation* of $\mathcal{R}$ is a tuple $\check{\mathcal{R}} = (\chi, \mathsf{dig}, \pi^{\mathrm{com}}, r^{\mathrm{com}}, +^{\mathrm{com}}, \mathsf{near})$ such that

- χ is an encoding for $\mathcal{R}^{\mathrm{com}}$.
- $\pi^{\mathrm{com}} : \mathcal{R}^{\mathrm{com}} \to \mathbb{C}^{\mathrm{com}}$ is a computable function with $\pi^{\mathrm{com}} = \pi_{|\mathcal{R}^{\mathrm{com}}}$.
- $\mathsf{dig} : \mathbb{N} \to \mathcal{R}^{\mathrm{com}}$ is a sequential enumeration of the elements of $\mathcal{R}^{\mathrm{dig}}$.
- $r^{\mathrm{com}} : \mathcal{R}^{\mathrm{com}} \to \bar{\mathbb{R}}^{\mathrm{lcom},>}$ is a computable function with $r^{\mathrm{com}} = r_{|\mathcal{R}^{\mathrm{com}}}$.
- $+^{\mathrm{com}} : \mathcal{R}^{\mathrm{com}} \times \mathbb{C}^{\mathrm{com}} \rightharpoonup \mathcal{R}^{\mathrm{com}}$ is a computable function with $\zeta +^{\mathrm{com}} \delta = \zeta + \delta$ for all $\zeta \in \mathcal{R}^{\mathrm{com}}$ and $\delta \in \mathcal{B}^{\mathrm{com}}_{r_\zeta}$.
- $\mathsf{near} \subseteq (\mathcal{R}^{\mathrm{com}})^2$ is an enumerable relation with

$$\mathsf{near}(\zeta, \xi) \Leftrightarrow \xi \in \mathcal{B}_{\zeta, r_\zeta}.$$

Simplifying notations $\pi^{\mathrm{com}} \to \pi$, $r^{\mathrm{com}} \to r$ and $\zeta +^{\mathrm{com}} \delta \to \zeta + \delta$, we thus have the following signature:

$$\begin{aligned} \pi&: & \mathcal{R}^{\mathrm{com}} &\to \mathbb{C}^{\mathrm{com}} \\ r&: & \mathcal{R}^{\mathrm{com}} &\to \bar{\mathbb{R}}^{\mathrm{lcom},>} \\ +&: \mathcal{R}^{\mathrm{com}} \times \mathbb{C}^{\mathrm{com}} &\rightharpoonup \mathcal{R}^{\mathrm{com}} \end{aligned}$$

PROPOSITION 3.12. *Any computable Riemann surface $\mathcal{R}$ admits a computable intrinsic representation.*

Proof. Let $\mathcal{R}$ be the limit of a digital covering sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$. For $\check{\mathcal{R}}$, we take the set of encodings of points $\zeta \in \mathcal{R}^{\mathrm{com}}$ and we already have a computable mapping $\pi : \mathcal{R}^{\mathrm{com}} \to \mathbb{C}^{\mathrm{com}}$.

Let $\check{\zeta}: n \geqslant n_\zeta \mapsto \zeta_n$ be the encoding of a point $\zeta \in \mathcal{R}^{\mathrm{com}}$. The distance r_{ζ_n} of to the border $\partial\mathcal{R}_n$ is easily computed for each $n \geqslant n_\zeta$. Since $r_{\zeta_n} \leqslant r_{\zeta_{n+1}} \leqslant \cdots$ and $\lim_{n\to\infty} r_{\zeta_n} = r_\zeta$, the sequence $i \mapsto r_{\zeta_{n_\zeta+i}}$ encodes $r_\zeta \in \bar{\mathbb{R}}^{\mathrm{lcom},>}$. Similarly, given $\delta \in \mathbb{C}^{\mathrm{com}}$ with $|\delta| < r_\zeta$, it is easy to compute $\zeta_n + \delta$ in $\mathcal{R}_n$ for each sufficiently large $n \geqslant n_0$. Then the sequence $n \geqslant n_0 \mapsto \zeta_n + \delta$ encodes $\zeta + \delta$.

Finally, $\mathcal{R}^{\mathrm{dig}} = \varphi_{0;}(\mathcal{R}_0^{\mathrm{dig}}) \cup \varphi_{1;}(\mathcal{R}_1^{\mathrm{dig}}) \cup \cdots$ yields an enumeration dig of $\mathcal{R}^{\mathrm{dig}}$. Given $\zeta, \xi \in \mathcal{R}^{\mathrm{com}}$ with encodings $\check{\zeta}: n \geqslant n_\zeta \mapsto \zeta_n$ and $\check{\xi}: n \geqslant n_\xi \mapsto \xi_n$, we may ultimately check whether $\mathsf{near}(\zeta, \xi)$ holds: given $n \in \mathbb{N}$, we test whether $n \geqslant \max(n_\zeta, n_\xi)$ and $\xi_n \in \mathcal{B}_{\zeta_n, r_{\zeta_n}}$. □

PROPOSITION 3.13. *Let $\mathcal{R}$ be a Riemann surface with a computable intrinsic representation. Then $\mathcal{R}$ is a computable Riemann surface.*

Proof. Let $\{\zeta_0, \zeta_1, \ldots\}$ be the enumeration of $\mathcal{R}^{\mathrm{dig}}$ and $\{E_0, E_1, \ldots\}$ an enumeration of all pairs $(i, j) \in \mathbb{N}^2$ such that $\mathsf{near}(\zeta_i, \zeta_j)$ holds.

For each $n \in \mathbb{N}$, we may compute a square $\mathcal{Q}_n \subseteq \mathcal{B}_{\pi(\zeta_n), r_{\zeta_n}}$ with corners in $\mathbb{Z}[\mathrm{i}]\ \mu_n$, for some $\mu_n \in 2^{\mathbb{Z}}$ such that $r_{\zeta_n}/8 < \mu_n < r_{\zeta_n}/2$. Now let $\mathcal{R}_n = (\mathcal{Q}_0 \amalg \cdots \amalg \mathcal{Q}_n)/\sim$, where $\sim$ is the smallest equivalence relation induced by identifying matching squares in $\mathcal{Q}_i$ and $\mathcal{Q}_j$ for pairs $(i, j) \in \{E_0, \ldots, E_n\}$. We claim that the limit $\tilde{\mathcal{R}}$ of the induced digital covering sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$ is isomorphic to $\mathcal{R}$.

Indeed, we have natural coverings $\psi_n: \mathcal{R}_n \to \mathcal{R}$ for each n, which pass to the limit $\psi: \tilde{\mathcal{R}} \to \mathcal{R}$. Inversely, for each n, the set $\mathcal{B}_{\zeta_0, r_{\zeta_0}} \cup \cdots \cup \mathcal{B}_{\zeta_n, r_{\zeta_n}}$ can be immersed in some $\mathcal{R}_{k(n)}$, where $k(n)$ is large enough such that all pairs (i, j) with $\zeta_j \in \mathcal{B}_{\zeta_i, r_{\zeta_i}}$ are among $\{E_0, \ldots, E_{k(n)}\}$. Gluing these immersions together, we this obtain an immersion $\iota: \mathcal{R} \to \tilde{\mathcal{R}}$ with $\psi \circ \iota = \mathrm{Id}_{\mathcal{R}}$, proving that $\tilde{\mathcal{R}} \cong \mathcal{R}$. □

3.6. Optional features of computable Riemann surfaces

Let $\mathcal{R}$ be a computable Riemann surface. In certain cases, we may design an algorithm $r.: \mathcal{R}^{\mathrm{com}} \to \bar{\mathbb{R}}^{\mathrm{com},>}$ to compute the distance of a point $\zeta \in \mathcal{R}^{\mathrm{com}}$ to the border. In that case, we say that $\mathcal{R}$ is a *delimited computable Riemann surface*.

Remark 3.14. One might prefer to call computable Riemann surfaces in our sense *lower computable Riemann surfaces* and delimited computable Riemann surfaces simply computable Riemann surfaces (and similarly for computable open sets). However, in what follows, we will mainly have to deal with computable Riemann surfaces for which we do not have a computable distance function $r.: \mathcal{R}^{\mathrm{com}} \to \bar{\mathbb{R}}^{\mathrm{com},>}$. Therefore, we will stick to the original definition.

Assume that $\mathcal{R} = \lim \mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$ and consider two points $\zeta, \zeta' \in \mathcal{R}^{\mathrm{com}}$. Even under the assumption that $\pi(\zeta) = \pi(\zeta')$, we notice that there exists no test in order to decide whether $\zeta = \zeta'$. Indeed, given encodings $\check{\zeta}: n \geqslant n_\zeta \mapsto \zeta_n$ and $\check{\zeta}': n \geqslant n_{\zeta'} \mapsto \zeta'_n$ of ζ resp. ζ', we do not know whether there exists an index n with $\zeta_n = \zeta'_n$. Nevertheless, we naturally do have such a test in the case when the coverings φ_i are embeddings. In this case, we say that $\mathcal{R}$ has *computable branches*. Conversely, assume that we have a *conditional equality test*

$$=.: \ \mathcal{R}^{\mathrm{com}} \times \mathcal{R}^{\mathrm{com}} \times \{\mathbf{true}, \mathbf{false}\} \ \to \ \{\mathbf{true}, \mathbf{false}\}$$

where $\zeta =_b \zeta'$ returns the result of the test $\zeta = \zeta'$, provided that we are given the answer b to the test $\pi(\zeta) = \pi(\zeta')$. Equivalently, one may assume a predicate

$$\mathsf{near}: \ \mathcal{R}^{\mathrm{com}} \times \mathcal{R}^{\mathrm{com}} \ \rightharpoonup \ \{\mathbf{true}, \mathbf{false}\}$$

such that $\mathsf{near}(\zeta, \xi)$ holds if and only if $\xi \in \mathcal{B}_{\zeta, r_\zeta}$, provided that $\pi(\xi) \in \mathcal{B}_{\pi(\zeta), r_\zeta}$. Then we may associate a new digital Riemann surface $\tilde{\mathcal{R}}_k$ to each $\mathcal{R}_k$, by identifying all squares $\mathcal{Q}_a$ with $a \in A_{\mathcal{R}_k}$ whose centers are equal (using the normalization algorithm from section 3.2). This leads to a new representation $k \mapsto (\tilde{\mathcal{R}}_k, \tilde{\varphi}_k)$ of $\mathcal{R}$, for which the induced coverings $\tilde{\varphi}_k$ are embeddings. When using the atlas representation, $\mathcal{R}$ has computable branches if and only if we have a computable test for deciding whether $\operatorname{im} \mathsf{lift}(a, \cdot) \cap \operatorname{im} \mathsf{lift}(b, \cdot) \neq \varnothing$.

4. Constructions of computable Riemann surfaces

4.1. Computable coverings

Consider two computable Riemann surfaces $\mathcal{R}$ and $\mathcal{S}$. A covering $\xi : \mathcal{R} \to \mathcal{S}$ is said to be *computable* if its restriction to $\mathcal{R}^{\mathrm{com}}$ is a computable mapping $\mathcal{R}^{\mathrm{com}} \to \mathcal{S}^{\mathrm{com}}$. A *digital representation* of such a covering is a triple $\check{\xi} = (\check{\mathcal{R}}, \check{\mathcal{S}}, \xi^{\mathrm{seq}})$, such that $\check{\mathcal{R}} : n \mapsto (\mathcal{R}_n, \varphi_n)$ represents $\mathcal{R}$, $\check{\mathcal{S}} : n \mapsto (\mathcal{S}_n, \psi_n)$ represents $\mathcal{S}$ and $\xi^{\mathrm{seq}} : n \mapsto \xi_n$ is a computable sequence of digital coverings $\xi_n : \mathcal{R}_n \to \mathcal{S}_n$, such that

$$\begin{array}{ccccccc} \mathcal{R}_0 & \xrightarrow{\varphi_0} & \mathcal{R}_1 & \xrightarrow{\varphi_1} & \mathcal{R}_2 & \xrightarrow{\varphi_2} & \cdots \\ \downarrow \xi_0 & & \downarrow \xi_1 & & \downarrow \xi_2 & & \\ \mathcal{S}_0 & \xrightarrow{\psi_0} & \mathcal{S}_1 & \xrightarrow{\psi_1} & \mathcal{S}_2 & \xrightarrow{\psi_2} & \cdots \end{array} \tag{4.1}$$

commutes and $\xi(\varphi_{n;}(\zeta)) = \psi_{n;}(\xi_n(\zeta))$ for any $n \in \mathbb{N}$. If each ξ_i is an immersion, then we call $\check{\xi}$ a *computable immersion* (of representations). If ξ is also surjective, then we call $\check{\xi}$ a *computable subdivision* (of representations), and $\check{\mathcal{R}}$ is said to be a subdivision of $\check{\mathcal{S}}$.

LEMMA 4.1. *Let $\check{\mathcal{R}} : n \mapsto (\mathcal{R}_n, \varphi_n)$ be the representation of a computable Riemann surface. Then we may compute a computable subdivision $\check{\mathcal{S}} : n \mapsto (\mathcal{S}_n, \psi_n)$ of $\check{\mathcal{R}}$, such that there exist $\varepsilon_n > 0$ with $r_\zeta > \varepsilon_n$ for all $n \in \mathbb{N}$ and $\zeta \in \psi_{n;}(\mathcal{S}_n)$.*

Proof. Without loss of generality, we may assume that the $\mathcal{R}_n$ are encoded at scales $\lambda_{\mathcal{R}_0} > \lambda_{\mathcal{R}_1} > \cdots$. Given a digital Riemann surface $\mathcal{T}$ encoded by $(\lambda, A, \pi, \square)$, let $\circledcirc_\lambda \mathcal{T}$ stand for its restriction to the subset of *inner nodes* $a \in A$ which admit four distinct neighbours b_1, $b_2, b_3, b_4 \in A$. Taking $\mathcal{S}_n = \circledcirc_{\lambda_{\mathcal{R}_n}} \circledcirc_{\lambda_{\mathcal{R}_n}} \mathcal{R}_n$, $\psi_n = \varphi_{n|\mathcal{S}_n}$ and $\varepsilon_n = \lambda_{\mathcal{R}_n}$, the inclusion mappings $\mathcal{S}_n \to \mathcal{R}_n$ determine a computable immersion of the Riemann surface $\mathcal{S}$ represented by $\check{\mathcal{S}} : n \mapsto (\mathcal{S}_n, \psi_n)$ into $\mathcal{R}$. Since $\lambda_{\mathcal{R}_n} \to 0$, this immersion is actually a subdivision and we have $r_{\varphi_{n;}(\zeta_n)} \geqslant r_{\zeta_n} > \varepsilon_n := \lambda_{\mathcal{R}_n}$ for all $\zeta_n \in \mathcal{S}_n$. □

LEMMA 4.2. *Let $\mathcal{R}$ be the limit of a computable covering sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$ and $\mathcal{C} \subseteq \mathcal{R}$ a digital Riemann surface such that $\bar{\mathcal{C}}$ is compact. Then we may compute an $n \in \mathbb{N}$ and a digital Riemann surface $\mathcal{T} \subseteq \mathcal{R}_n$ with $\varphi_{n;}(\mathcal{T}) \supseteq \bar{\mathcal{C}}$.*

Proof. The set $\{\varphi_{0;}(\mathcal{R}_0), \varphi_{1;}(\mathcal{R}_1), \ldots\}$ forms an open covering of $\bar{\mathcal{C}}$. Since $\bar{\mathcal{C}}$ is compact, it follows that there exists an $k \in \mathbb{N}$ with $\varphi_{k;}(\mathcal{R}_k) \supseteq \bar{\mathcal{C}}$. Since $\varphi_{k;}(\mathcal{R}_k)$ and $\mathcal{C}$ are both digital Riemann surfaces, we may actually check whether $\varphi_{k;}(\mathcal{R}_k) \supseteq \bar{\mathcal{C}}$, and therefore compute the first k for which this holds. □

PROPOSITION 4.3. *Let $\xi : \mathcal{R} \to \mathcal{S}$ be a computable covering. Let $\check{\mathcal{R}} : n \mapsto (\mathcal{R}_n, \varphi_n)$ and $\check{\mathcal{S}} : n \mapsto (\mathcal{S}_n, \psi_n)$ be representations for $\mathcal{R}$ resp. $\mathcal{S}$. Modulo subdividing $\check{\mathcal{R}}$ and reindexing $\check{\mathcal{S}}$, the covering ξ admits a computable digital representation of the form $\check{\xi} = (\check{\mathcal{R}}, \check{\mathcal{S}}, \xi^{\mathrm{seq}})$.*

Proof. By lemma 4.1, we may assume without loss of generality that there exist $\varepsilon_n > 0$ with $r_\zeta > \varepsilon_n$ for all $n \in \mathbb{N}$ and $\zeta \in \varphi_{n;}(\mathcal{R}_n)$. In particular, $\mathcal{C}_n = \overline{\varphi_{n;}(\mathcal{R}_n)}$ is a compact subset of $\mathcal{R}$ for all $n \in \mathbb{N}$. By lemma 4.2, we may compute a digital Riemann surface $\mathcal{T}_{k_n} \subseteq \mathcal{S}_{k_n}$ with $\psi_{k_n;}(\mathcal{T}_n) \supseteq \xi(\mathcal{C}_n)$. We next increase k_n further until there exists a digital covering $\xi_n : \mathcal{C}_n \to \mathcal{T}_{k_n} \subseteq \mathcal{S}_{k_n}$ which commutes with $\xi \circ \varphi_{n;} = \psi_{k_n;} \circ \xi_n$. On the one hand, the digital coverings $\xi_n : \mathcal{C}_n \to \mathcal{T}_{k_n}$, whose incarnations at a suitable scale are finite in number, can easily be computed. Using the predicate near, we also have an ultimate test for checking whether $\xi \circ \varphi_{n;} = \psi_{k_n;} \circ \xi_n$. Trying all values of n in parallel, we know that one of these tests will ultimately succeed. Increasing k_n still further so as to ensure that $k_0 < k_1 < \cdots$, this completes the construction of the digital representation of ξ. □

Remark 4.4. A representation $\check{\mathcal{R}} : n \mapsto (\mathcal{R}_n, \varphi_n)$ of a computable Riemann surface is said to be *proper* if there exist $\varepsilon_n > 0$ with $r_\zeta > \varepsilon_n$ for all $n \in \mathbb{N}$ and $\zeta \in \varphi_{n;}(\mathcal{R}_n)$. From the proof of proposition 4.3, it follows that it is not necessary to subdivide $\check{\mathcal{R}}$, provided that $\check{\mathcal{R}}$ is proper.

A *computable covering sequence* is a computable sequence

$$\mathcal{R}_0 \xrightarrow{\xi_0} \mathcal{R}_1 \xrightarrow{\xi_1} \mathcal{R}_2 \xrightarrow{\xi_2} \cdots \tag{4.2}$$

where each $\mathcal{R}_n$ is a computable Riemann surface and each $\xi_n : \mathcal{R}_n \to \mathcal{R}_{n+1}$ a computable covering. Let $\check{\mathcal{R}}_n : k \mapsto (\mathcal{R}_{n,k},\ \varphi_{n,k})$ be a proper representation of $\mathcal{R}_n$ for each n. By induction over n, and modulo reindexation of $\check{\mathcal{R}}_n$, we may construct a digital representation $(\check{\mathcal{R}}_n, \check{\mathcal{R}}_{n+1}, k \mapsto \xi_{n,k})$ for ξ_n, such that we have the following commutative diagram:

$$\begin{array}{ccccccc}
\mathcal{R}_{0,0} & \xrightarrow{\varphi_{0,0}} & \mathcal{R}_{0,1} & \xrightarrow{\varphi_{0,1}} & \mathcal{R}_{0,2} & \xrightarrow{\varphi_{0,2}} & \cdots \\
\downarrow{\scriptstyle \xi_{0,0}} & & \downarrow{\scriptstyle \xi_{0,1}} & & \downarrow{\scriptstyle \xi_{0,2}} & & \\
\mathcal{R}_{1,0} & \xrightarrow{\varphi_{1,0}} & \mathcal{R}_{1,1} & \xrightarrow{\varphi_{1,1}} & \mathcal{R}_{1,2} & \xrightarrow{\varphi_{1,2}} & \cdots \\
\downarrow{\scriptstyle \xi_{1,0}} & & \downarrow{\scriptstyle \xi_{1,1}} & & \downarrow{\scriptstyle \xi_{1,2}} & & \\
\vdots & & \vdots & & \vdots & &
\end{array}$$

In particular, we obtain a new computable Riemann surface

$$\begin{aligned}
\mathcal{R} &= \lim \mathcal{R}_0 \xrightarrow{\xi_0} \mathcal{R}_1 \xrightarrow{\xi_1} \cdots \\
&:= \lim \mathcal{R}_{0,0} \overset{\varphi_{1,0} \circ \xi_{0,0}}{\longrightarrow} \mathcal{R}_{0,0} \overset{\varphi_{2,1} \circ \xi_{1,1}}{\longrightarrow} \cdots.
\end{aligned}$$

We call $\mathcal{R}$ the *limit* of the computable covering sequence (4.2). This limit satisfies the following universal property:

PROPOSITION 4.5. *For every Riemann surface $\mathcal{S}$ and coverings $\sigma_n : \mathcal{R}_n \to \mathcal{S}$, there exists a unique covering $\rho : \mathcal{R} \to \mathcal{S}$ with $\sigma_n = \rho \circ \xi_{n;}$ for all n. Moreover, if $\mathcal{S}$ is computable and the σ_n are given by a computable mapping, then ρ is also computable and we may compute it as a function of $\mathcal{S}$ and the σ_n.* □

4.2. Disjoint unions and covering products

Let $\mathcal{R}$ and $\mathcal{S}$ be two digital Riemann surfaces which are encoded at the same scale λ. We define their disjoint union $\mathcal{R} \amalg \mathcal{S}$ by

$$\begin{aligned}
\lambda_{\mathcal{R}\amalg\mathcal{S}} &= \lambda \\
A_{\mathcal{R}\amalg\mathcal{S}} &= A_{\mathcal{R}} \amalg A_{\mathcal{S}} \\
\pi_{\mathcal{R}\amalg\mathcal{S}}(a) &= \begin{cases} \pi_{\mathcal{R}}(a) & \text{if } a \in A_{\mathcal{R}} \\ \pi_{\mathcal{S}}(a) & \text{if } a \in A_{\mathcal{S}} \end{cases} \\
a \boxplus_{\mathcal{R}\amalg\mathcal{S}} b &\Leftrightarrow (a, b \in A_{\mathcal{R}} \wedge a \boxplus_{\mathcal{R}} b) \vee (a, b \in A_{\mathcal{S}} \wedge a \boxplus_{\mathcal{S}} b)
\end{aligned}$$

It is not hard to verify that this construction does not depend on λ and that $\mathcal{R} \amalg \mathcal{S}$ is indeed a digital Riemann surface. We have natural inclusions $\iota_1 \colon \mathcal{R} \to \mathcal{R} \amalg \mathcal{S}$ and $\iota_2 \colon \mathcal{S} \to \mathcal{R} \amalg \mathcal{S}$. The disjoint union satisfies the following universal property:

PROPOSITION 4.6. *Given any digital Riemann surface $\mathcal{T}$ with digital coverings $\xi_1 \colon \mathcal{R} \to \mathcal{T}$ and $\xi_2 \colon \mathcal{S} \to \mathcal{T}$, there exists a unique covering $\xi = \xi_1 \amalg \xi_2 \colon \mathcal{R} \amalg \mathcal{S} \to \mathcal{T}$ with $\xi_1 = \xi \circ \iota_1$ and $\xi_2 = \xi \circ \iota_2$. Moreover, ξ is a digital covering which can be computed as a function of $\mathcal{T}$, ξ_1 and ξ_2.* □

Similarly, we define the *covering product* $\mathcal{R} \rtimes \mathcal{S}$ of $\mathcal{R}$ and $\mathcal{S}$ by taking

$$\begin{aligned} \lambda_{\mathcal{R} \rtimes \mathcal{S}} &= \lambda \\ A_{\mathcal{R} \rtimes \mathcal{S}} &= \{(a, b) \in A_{\mathcal{R}} \times A_{\mathcal{S}} \colon \pi_{\mathcal{R}}(a) = \pi_{\mathcal{S}}(b)\} \\ \pi_{\mathcal{R} \rtimes \mathcal{S}}(a, b) &= \pi_{\mathcal{R}}(a) = \pi_{\mathcal{S}}(b) \\ (a, b) \sqcap_{\mathcal{R} \rtimes \mathcal{S}} (a', b') &\Leftrightarrow a \sqcap_{\mathcal{R}} a' \wedge b \sqcap_{\mathcal{S}} b' \end{aligned}$$

We have natural digital coverings $\pi_1 \colon \mathcal{R} \rtimes \mathcal{S} \to \mathcal{R}$ and $\pi_2 \colon \mathcal{R} \rtimes \mathcal{S} \to \mathcal{S}$ which are not necessarily surjective. The covering product does satisfy the following universal property:

PROPOSITION 4.7. *Given any digital Riemann surface $\mathcal{T}$ with digital coverings $\xi_1 \colon \mathcal{T} \to \mathcal{R}$ and $\xi_2 \colon \mathcal{T} \to \mathcal{S}$, then there exists a unique covering $\xi = \xi_1 \rtimes \xi_2 \colon \mathcal{T} \to \mathcal{R} \rtimes \mathcal{S}$ with $\xi_1 = \pi_1 \circ \xi$ and $\xi_2 = \pi_2 \circ \xi$. Moreover, ξ is a digital covering which can be computed as a function of $\mathcal{T}$, ξ_1 and ξ_2.* □

Let $\mathcal{R}$ and $\mathcal{S}$ be computable Riemann surfaces represented by $n \mapsto (\mathcal{R}_n, \varphi_n)$ resp. $n \mapsto (\mathcal{S}_n, \psi_n)$. The *disjoint union* of $\mathcal{R}$ and $\mathcal{S}$ is the computable Riemann surface represented by the sequence $n \mapsto (\mathcal{R}_n \amalg \mathcal{S}_n, \varphi_n \amalg \psi_n)$. The sequences $n \mapsto (\iota_1 \colon \mathcal{R}_n \to \mathcal{R}_n \amalg \mathcal{S}_n)$ and $n \mapsto (\iota_2 \colon \mathcal{S}_n \to \mathcal{R}_n \amalg \mathcal{S}_n)$ determine computable immersions $\mathcal{R} \to \mathcal{R} \amalg \mathcal{S}$ and $\mathcal{S} \to \mathcal{R} \amalg \mathcal{S}$ and the universal properties for $\mathcal{R}_n \amalg \mathcal{S}_n$ pass to the limit. Similarly, the *covering product* $\mathcal{R} \rtimes \mathcal{S}$ of $\mathcal{R}$ and $\mathcal{S}$ is the computable Riemann surfaces represented by the sequence $n \mapsto (\mathcal{R}_n \rtimes \mathcal{S}_n, \varphi_n \rtimes \psi_n)$. Again, we have natural computable coverings $\pi_1 \colon \mathcal{R} \rtimes \mathcal{S} \to \mathcal{R}$ and $\pi_2 \colon \mathcal{R} \rtimes \mathcal{S} \to \mathcal{S}$ which satisfy the universal property for products.

PROPOSITION 4.8. *Let $\mathcal{R}$ and $\mathcal{S}$ be computable Riemann surfaces.*

a) If $\mathcal{R}$ and $\mathcal{S}$ are delimited, then so are $\mathcal{R} \amalg \mathcal{S}$ and $\mathcal{R} \rtimes \mathcal{S}$.

b) If $\mathcal{R}$ and $\mathcal{S}$ have computable branches, then so have $\mathcal{R} \amalg \mathcal{S}$ and $\mathcal{R} \rtimes \mathcal{S}$.

Proof. All properties are easy. For instance, given $\zeta \in \mathcal{R} \rtimes \mathcal{S}$, we have

$$r_\zeta = \min(r_{\pi_1(\zeta)}, r_{\pi_2(\zeta)}).$$ □

4.3. Quotient spaces and gluing at a point

Let $\mathcal{R} = \lim \mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$ be a computable Riemann surface and $\eqsim \subseteq (\mathcal{R}^{\mathrm{com}})^2$ a sequentially enumerable relation with $\zeta \eqsim \xi \Rightarrow \pi(\zeta) = \pi(\xi)$. In particular, we may compute a computable sequence $k \mapsto E_k$, where each E_k is a pair $(\zeta_k, \xi_k) \in (\mathcal{R}_{n_k}^{\mathrm{com}})^2$ such that $(\varphi_{n_k;n}(\zeta_k), \psi_{n_k;n}(\zeta_k))$ is the k-th pair in the enumeration of $\eqsim$.

For each $n \in \mathbb{N}$, let $\sim_n$ be the smallest equivalence relation on $\mathcal{R}_n$ generated by the relations $\varphi_{n_k;n}(\zeta_k) \sim_n \psi_{n_k;n}(\xi_k)$ for $n_k \leqslant n$ and $k \leqslant n$. Setting $\mathcal{S}_n = \mathcal{R}_n / \sim_n$, we have natural computable coverings $\pi_n \colon \mathcal{R}_n \to \mathcal{S}_n$ and $\psi_n = (\pi_{n+1} \circ \varphi_n) / \sim_n \colon \mathcal{S}_n \to \mathcal{S}_{n+1}$. Let $\mathcal{S} = \mathcal{R} / \eqsim$ be the limit of $\mathcal{S}_0 \xrightarrow{\psi_0} \mathcal{S}_1 \xrightarrow{\psi_1} \cdots$. The mappings π_n induce a computable surjective covering $\pi_\eqsim \colon \mathcal{R} \to \mathcal{S}$. For every $\zeta, \xi \in \mathcal{R}$ we have $\zeta \eqsim \xi \Rightarrow \pi_\eqsim(\zeta) = \pi_\eqsim(\xi)$. It is not hard to verify the following universal property of $\mathcal{S}$:

PROPOSITION 4.9. *Given a Riemann surface $\mathcal{T}$ and a covering $\tilde{\pi}: \mathcal{R} \to \mathcal{T}$ with $\zeta \eqsim \xi \Rightarrow \tilde{\pi}(\zeta) = \tilde{\pi}(\xi)$, there exists a unique covering $\xi: \mathcal{S} \to \mathcal{T}$ with $\tilde{\pi} = \xi \circ \pi_{\eqsim}$. Moreover, if $\mathcal{T}$ and $\tilde{\pi}$ are computable, then so is ξ and we may compute it as a function of $\mathcal{T}$ and $\tilde{\pi}$.* □

Let us now consider two computable Riemann surfaces $\mathcal{R}$ and $\mathcal{S}$. Given $\zeta \in \mathcal{R}^{\mathrm{com}}$ and $\xi \in \mathcal{R}^{\mathrm{com}}$ with $\pi_{\mathcal{R}}(\zeta) = \pi_{\mathcal{S}}(\xi)$, consider the relation $\eqsim$ on $\mathcal{R} \amalg \mathcal{S}$ which is reduced to the singleton $\{(\zeta, \xi)\}$. We call $\mathcal{R}_{\zeta} \bowtie_{\xi} \mathcal{S} = (\mathcal{R} \amalg \mathcal{S})/\eqsim$ the *join* of $\mathcal{R}$ and $\mathcal{S}$ at (ζ, ξ). If ζ and ξ are not important, or clear from the context, then we also write $\mathcal{R} \bowtie \mathcal{S}$ for $\mathcal{R}_{\zeta} \bowtie_{\xi} \mathcal{S}$. We will denote the natural coverings $\mathcal{R} \to \mathcal{R} \bowtie \mathcal{S}$ and $\mathcal{S} \to \mathcal{R} \bowtie \mathcal{S}$ by θ_1 resp. θ_2.

PROPOSITION 4.10. *Assume that $\mathcal{R}$ and $\mathcal{S}$ are connected. Then $\theta_1(\mathcal{R}) \cap \theta_2(\mathcal{S})$ is connected.*

Proof. Assume for contradiction that $\theta_1(\mathcal{R}) \cap \theta_2(\mathcal{S})$ is not connected and let $\mathcal{R} \bowtie \mathcal{S} = \mathcal{U} \amalg \mathcal{V}$, where $\mathcal{U}$ is the connected component of $\theta_1(\zeta) = \theta_2(\xi)$. Then we may define an equivalence relation $\sim'$ on $\mathcal{R} \amalg \mathcal{S}$ by $\zeta' \sim' \xi' \Leftrightarrow \zeta' = \xi' \vee \theta_1(\zeta') = \theta_2(\xi') \in \mathcal{U}$. The quotient set $\mathcal{T} = (\mathcal{R} \amalg \mathcal{S})/\sim'$ has a natural structure of a Riemann surface and there exists a natural covering $\mathcal{T} \to \mathcal{R} \bowtie \mathcal{S}$. By the universal property of $\mathcal{R} \bowtie \mathcal{S}$, it follows that $\mathcal{T} \cong \mathcal{R} \bowtie \mathcal{S}$, which is impossible. □

The proposition ensures in particular that we may apply the following classical theorem:

THEOREM 4.11. (VAN KAMPEN) *Let A and B be path-connected topological spaces, such that $A \cap B$ is non-empty and path connected. Denote by ι_1 and ι_2 the natural inclusions of $A \cap B$ in A resp. B. Then the homotopy group of $A \cup B$ is given by*

$$\pi_1(A \cup B) = (\pi_1(A) * \pi_1(B))/H,$$

where H is the normal subgroup of the free product $\pi_1(A) * \pi_2(B)$ of $\pi_1(A)$ and $\pi_2(B)$ generated by elements $\iota_1(\alpha)\, \iota_2(\alpha^{-1})$ with $\alpha \in \pi_1(A \cap B)$.

COROLLARY 4.12. *If $\mathcal{R}$ and $\mathcal{S}$ are simply connected computable Riemann surfaces, then so is $\mathcal{R} \bowtie \mathcal{S}$.*

4.4. Computable rooted Riemann surfaces

A *broken line path* is a finite sequence $\delta = (\delta_1, \ldots, \delta_l) \in \mathbb{C}^l$ and we write

$$\begin{aligned} |\delta| &= l \\ \|\delta\| &= \delta_1 + \cdots + \delta_l \end{aligned}$$

Intuitively speaking, δ corresponds to a path $0 \to \delta_1 \to \cdots \to \delta_1 + \cdots + \delta_l$. We write $\mathbb{P}$ for the set of broken line paths and denote by

$$\begin{aligned} \mathbb{P}^{\mathrm{dig}} &= \{(\delta_1, \ldots, \delta_l) \in \mathbb{P} : \delta_1, \ldots, \delta_l \in \mathbb{C}^{\mathrm{dig}}\} \\ \mathbb{P}^{\mathrm{com}} &= \{(\delta_1, \ldots, \delta_l) \in \mathbb{P} : \delta_1, \ldots, \delta_l \in \mathbb{C}^{\mathrm{com}}\} \end{aligned}$$

the subsets of *digital* and *computable* paths. The empty path is denoted by ϵ. We say that δ' is a *truncation* of δ and write $\delta' \trianglelefteqslant \delta$ if $\delta' = (\delta_1, \ldots, \delta_i)$ for some $i \leqslant |\delta|$. Given two paths $\delta, \delta' \in \mathbb{P}$, we write $\delta + \delta' = (\delta_1, \ldots, \delta_{|\delta|}, \delta'_1, \ldots, \delta'_{|\delta'|})$. When no confusion is possible, paths of length 1 will be identified with numbers. If $\delta \neq \epsilon$, then we will also write $\delta^{\backslash}$ for the path $(\delta_1, \ldots, \delta_{|\delta|-1})$.

A Riemann surface $\mathcal{R}$ is said to be *rooted* if it admits a special element $\bullet \in \mathcal{R}$ called the *root* of $\mathcal{R}$. If $\mathcal{R}$ is also computable and $\bullet \in \mathcal{R}^{\mathrm{com}}$, then we call $\mathcal{R}$ a computable rooted Riemann surface. Unless explicitly stated otherwise, we will always assume that rooted Riemann surfaces are connected. A root-preserving covering between two rooted Riemann surfaces will be called a *rooted covering*. We denote by $\mathbb{S}_{\bullet}^{\mathrm{com}}$ the class of computable rooted Riemann surfaces. Given $\mathcal{R} \in \mathbb{S}_{\bullet}^{\mathrm{com}}$, we have an additional method $\bullet: () \to \mathcal{R}^{\mathrm{com}}$ in the signature of $\mathcal{R}^{\mathrm{com}}$.

Let $\mathcal{R}$ be a computable rooted Riemann surface. We define the *path domain* $\mathbb{P}_\mathcal{R}$ of $\mathcal{R}$ to be the set of $\delta = (\delta_1, ..., \delta_l) \in \mathbb{P}$, so that

$$\begin{aligned} \bullet + \epsilon &= \bullet \\ \bullet + (\delta_1) &= (\bullet + \epsilon) + \delta_1 \\ &\vdots \\ \bullet + (\delta_1, ..., \delta_l) &= (\bullet + (\delta_1, ..., \delta_{l-1})) + \delta_l \end{aligned}$$

are all well-defined. We will also write $\varepsilon_\mathcal{R} = \bullet_\mathcal{R} + \varepsilon$. The *digital* and *computable path domains* of $\mathcal{R}$ are defined by

$$\begin{aligned} \mathbb{P}_\mathcal{R}^{\mathrm{dig}} &= \mathbb{P}_\mathcal{R} \cap \mathbb{P}^{\mathrm{dig}} \\ \mathbb{P}_\mathcal{R}^{\mathrm{com}} &= \mathbb{P}_\mathcal{R} \cap \mathbb{P}^{\mathrm{com}} \end{aligned}$$

We notice that $\mathbb{P}_\mathcal{R}^{\mathrm{dig}}$ is an abstract computable set with a computable equality test, whereas $\mathbb{P}_\mathcal{R}^{\mathrm{com}}$ is only an effective set. A broken line path $\delta = (\delta_1, ..., \delta_l) \in \mathbb{P}_\mathcal{R}$ naturally induces a continuous path $\phi_{\delta,\mathcal{R}}: [0,1] \to \mathcal{R}$ by setting

$$\phi_{\delta,\mathcal{R}}((i+t)/n) = (\delta_1, ..., \delta_{i-1}, t\,\delta_i)_\mathcal{R}$$

for $i \in \{0, ..., l-1\}$ and $t \in [0,1]$. This path is rooted in the sense that $\phi_{\delta,\mathcal{R}}(0) = \bullet_\mathcal{R}$.

PROPOSITION 4.13. *Let $\mathcal{R}$ and $\mathcal{S}$ be computable rooted Riemann surfaces. Then there exists at most one rooted covering $\psi: \mathcal{R} \to \mathcal{S}$. Such a covering is necessarily computable and computable as a function of $\mathcal{R}$ and $\mathcal{S}$.*

Proof. Assume that there exists a covering $\psi: \mathcal{R} \to \mathcal{S}$. By continuity, it suffices to show how to compute $\psi(\zeta)$ for all $\zeta \in \mathcal{R}^{\mathrm{dig}}$. Since $\mathcal{R}$ is connected, there exists a path $\delta_\zeta \in \mathbb{P}_\mathcal{R}^{\mathrm{dig}}$ with $\zeta = (\delta_\zeta)_\mathcal{R}$. Given ζ, we claim that we may compute such a path δ_ζ. Indeed, the set $\mathbb{P}_\mathcal{R}^{\mathrm{dig}}$ is enumerable and, given $\delta \in \mathbb{P}_\mathcal{R}^{\mathrm{dig}}$, we may ultimately test whether $\delta_\mathcal{R} = \zeta$. We perform these ultimate tests in parallel, for all $\delta \in \mathbb{P}_\mathcal{R}^{\mathrm{dig}}$, until one of them succeeds. Since $\mathcal{S}$ is connected, we have $\mathbb{P}_\mathcal{R} \subseteq \mathbb{P}_\mathcal{S}$, so our claim implies $\psi(\zeta) = \psi((\delta_\zeta)_\mathcal{R}) = (\delta_\zeta)_\mathcal{S}$. □

PROPOSITION 4.14. *Let $\mathcal{R}$ be a computable rooted Riemann surface and assume that $\mathbb{P}$ is given the natural topology of $\mathbb{C}^0 \amalg \mathbb{C}^1 \amalg \mathbb{C}^2 \amalg \cdots$. Then*

a) *$\mathbb{P}_\mathcal{R}^{\mathrm{dig}}$, $\mathbb{P}_\mathcal{R}^{\mathrm{com}}$ and $\mathbb{P}_\mathcal{R}$ are open subsets of $\mathbb{P}^{\mathrm{dig}}$, $\mathbb{P}^{\mathrm{com}}$ resp. $\mathbb{P}$.*

b) *$\mathbb{P}_\mathcal{R}^{\mathrm{com}}$ is a dense subset of $\mathbb{P}_\mathcal{R}$ and $\mathbb{P}_\mathcal{R}^{\mathrm{dig}}$ is a dense subset of both $\mathbb{P}_\mathcal{R}$ and $\mathbb{P}_\mathcal{R}^{\mathrm{dig}}$.*

Proof. Let us prove the proposition by induction over l for each of the subspaces $\mathbb{P}_\mathcal{R}^{\mathrm{dig}} \cap \mathbb{C}^l$, $\mathbb{P}_\mathcal{R}^{\mathrm{com}} \cap \mathbb{C}^l$, etc. The assertions are clear for $l = 0$. Assume that $U_l = \mathbb{P}_\mathcal{R} \cap \mathbb{C}^l$ is open, with $U_l^{\mathrm{com}} = \mathbb{P}_\mathcal{R}^{\mathrm{com}} \cap \mathbb{C}^l$ as a dense subset. We have

$$U_{l+1} = \mathbb{P}_\mathcal{R} \cap \mathbb{C}^{l+1} = \{\delta \in U_l \times \mathbb{C}: |\delta_{l+1}| < \rho(\delta^\backslash)\},$$

where $\rho: U_l \to \mathbb{R}^>; \delta \mapsto r_{\delta_\mathcal{R}}$. Now the restriction $\rho_{|U_l^{\mathrm{com}}}: U_l \to \mathbb{R}^{\mathrm{lcom},>}$ is computable, so ρ is lower continuous, by theorem 2.3. Assume that $\delta \in U_{l+1}$ and let $\varepsilon = \rho(\delta^\backslash) - |\delta_{l+1}|$. Then $\delta^\backslash$ admits an open neighbourhood $V \subseteq U_l$ with $\rho(\eta) > |\delta_{l+1}| + \varepsilon/2$ for all $\eta \in V$. Consequently, $V \times \mathcal{B}_{\delta_{l+1}, \varepsilon/2} \subseteq U_{l+1}$ is an open neighbourhood of δ. This proves that U_{l+1}, U_{l+1}^{com} and $U_{l+1}^{\mathrm{dig}} = \mathbb{P}_\mathcal{R}^{\mathrm{dig}} \cap \mathbb{C}^l$ are open subsets of $\mathbb{P}$, $\mathbb{P}^{\mathrm{dig}}$ resp. $\mathbb{P}^{\mathrm{com}}$. In order to show that U_{l+1}^{com} is a dense subset of V, it suffices to prove that any open ball $V \subseteq U_{l+1}$ intersects U_{l+1}^{com}. Now $V^\backslash = \{\delta^\backslash: \delta \in V\}$ is an open ball of U_l, which intersects U_l^{com}, say in δ. Furthermore, $\{\varepsilon \in \mathbb{C}: \delta + \varepsilon \in V\}$ is a disk with radius $0 < \rho < r_{\delta_\mathcal{R}}$. Taking $\varepsilon \in \mathbb{C}^{\mathrm{dig}}$ with $|\varepsilon| < \rho$, we thus have $\delta + \varepsilon \in V \cap U_{l+1}^{\mathrm{com}}$. The other density statements are proved similarly. □

PROPOSITION 4.15. *Let $\mathcal{R}$ be the limit of a computable covering sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$.*

a) If $\mathcal{R}_0, \mathcal{R}_1, \ldots$ are all connected, then so is $\mathcal{R}$.

b) If $\mathcal{R}_0, \mathcal{R}_1, \ldots$ are all simply connected, then so is $\mathcal{R}$.

Proof. Assume that $\mathcal{R} = \mathcal{U} \amalg \mathcal{V}$ where $\mathcal{U}$ and $\mathcal{V}$ are non-empty open sets. Then $\varphi_{n;}(\mathcal{R}_n)$ both intersects $\mathcal{U}$ and $\mathcal{V}$ for sufficiently large n. Consequently, $\mathcal{R}_n = \varphi_{n;}^{-1}(\mathcal{U}) \amalg \varphi_{n;}^{-1}(\mathcal{V})$ is not connected. This proves (a). As to (b), assume that $\mathcal{R}_0, \mathcal{R}_1, \ldots$ are simply connected and consider a loop $\gamma : [0,1] \to \mathcal{R}$ with $\gamma(0) = \gamma(1)$. Then $\operatorname{im}\gamma$ is compact, so $\varphi_{k;}(\mathcal{R}_k) \supseteq \operatorname{im}\gamma$ for a sufficiently large k. In a similar way as in lemma 4.2, we may find a $n \geqslant k$ such that the restriction of $\varphi_{n;}$ to $\varphi_{k;n}(\mathcal{R}_k)$ is a homeomorphism. But then $\varphi_{n;}^{-1} \circ \gamma$ is a loop in $\mathcal{R}_n$ which may be contracted to a point. Composing with $\varphi_{n;}$, we obtain a contraction of γ into a point. □

PROPOSITION 4.16. *Given a not necessary connected computable rooted Riemann surface $\mathcal{R}$, we may compute the connected component $\mathcal{R}^\bullet$ of the root.*

Proof. Let $\mathcal{R} = \lim \mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$. Modulo taking a subsequence, we may assume without loss of generality that $\mathcal{R}_0$ contains a point $\bullet_{\mathcal{R}_0}$ with $\bullet_{\mathcal{R}} = \varphi_{0;}(\bullet_{\mathcal{R}_0})$. It is easy to compute the connected component $\mathcal{R}_n^\bullet$ of $\bullet_{\mathcal{R}_n} = \varphi_{0;n}(\bullet_{\mathcal{R}_0})$ in $\mathcal{R}_n$ for each $n \in \mathbb{N}$. By proposition 4.15(a), the limit of the sequence $\mathcal{R}_n^\bullet$ yields $\mathcal{R}^\bullet$. □

4.5. Organic Riemann surfaces

Assume now that we are given an enumerable set of paths $\Delta \subseteq \mathbb{P}^{\mathrm{dig}}$ and a computable mapping $r : \Delta \to \bar{\mathbb{R}}^{\mathrm{lcom},>}$ such that, given $\delta \in \Delta$ and $\varepsilon \in \mathbb{C}^{\mathrm{dig}}$, we have $\delta + \varepsilon \in \Delta$ if and only if $|\varepsilon| < r_\delta$. Reordering terms when necessary, we may assume that Δ is presented as an enumeration $\Delta = \{\delta_0, \delta_1, \ldots\}$ such that $\delta_i \trianglelefteq \delta_j \Rightarrow i \leqslant j$ for all $i, j \in \mathbb{N}$. Assume that we are also given a number $z_0 \in \mathbb{C}^{\mathrm{com}}$; we call (z_0, Δ, r) an *organic triple*.

Let us define a computable rooted covering sequence $\mathcal{O}_0 \xrightarrow{\omega_0} \mathcal{O}_1 \xrightarrow{\omega_1} \cdots$, such that $\delta_i, \delta_i + \varepsilon \in \mathbb{P}_{\mathcal{O}_n}^{\mathrm{dig}}$ for all $i \leqslant n$ and $\varepsilon \in \mathbb{C}^{\mathrm{dig}}$ with $\varepsilon < r_{(\delta_i)_{\mathcal{O}_n}}$. We proceed by induction over $n \in \mathbb{N}$. Denote by $\mathcal{S}_n$ the computable ball with center $z_0 + \|\delta_n\|$ and radius r_{δ_n}. We start with $\mathcal{O}_0 = \mathcal{S}_0$ and $\bullet_{\mathcal{S}_0} = z_0$. Assume that $\mathcal{O}_n$ has been constructed. Then the path $\delta_l{}^\backprime$ necessarily occurs before δ_l in our enumeration, whence $\delta_l = \delta_l{}^\backprime + (\delta_l)_{|\delta_l|} \in \mathbb{P}_{\mathcal{O}_i}^{\mathrm{dig}}$, so that $\zeta_n = (\delta_l)_{\mathcal{O}_n} \in \mathcal{O}_n^{\mathrm{com}}$ and $z_n = \pi(\zeta_n) \in \mathbb{C}^{\mathrm{com}}$ are well-defined. Now we take

$$\mathcal{O}_{n+1} = \mathcal{O}_n \ {}_{\zeta_n}\bowtie_{z_n} \mathcal{S}_{n+1}$$

with root $\theta_1(\bullet_{\mathcal{O}_n})$ and $\omega_n = \theta_1$. By construction, $\delta_i + \varepsilon \in \mathbb{P}_{\mathcal{O}_{n+1}}^{\mathrm{dig}}$ for all $i \leqslant n+1$ and $\varepsilon \in \mathbb{C}^{\mathrm{dig}}$ with $\varepsilon < r_{\delta_i}$. Indeed, if $i \leqslant n$, then $(\delta_i + \varepsilon)_{\mathcal{O}_{n+1}} = \theta_1((\delta_i + \varepsilon)_{\mathcal{O}_n})$. If $i = n+1$, then $(\delta_l + \varepsilon)_{o_{n+1}} = \theta_2(z_n + \varepsilon)$. This completes the construction of our covering sequence. Its limit $\mathcal{O} = \mathcal{O}_{z_0,\Delta} = \mathcal{O}_{z_0,\Delta,r}$ is called the *organic Riemann surface* associated to (z_0, Δ, r). Organic Riemann surfaces are always simply connected, by corollary 4.12 and proposition 4.15. They satisfy two universal properties:

PROPOSITION 4.17. *Given a rooted Riemann surface $\mathcal{T}$ with $\pi(\bullet_{\mathcal{S}}) = z_0$ and $\mathbb{P}_{\mathcal{T}} \supseteq \Delta$, there exists a unique rooted covering $\psi : \mathcal{O} \to \mathcal{T}$. Moreover, if $\mathcal{T}$ is computable, then ψ is computable and computable as a function of $\mathcal{T}$.*

Proof. Let us show by induction over $n \in \mathbb{N}$ that there exists a unique rooted covering $\psi_n : \mathcal{O}_n \to \mathcal{T}_n$, where

$$\mathcal{T}_n = \bigcup_{i \leqslant n} \mathcal{B}_{(\delta_i)_{\mathcal{T}}, r_{(\delta_i)_{\mathcal{T}}}} \subseteq \mathcal{T}.$$

This is clear for $n = 0$. Assume that the assertion holds for a given $n \in \mathbb{N}$. There exists a covering

$$\sigma_{n+1}\colon \mathcal{S}_{n+1} \to \mathcal{B}_{(\delta_{n+1})_{\mathcal{T}}, r_{(\delta_{n+1})_{\mathcal{T}}}} \subseteq \mathcal{T}_{n+1}.$$

By the universal property of joins, it follows that there exists a rooted covering $\psi_{n+1}\colon \mathcal{O}_{n+1} \to \mathcal{T}_{n+1}$ with $\psi_{n+1} \circ \theta_1 = \psi_n$ and $\psi_{n+1} \circ \theta_2 = \sigma_{n+1}$. We obtain ψ by proposition 4.5 and we conclude by proposition 4.13. □

PROPOSITION 4.18. *Let (z_0, Δ, r) and (z_0, Δ', r') be organic triples with $\Delta \subseteq \Delta'$. Then there exists a unique rooted covering $\psi\colon \mathcal{O}_{z_0,\Delta,r} \to \mathcal{O}_{z_0,\Delta',r'}$, which is computable and computable as a function of (z_0, Δ, r) and (z_0, Δ', r').*

Proof. Notice that $r_\delta \leqslant r'_\delta$ for all $\delta \in \Delta$. Denote the counterparts of $\mathcal{O}_n$, $\mathcal{S}_n$, etc. in the construction of $\mathcal{O}_{z_0,\Delta',r'}$ by $\mathcal{O}'_n$, $\mathcal{S}'_n$, etc. For each $n \in \mathbb{N}$, there exists a computable $k_n \in \mathbb{N}$ such that $\delta_0, \ldots, \delta_n \in \{\delta'_0, \ldots, \delta'_{k_n}\}$. By a similar induction as in the proof of proposition 4.17, one shows that there exists a rooted covering $\psi_n\colon \mathcal{O}_n \to \mathcal{O}'_{k_n}$ for every $n \in \mathbb{N}$. Passing to the limit, we obtain ψ. □

Remark 4.19. If we only have a mapping $r\colon \Delta \to \bar{\mathbb{R}}^{\mathrm{lcom},>}$ such that $\delta + \varepsilon \in \Delta$ for any $\delta \in \Delta$ and $\varepsilon \in \mathbb{C}^{\mathrm{dig}}$ with $|\varepsilon| < r_\delta$, then we may still define $\mathcal{O}_{z_0,\Delta,r} = \mathcal{O}_{z_0,\Delta',r}$, where

$$\Delta' = \{(\delta_1, \ldots, \delta_l) \in \Delta\colon \forall i, |\delta_i| < r_{(\delta_1,\ldots,\delta_{i-1})}\}$$

is an enumerable set, which fulfills the stronger requirement that $\delta + \varepsilon \in \Delta'$ if and only if $|\varepsilon| < r_\delta$.

4.6. Universal computable covering spaces

Let $\mathcal{R}$ be a computable rooted Riemann surface. The construction of organic Riemann surfaces may in particular be applied for $\Delta = \mathbb{P}_{\mathcal{R}}^{\mathrm{dig}}$, $r_\delta = r_{\delta_{\mathcal{R}}}$ and $z_0 = \pi(\bullet_{\mathcal{R}})$. In that case, we denote $\mathcal{R}^\sharp = \mathcal{O}_{z_0,\Delta,r}$ and it can be proved that $\mathbb{P}_{\mathcal{R}^\sharp} = \mathbb{P}_{\mathcal{R}}$. In the construction of $\mathcal{R}^\sharp_n = \mathcal{O}_n$, each $\mathcal{S}_n$ is naturally isomorphic to the ball $\mathcal{B}_{(\delta_n)_{\mathcal{R}}, r_{(\delta_n)_{\mathcal{R}}}} \subseteq \mathcal{R}$. By induction over n, each $\mathcal{R}_n$ therefore comes with a natural rooted covering $\flat_n\colon \mathcal{R}^\sharp_n \to \mathcal{R}$. Taking limits, we obtain a natural rooted covering $\flat\colon \mathcal{R}^\sharp \to \mathcal{R}$ and it is readily verified that $\flat(\delta_{\mathcal{R}^\sharp}) = \delta_{\mathcal{R}}$ for all $\delta \in \mathbb{P}$. The *universal computable covering space* $\mathcal{R}^\sharp$ admits the following universal properties:

PROPOSITION 4.20. *Given a rooted covering $\tau\colon \mathcal{T} \to \mathcal{R}$ with $\mathbb{P}_{\mathcal{T}} = \mathbb{P}_{\mathcal{R}}$, there exists a unique rooted covering $\psi\colon \mathcal{R}^\sharp \to \mathcal{T}$ and ψ satisfies $\flat = \tau \circ \psi$. If τ is computable, then ψ is computable and computable as a function of τ.*

Proof. With $\psi_n\colon \mathcal{R}^\sharp_n \to \mathcal{T}_n$ as in the proof of proposition 4.17, the universal property of joins implies that $\flat_n = \tau \circ \psi_n$ for all $n \in \mathbb{N}$. Taking limits for $n \to \infty$, we conclude that $\flat = \tau \circ \psi$. □

PROPOSITION 4.21. *Given a computable rooted covering $\psi\colon \mathcal{R} \to \mathcal{S}$, there exists a unique rooted covering $\psi^\sharp\colon \mathcal{R}^\sharp \to \mathcal{S}^\sharp$ and we have $\psi \circ \flat_{\mathcal{R}} = \flat_{\mathcal{S}} \circ \psi^\sharp$. Moreover, $\psi^\sharp$ is computable and computable as a function of ψ.*

Proof. The existence, uniqueness and computability properties of $\psi^\sharp$ follow from proposition 4.18. The rooted coverings $\psi \circ \flat_{\mathcal{R}}$ and $\flat_{\mathcal{S}} \circ \psi^\sharp$ are identical by proposition 4.13. □

PROPOSITION 4.22. *Let $\varphi\colon \mathcal{R} \to \mathcal{S}$ be a root-preserving computable covering between two rooted computable Riemann surfaces $\mathcal{R}$ and $\mathcal{S}$ with $\mathbb{P}_{\mathcal{R}} \supseteq \mathbb{P}_{\mathcal{S}}$. Then any path $\gamma\colon [0,1] \to \mathcal{S}$ with $\gamma(0) = \bullet_{\mathcal{S}}$ can be lifted uniquely to a path $\tilde{\gamma}\colon [0,1] \to \mathcal{R}$ with $\tilde{\gamma}(0) = \bullet_{\mathcal{R}}$.*

Proof. Let $\varepsilon = \min\{r_{\gamma(t)} : t \in [0,1]\}$. Since γ is uniformly continuous, we may approximate γ by a broken line path $\delta \in \mathbb{P}^{\mathrm{dig}}$ with

$$\|\gamma - \phi_{\delta,\mathcal{S}}\| = \min\{|\gamma(t) - \phi_{\delta,\mathcal{S}}(t)| : t \in [0,1]\} < \varepsilon/2.$$

Since $\operatorname{im} \phi_{\delta,\mathcal{S}} \subseteq \operatorname{im} \gamma + \mathcal{B}_{\varepsilon/2} \subseteq \mathcal{S}$, we have $\delta \in \mathbb{P}^{\mathrm{dig}}_{\mathcal{S}} \subseteq \mathbb{P}^{\mathrm{dig}}_{\mathcal{R}}$. Consequently, δ lifts to a path $\phi_{\delta,\mathcal{R}}$ on $\mathcal{R}$. Since $\mathbb{P}_{\mathcal{R}} \supseteq \mathbb{P}_{\mathcal{S}}$, we also have $r_\zeta \geqslant r_{\varphi(\zeta)}$ for all $\zeta \in \mathcal{R}$. Consequently, $\operatorname{im} \phi_{\delta,\mathcal{R}} + \mathcal{B}_{\varepsilon/2} \subseteq \mathcal{R}$, so that γ lifts to the path $\tilde{\gamma}(t) = \phi_{\delta,\mathcal{R}^\sharp}(t) + (\gamma(t) - \phi_{\delta,\mathcal{R}}(t))$: $[0, 1] \to \mathcal{R}$. □

COROLLARY 4.23. *$\mathcal{R}^\sharp$ is isomorphic to the universal covering space of $\mathcal{R}$.* □

4.7. Digital covering spaces

Let $\mathcal{R}$ be a rooted digital Riemann surface, encoded by $\check{\mathcal{R}} = (\lambda, A, \pi, \square)$. Assume that $\bullet_A \in A$ is such that $\bullet_{\mathcal{R}} \in \bar{\mathcal{S}}_{\bullet_A}$. In this section, we will then show that the universal covering space $\mathcal{R}^\sharp$ can be constructed in a more explicit way.

A *digital path* is a tuple $\delta = (\delta_1, \ldots, \delta_l)$ with $\delta_1, \ldots, \delta_l \in \{\pm 1, \pm \mathrm{i}\}$. We denote by $\mathbb{P}_A$ the set of digital paths δ on A, for which $\bullet_A, \bullet_A + \delta_1, \ldots, \bullet_A + \delta = \bullet_A + \delta_1 + \cdots + \delta_l \in A$. Given $\delta \in \mathbb{P}_A$, we write $\delta_A = \bullet_A + \delta \in A$. The set $\mathbb{P}_A$ comes with a natural projection $\pi : \mathbb{P}_A \to \mathbb{Z}[\mathrm{i}]$; $\delta \mapsto \pi(\delta_A)$ and a natural adjacency relation: $\delta \square \delta'$ if and only if $\delta = \delta' + \varepsilon$ or $\delta' = \delta + \varepsilon$ for some $\varepsilon \in \{\pm 1, \pm \mathrm{i}\}$.

Let $\mathbb{P}_{A,n}$ be the subset of $\mathbb{P}_A$ of paths of lengths $\leqslant n$. Then $\mathcal{P}_n = (\lambda, \mathbb{P}_{A,n}, \pi, \square)$ is a Riemann pasting and we denote by $\mathcal{R}^\sharp_n = \mathcal{P}^*_n = (\lambda, A^\sharp_n, \pi, \square)$ its associated digital Riemann surface. The root $\bullet_{\mathcal{R}}$ can be lifted to a root $\bullet_{\mathcal{R}^\sharp_n}$ of $\mathcal{R}^\sharp_n$ for $n \geqslant 2$ and the natural inclusions $i_n : \mathcal{P}_n \to \mathcal{P}_{n+1}$ induce natural rooted coverings $\iota_n : \mathcal{R}^\sharp_n \to \mathcal{R}^\sharp_{n+1}$ for $n \geqslant 2$.

PROPOSITION 4.24. *With the above notations the limit $\tilde{\mathcal{R}}^\sharp$ of $\mathcal{R}^\sharp_2 \xrightarrow{\iota_2} \mathcal{R}^\sharp_3 \xrightarrow{\iota_3} \cdots$ is isomorphic to the universal covering space $\mathcal{R}^\sharp$ of $\mathcal{R}$.*

Proof. In view of proposition 4.13, it suffices to prove that there exist rooted coverings $\mathcal{R}^\sharp \to \tilde{\mathcal{R}}^\sharp$ and $\tilde{\mathcal{R}}^\sharp \to \mathcal{R}^\sharp$. Since $\mathbb{P}_{\mathcal{R}^\sharp_n} \subseteq \mathbb{P}_{\mathcal{R}} = \mathbb{P}_{\mathcal{R}^\sharp}$, we have natural rooted coverings $\mathcal{R}^\sharp_n \to \mathcal{R}^\sharp$. This yields a rooted covering $\tilde{\mathcal{R}}^\sharp \to \mathcal{R}^\sharp$ when passing to the limit. Conversely, any path $\delta \in \mathbb{P}_{\mathcal{R}^\sharp}$ can naturally be approximated by a digital path $\tilde{\delta} \in \mathbb{P}_A$, in the sense that $\|\phi_{\delta,\mathcal{R}} - \phi_{(c_{\bullet_A} - \bullet_{\mathcal{R}}) + \lambda\tilde{\delta},\mathcal{R}}\| < 3\lambda/2$, after possible reparameterization of $\phi_{\delta,\mathcal{R}}$. Setting $n = |\tilde{\delta}|$, we then have $\delta \in \mathbb{P}_{\mathcal{R}^\sharp_{n+2}} \subseteq \mathbb{P}_{\tilde{\mathcal{R}}^\sharp}$, which shows the existence of a rooted covering $\mathcal{R}^\sharp \to \tilde{\mathcal{R}}^\sharp$. □

PROPOSITION 4.25. *The mappings ι_n are injective.*

Proof. The theoretical definition of the normalization of a Riemann pasting also applies in the case of $(\lambda, \mathbb{P}_A, \pi, \square)$ when $\mathbb{P}_A$ is infinite and one has $\delta^* \sim \varepsilon^*$ resp. $\delta^* \square^* \varepsilon^*$ for δ, $\varepsilon \in \mathbb{P}_A$ if and only if these relations hold for a sufficiently large n with $\delta, \varepsilon \in \mathbb{P}_{A,n}$. For each $a \in A^\sharp$ there exists a digital path δ_a of smallest length with $(\delta_a)_{A^\sharp} = a$ and we denote this length by $|a|$. Let $\mathcal{B}_n = \{a \in A^\sharp : |a| \leqslant n\}$ for each $n \in \mathbb{N}$, so that $\mathcal{B}_0 \subseteq \mathcal{B}_1 \subseteq \cdots$. For every $a \in \mathcal{B}_n$, the path δ_a induces an element $a_n = (\delta_a)_{A^\sharp_n}$ of $A^\sharp_n$, which shows that the natural rooted covering $\mathcal{R}^\sharp_n \to \mathcal{B}_n$ is surjective. Since $\mathcal{R}^\sharp_n$ is obtained by gluing a finite number of squares to $\mathcal{R}^\sharp_{n-1}$, corollary 4.12 implies that $\mathcal{R}^\sharp_n$ is simply connected, by induction over n. Consequently, $\mathcal{R}^\sharp_n$ is isomorphic to $\mathcal{B}_n$ for each n, and $\mathcal{R}^\sharp_0 \subseteq \mathcal{R}^\sharp_1 \subseteq \cdots$, as desired. □

COROLLARY 4.26. *Let $\mathcal{R}$ be a rooted digital Riemann surface and let $\delta, \delta' \in \mathbb{P}_{\mathcal{R}}^{\mathrm{com}}$ be such that $\|\delta\| = \|\delta'\|$. Then there exists an algorithm to decide whether $\delta_{\mathcal{R}}$ and $\delta'_{\mathcal{R}}$ are homotopic.*

Proof. Since $\mathcal{R}^\sharp$ has computable branches by proposition 4.25, we have a test for deciding whether $\delta_{\mathcal{R}^\sharp} = \delta'_{\mathcal{R}^\sharp}$. Now this is the case if and only if $\delta_{\mathcal{R}}$ and $\delta'_{\mathcal{R}}$ are homotopic. □

Remark 4.27. Several other algorithms can be developed in order to obtain topological information about digital Riemann surfaces. For instance, let us sketch an algorithm to compute generators of the homotopy group $\pi_1(\mathcal{R})$:

1. Let $\delta = \epsilon$, $\Delta = \{\}$, $\Pi := \{\}$, let $\mathcal{I}$ be the restriction of $\mathcal{R}$ to $\bullet_A$.
2. Let $\Delta := \Delta \cup ((\delta + \{\pm 1, \pm \mathrm{i}\}) \cap \mathbb{P}_{A_{\mathcal{R}}})$.
3. If $\Delta = \varnothing$ then return Π.
4. Pick an element $\delta = (\delta_1, ..., \delta_l) \in \Delta$ of minimal length l and set $\Delta := \Delta \setminus \{\delta\}$.
5. If $\delta \in \mathbb{P}_{A_{\mathcal{I}}}$, then go to step 3.
6. Let $\tilde{\mathcal{I}}$ be obtained by gluing a new square above $\pi(\delta_A)$ to $\mathcal{Q}_{(\delta_1,...,\delta_{l-1})A_{\mathcal{I}}}$.
7. If there exists a $\delta' \in \mathbb{P}_{A_{\mathcal{I}}}$ with $\delta'_A = \delta_A$, then set $\Pi := \Pi \cup \{\delta' + (-\delta_l, ..., -\delta_0)\}$ and identify $\delta'_{A_{\tilde{\mathcal{I}}}}$ with $\delta_{A_{\tilde{\mathcal{I}}}}$ inside $\tilde{\mathcal{I}}$.
8. Replace $\mathcal{I}$ by $\tilde{\mathcal{I}}$ and go to step 2.

The above algorithm returns a set of digital paths Π each of which elements corresponds to a generator in $\pi_1(\mathcal{R})$.

4.8. Convolution products

Let $\mathcal{R}$ and $\mathcal{S}$ be two computable Riemann surfaces with roots above 0. Organic Riemann surfaces are also useful for the construction of a new Riemann surface $\mathcal{R}*\mathcal{S}$ such that the convolution product of analytic functions f and g on $\mathcal{R}$ resp. $\mathcal{S}$ will be defined on $\mathcal{R}*\mathcal{S}$.

A *digital folding* on $\mathcal{R}$ is a computable mapping $\eta\colon \{0, ..., l_1\} \times \{0, ..., l_2\} \to \mathcal{R}^{\mathrm{dig}}$ such that $\eta(j_1, j_2) \in \mathcal{B}_{\eta(i_1,i_2), r_{\eta(i_1,i_2)}}$ for all $0 \leqslant i_1 \leqslant j_1 \leqslant |\eta|_1 := l_1$ and $0 \leqslant i_2 \leqslant j_2 \leqslant |\eta|_2 := l_2$ with $j_1 - i_1 \leqslant 1$ and $j_2 - i_2 \leqslant 1$. We denote by $\mathbb{F}_{\mathcal{R}}^{\mathrm{dig}}$ the enumerable set of digital foldings on $\mathcal{R}$. We also write $\mathbb{F}_{\mathcal{R},\bullet}^{\mathrm{dig}} \subseteq \mathbb{F}_{\mathcal{R}}^{\mathrm{dig}}$ for the subset of rooted digital foldings η with $\eta(i_1, \cdot) = \bullet_{\mathcal{R}}$. Given $\eta \in \mathbb{F}_{\bullet}^{\mathrm{dig}} = \mathbb{F}_{\mathbb{C}^{\mathrm{dig}},\bullet}^{\mathrm{dig}}$, we define $\eta^! \in \mathbb{F}_{\bullet}^{\mathrm{dig}}$ by

$$\eta^!(i_1, i_2) = \eta(i_1, |\eta|_2) - \eta(i_1, |\eta|_2 - i_2).$$

We define H to be the set of all foldings $\eta \in \mathbb{F}_{\bullet}^{\mathrm{dig}}$ with $\eta(0, \cdot) = 0$, such that η and $\eta^!$ lift to rooted foldings on $\mathcal{R}$ resp. $\mathcal{S}$. We notice that H is enumerable.

Now any $\eta \in \mathrm{H}$ induces a path $\delta = \delta_\eta \in \mathbb{P}^{\mathrm{dig}}$ by $\delta_i = \eta(i_1, l) - \eta(i_1 - 1, l)$, where $|\delta| = l = |\eta|_2$. By construction, we have $\delta \in \mathbb{P}_{\mathcal{R}}^{\mathrm{dig}}$ and $\delta^! = (\delta_l, ..., \delta_1) = \delta_{\eta^!} \in \mathbb{P}_{\mathcal{S}}^{\mathrm{dig}}$. Let $\Delta \subseteq \mathbb{P}^{\mathrm{dig}}$ be the enumerable set of all paths which are induced by foldings in H. Given $\delta = (\delta_1, ..., \delta_l) \in \Delta$, we let

$$r_\delta = \min\{r_{\delta_{\mathcal{R}}}, r_{\bullet_{\mathcal{S}}} - |\delta_l|, r_{(\delta_l)_{\mathcal{S}}} - |\delta_{l-1}|, ..., r_{(\delta_l + \cdots + \delta_2)_{\mathcal{S}}} - |\delta_1|, r_{\delta^!_{\mathcal{S}}}\} \in \bar{\mathbb{R}}^{\mathrm{lcom},>}. \tag{4.3}$$

Given $\varepsilon \in \mathbb{C}^{\mathrm{dig}}$ with $|\varepsilon| < r_\delta$, we claim that $\delta + \varepsilon \in \Delta$. Indeed, let $\eta \in \mathrm{H}$ be such that $\delta = \delta_\eta$ and define $\eta' : \{0, ..., k+1\} \times \{0, ..., l+1\}$ with $k := |\eta|_1$ by

$$\eta'(i_1, i_2) = \begin{cases} \eta(i_1, i_2) & \text{if } i \leqslant k \text{ and } i_2 \leqslant l \\ \eta(k, i_2) & \text{if } i_1 = k+1 \text{ and } i_2 \leqslant l \\ \eta(i_1, l) & \text{if } i_1 \leqslant k \text{ and } i_2 = l+1 \\ \eta(k, l) + \varepsilon & \text{if } i_1 = k+1 \text{ and } i_2 = l+1 \end{cases}$$

By construction, we have $\eta' \in \mathrm{H}$ and $\delta_{\eta'} = \delta + \varepsilon$. In view of remark 4.19, we may now define the convolution product of $\mathcal{R}$ and $\mathcal{S}$ by $\mathcal{R} * \mathcal{S} = \mathcal{O}_{0,\Delta,r}$.

PROPOSITION 4.28. *Let $\eta : [0,1]^2 \to \mathbb{C}$ be a continuous function with $\eta(0, \cdot) = \eta(\cdot, 0) = 0$, such that η and its mirror $\eta^! : (t_1, t_2) \mapsto \eta(t_1, 1) - \eta(t_1, 1 - t_2)$ lift into functions $\eta_{\mathcal{R}}$ and $\eta^!_{\mathcal{S}}$ on $\mathcal{R}$ resp. $\mathcal{S}$ with $\eta_{\mathcal{R}}(0,0) = \bullet_{\mathcal{R}}$ and $\eta^!_{\mathcal{S}}(0,0) = \bullet_{\mathcal{S}}$. Then the path $\gamma : [0,1] \to \mathbb{C}; t \mapsto \eta(t, 1)$ can be lifted to a path $\gamma_{\mathcal{R}*\mathcal{S}}$ on $\mathcal{R}*\mathcal{S}$. In particular, given f and g on $\mathcal{R}$ resp. $\mathcal{S}$, the convolution product $f*g$ can be analytically continued along γ:*

$$(f*g)(\gamma_{\mathcal{R}*\mathcal{S}}(t)) = \int_{\phi_{\eta(t,\cdot),\mathcal{R}}} f(\zeta)\, g(\zeta^!)\, \mathrm{d}\zeta,$$

where $\zeta^! = \eta^!_{\mathcal{S}}(t, 1-u)$, whenever $\zeta = \eta_{\mathcal{R}}(t, u)$.

Proof. We first observe that a digital folding $\eta \in \mathbb{F}^{\mathrm{dig}}_{\mathcal{R}}$ induces a natural continuous mapping $\phi_{\eta,\mathcal{R}} : [0,1]^2 \to \mathcal{R}$ by

$$\begin{aligned} \phi_{\eta,\mathcal{R}}((i_1 + t_1)/l_1, (i_2 + t_2)/l_2) &= \sum_{\epsilon_1, \epsilon_2 \in \{0,1\}} c_{\epsilon_1}(t_1)\, c_{\epsilon_2}(t_2)\, \eta(i_1 + \epsilon_1, i_2 + \epsilon_2) \\ c_\epsilon(t) &= \begin{cases} 1 - t & \text{if } \epsilon = 0 \\ t & \text{otherwise} \end{cases} \end{aligned}$$

Let

$$\varepsilon = \min_{t \in [0,1]^2} \min \left(r_{\phi_{\eta,\mathcal{R}}(t)}, r_{\phi_{\eta^!,\mathcal{S}}(t)} \right).$$

Since η is uniformly continuous, we may approximate it by a digital folding $\tilde{\eta} \in \mathbb{F}^{\mathrm{dig}}$ with

$$\| \eta - \phi_{\tilde{\eta},\mathbb{C}} \| = \max_{t \in [0,1]^2} |\eta(t) - \phi_{\tilde{\eta},\mathbb{C}}(t)| < \varepsilon/2.$$

Moreover, we may take $\tilde{\eta}$ such that $\tilde{\eta}(0, \cdot) = \tilde{\eta}(\cdot, 0) = 0$ and $\tilde{\eta}(|\tilde{\eta}|_1, |\tilde{\eta}|_2) = \eta(1,1)$. By our choice of ε, the foldings $\tilde{\eta}$ and $\tilde{\eta}^!$ lift to $\mathcal{R}$ resp. $\mathcal{S}$, so that $\tilde{\eta} \in \mathrm{H}$. Moreover, the broken line path $\tilde{\delta} = \delta_{\tilde{\eta}}$ satisfies $|\tilde{\delta}_i| < r_{(\tilde{\delta}_1, ..., \tilde{\delta}_{i-1})}$ for all $i \leqslant |\tilde{\delta}|$, again by the choice of ε. Consequently, $\tilde{\delta} \in \mathbb{P}^{\mathrm{dig}}_{\mathcal{R}*\mathcal{S}}$ and its associated continuous path $\tilde{\gamma}_{\mathbb{C}} = \phi_{\tilde{\delta},\mathbb{C}}$ lifts to a path $\tilde{\gamma}_{\mathcal{R}*\mathcal{S}}$ on $\mathcal{R}*\mathcal{S}$ with the same endpoints as $\gamma_{\mathcal{R}*\mathcal{S}}$. Once more by the choice of ε, we have $r_{\tilde{\gamma}_{\mathcal{R}*\mathcal{S}}(t)} \geqslant \varepsilon/2$ for all $t \in [0, 1]$ and $\| \gamma - \tilde{\gamma}_{\mathbb{C}} \| < \varepsilon/2$. Consequently, γ lifts to the path $\gamma_{\mathcal{R}*\mathcal{S}} : t \mapsto \tilde{\gamma}_{\mathcal{R}*\mathcal{S}}(t) + (\tilde{\gamma}_{\mathbb{C}}(t) - \gamma(t))$ on $\mathcal{R}*\mathcal{S}$. □

The convolution product $\mathcal{R}*\mathcal{S}$ comes with natural computable rooted coverings $\varpi_{\mathcal{R}} : \mathcal{R}*\mathcal{S} \to \mathcal{R}$ and $\varpi_{\mathcal{S}} : \mathcal{R}*\mathcal{S} \to \mathcal{S}$, since any $\eta \in \mathrm{H}$ in particular induces a path $\delta \in \mathbb{P}^{\mathrm{dig}}_{\mathcal{R}} \cap \mathbb{P}^{\mathrm{dig}}_{\mathcal{S}}$ with $\delta_i = \eta(i, |\eta|_2) - \eta(i-1, |\eta|_2)$. The following universal property follows from proposition 4.18:

PROPOSITION 4.29. *Let $\varphi : \mathcal{R} \to \mathcal{R}'$ and $\psi : \mathcal{S} \to \mathcal{S}'$ be two computable rooted coverings. Then there exists a unique rooted covering $\varphi*\psi : \mathcal{R}*\mathcal{S} \to \mathcal{R}'*\mathcal{S}'$. This covering is computable and can be computed as a function of φ and ψ. Moreover, $\varpi_{\mathcal{R}'} \circ (\varphi*\psi) = \varpi_{\mathcal{R}}$ and $\varpi_{\mathcal{S}'} \circ (\varphi*\psi) = \varpi_{\mathcal{S}}$.* □

5. COMPUTABLE ANALYTIC FUNCTIONS

In [vdH05a], a computable analytic function f was defined locally as a "computable germ" with a computable method for analytic continuation. In section 5.1, we recall an improved version of this definition. In section 5.3, we define the new concepts of globally and incrementally computable analytic functions. These concepts allow for computations with analytic functions on computable Riemann surfaces as studied in the previous sections. A locally computable analytic function in the sense of section 5.1 will naturally give rise to a globally computable analytic function on an organic Riemann surface. However, common operations on globally analytic functions, as studied in sections 5.4 and 5.5, may give rise to computable Riemann surfaces which are not necessarily simply connected. Our new definition therefore has the advantage that identical branches may be detected effectively in many cases.

5.1. Locally computable analytic functions

Let $f = f_0 + f_1 z + \cdots \in \mathbb{C}[[z]]$ be a convergent power series at the origin. We will write r_f for its radius of convergence. Given $\rho \in \mathbb{R}^>$ with $\rho < r_f$, we also define

$$\| f \|_\rho = \max_{|z| \leqslant \rho} |f(z)|.$$

Finally, given $\delta \in \mathcal{B}_{r_f}$, we will denote by $f_{+\delta}$ the analytic continuation of f along the straightline segment $[0, \delta]$, so that $f_{+\delta}(z) = f(\delta + z)$ for small z.

A *locally computable analytic function* f is an object encoded by a quadruple

$$\check{f} = (\mathsf{series}(f), \underline{r}_f, \llbracket f \rrbracket_\cdot, f_{+\cdot}),$$

where

- $\mathsf{series}(f) \in \mathbb{C}^{\mathrm{com}}[[z]]^{\mathrm{com}}$ is a computable power series.
- $\underline{r}_f \in \bar{\mathbb{R}}^{\mathrm{lcom},>}$ is a lower bound for r_f.
- $\llbracket f \rrbracket_\cdot : \mathbb{R}^{\mathrm{com},>} \rightharpoonup \mathbb{R}^{\mathrm{rcom}}$ is a computable partial function, which yields an upper bound $\llbracket f \rrbracket_\rho \geqslant \| f \|_\rho$ for every $\rho < \underline{r}_f$.
- $f_{+\cdot} : \mathbb{C}^{\mathrm{com}} \rightharpoonup \mathbb{A}^{\mathrm{lcom}}$ is a computable partial function, which yields the analytic continuation $f_{+\delta}$ of f as a function of $\delta \in \mathbb{C}^{\mathrm{com}}$ with $|\delta| < \underline{r}_f$.

We denote by $\mathbb{A}^{\mathrm{lcom}}$ the set of locally computable analytic functions. Given $f \in \mathbb{A}^{\mathrm{lcom}}$, we call $\underline{r}_f$ its *computable radius of convergence*. Usually, $\underline{r}_f$ is smaller than the genuine radius of convergence of $\mathsf{series}(f)$.

Remark 5.1. We notice that the definition of the encoding $\check{f}$ is recursive, because of the method $f_{+\cdot}$ for analytic continuation. Such recursive quadruples can in their turn be encoded by terms in a suitable λ-calculus, and thereby make the definition fit into the setting introduced in section 2.1.

Example 5.2. One important example of a locally computable analytic function is the identity function z centered at a given $c \in \mathbb{C}^{\mathrm{com}}$. We may implement it as a function $\mathrm{Id} : c \mapsto \mathrm{Id}_c$ with

$$\begin{aligned} \mathsf{series}(\mathrm{Id}_c) &= z + c \\ \underline{r}_{\mathrm{Id}_c} &= +\infty \\ \llbracket \mathrm{Id}_c \rrbracket_\rho &= |c| + |\rho| \\ (\mathrm{Id}_c)_{+\delta} &= \mathrm{Id}_{c+\delta} \end{aligned}$$

The constructor $\mathbb{C}^{\mathrm{com}} \to \mathbb{A}^{\mathrm{lcom}}: c \mapsto c$ can be implemented in a similar way.

Example 5.3. Basic operations on $\mathbb{A}^{\mathrm{lcom}}$ can easily be implemented in a recursive manner. For instance, the addition of $f, g \in \mathbb{A}^{\mathrm{lcom}}$ may be computed by taking

$$\begin{aligned} \mathsf{series}(f+g) &= \mathsf{series}(f) + \mathsf{series}(g) \\ \underline{r}_{f+g} &= \min(\underline{r}_f, \underline{r}_g) \\ [\![f+g]\!]_\rho &= [\![f]\!]_\rho + [\![g]\!]_\rho \\ (f+g)_{+\delta} &= f_{+\delta} + g_{+\delta} \end{aligned}$$

In sections 3 and 4 of [vdH05a], algorithms were given for several other basic operations and for the resolution of differential equations. Modulo minor modifications, these algorithms remain valid. In particular, we have implementations for the following operations:

$$\begin{aligned} \iota: &\quad \mathbb{C}^{\mathrm{com}} &\to& \quad \mathbb{A}^{\mathrm{lcom}} \\ z: &\quad & & \quad \mathbb{A}^{\mathrm{lcom}} \\ +, -, \times: &\quad \mathbb{A}^{\mathrm{lcom}} \times \mathbb{A}^{\mathrm{lcom}} &\to& \quad \mathbb{A}^{\mathrm{lcom}} \\ /: &\quad \mathbb{A}^{\mathrm{lcom}} \times \mathbb{A}^{\mathrm{lcom}} &\rightharpoonup& \quad \mathbb{A}^{\mathrm{lcom}} \\ \mathrm{d}/\mathrm{d}z: &\quad \mathbb{A}^{\mathrm{lcom}} &\to& \quad \mathbb{A}^{\mathrm{lcom}} \\ \textstyle\int: &\quad \mathbb{A}^{\mathrm{lcom}} \times \mathbb{C}^{\mathrm{com}} &\to& \quad \mathbb{A}^{\mathrm{lcom}} \\ \exp: &\quad \mathbb{A}^{\mathrm{lcom}} &\to& \quad \mathbb{A}^{\mathrm{lcom}} \\ \log: &\quad \mathbb{A}^{\mathrm{lcom}} \times \mathbb{C}^{\mathrm{com}} &\rightharpoonup& \quad \mathbb{A}^{\mathrm{lcom}} \end{aligned} \tag{5.1}$$

In the cases of $\int$ and $\log$, the second argument specifies the value of the function at 0.

It is instructive to rethink the definition of $\mathbb{A}^{\mathrm{lcom}}$ in terms of signatures. First of all, we have a class of computable power series $\mathbb{C}^{\mathrm{com}}[[z]]^{\mathrm{com}}$ with a method for the extraction of coefficients

$$\cdot_: \ \mathbb{C}^{\mathrm{com}}[[z]]^{\mathrm{com}} \times \mathbb{N} \ \to \ \mathbb{C}^{\mathrm{com}}$$

Then the class $\mathbb{A}^{\mathrm{lcom}}$ of locally computable analytic functions is determined by the methods

$$\begin{aligned} \mathsf{series}: &\quad \mathbb{A}^{\mathrm{lcom}} &\to& \quad \mathbb{C}^{\mathrm{com}}[[z]]^{\mathrm{com}} \\ \underline{r}.: &\quad \mathbb{A}^{\mathrm{lcom}} &\to& \quad \bar{\mathbb{R}}^{\mathrm{lcom},>} \\ [\![\cdot]\!]_.: &\quad \mathbb{A}^{\mathrm{lcom}} \times \mathbb{R}^{\mathrm{com},>} &\rightharpoonup& \quad \mathbb{R}^{\mathrm{rcom},\geqslant} \\ \cdot_{+.}: &\quad \mathbb{A}^{\mathrm{lcom}} \times \mathbb{C}^{\mathrm{com}} &\rightharpoonup& \quad \mathbb{A}^{\mathrm{lcom}} \end{aligned} \tag{5.2}$$

For the last two methods, we understand that $[\![f]\!]_\rho$ and $f_{+\delta}$ are defined if and only if $\rho < \underline{r}_f$ resp. $|\delta| < \underline{r}_f$.

The recursive definition of $\mathbb{A}^{\mathrm{lcom}}$ raises the question when two elements $f, g \in \mathbb{A}^{\mathrm{lcom}}$ should be considered identical. In what follows, we will use the criterion that $f = g$ if and only if the signature (5.2) does not allow for the distinction of f and g. In other words, whenever $\delta_1, \ldots, \delta_l \in \mathbb{C}^{\mathrm{com}}$ and $\rho \in \mathbb{R}^{\mathrm{com},>}$ are such that $\tilde{f} = f_{+\delta_1+\cdots+\delta_l}$ and $\tilde{g} = g_{+\delta_1+\cdots+\delta_l}$ are both defined and $\rho < \underline{r}_{\tilde{f}}$, we require that $\mathsf{series}(\tilde{f}) = \mathsf{series}(\tilde{g})$, $\underline{r}_{\tilde{f}} = \underline{r}_{\tilde{g}}$ and $[\![\tilde{f}]\!]_\rho = [\![\tilde{g}]\!]_\rho$. We warn that we may have $\mathsf{series}(f) = \mathsf{series}(g)$ for two different elements $f, g \in \mathbb{A}^{\mathrm{lcom}}$.

Remark 5.4. There are a few changes between the present definition and the definition of computable analytic functions in [vdH05a]. First of all, we have loosened the requirements for bound computations by allowing the results of $\underline{r}_f$ and $[\![f]\!]_\rho$ to be only left resp. right computable. In [vdH05a], we also required a few additional global consistency conditions in our definition of computable analytic functions. The homotopy condition will no longer be needed because of theorem 5.7 below, even though its satisfaction may speed up certain algorithms. The continuity condition also becomes superfluous because of theorem 2.3.

5.2. Improved bounds and default analytic continuation

Given $f \in \mathbb{A}^{\mathrm{lcom}}$, we have already noticed that the computable radius of convergence $\underline{r}_f$ of f does not necessarily coincide with its theoretical radius of convergence r_f. This raises a problem when we want to analytically continue f, because we are not always able to effectively continue f at all points where f is theoretically defined. By contrast, bad upper bounds for $[\![f]\!]_\rho$ on compact disks only raise an efficiency problem. Indeed, we will show now how to improve bad bounds into exact bounds.

Let us first introduce some new concepts. The *path domain* $\mathbb{P}_f^{\mathrm{com}}$ of f is the set of $\delta \in \mathbb{P}^{\mathrm{com}}$ such that $|\delta_i| < \underline{r}_{f_{+\delta_1+\cdots+\delta_{i-1}}}$ for every $i \in \{1, ..., |\delta|\}$. Given $\delta \in \mathbb{P}_f^{\mathrm{com}}$, we denote $f_{+\delta} = f_{+\delta_1+\cdots+\delta_{|\delta|}}$ and $f(\delta) = f_{+\delta}(0)$. The *digital path domain* of f, which is defined by $\mathbb{P}_f^{\mathrm{dig}} = \mathbb{P}_f^{\mathrm{com}} \cap \mathbb{P}^{\mathrm{dig}}$, is enumerable. Given $f, g \in \mathbb{A}^{\mathrm{lcom}}$, we say that g *improves* f, and we write $f \sqsubseteq g$, if $\mathsf{series}(f) = \mathsf{series}(g)$, $\mathbb{P}_f^{\mathrm{com}} \subseteq \mathbb{P}_g^{\mathrm{com}}$ and $[\![g_{+\delta}]\!]_\rho \leqslant [\![f_{+\delta}]\!]_\rho$ for all $\delta \in \mathbb{P}_f^{\mathrm{com}}$ and $\rho < \underline{r}_{f_{+\delta}}$.

Assume now that we are given $\rho, \varepsilon \in \mathbb{R}^{\mathrm{com},>}$ with $\rho < \underline{r}_f$ and let us show how to compute an ε-approximation for $M = \| f \|_\rho$. Approximating $\underline{r}_f$ sufficiently far, we first compute an $R \in \mathbb{R}^{\mathrm{com}}$ with $\rho < R < \underline{r}_f$. Now let $B = [\![f]\!]_R$ and choose

$$n = \left\lceil \log \frac{(R-\rho)\,\varepsilon}{2\,R\,B} \Big/ \log \frac{\rho}{R} \right\rceil \tag{5.3}$$

sufficiently large such that

$$|f_n \zeta^n + f_{n+1} \zeta^{n+1} + \cdots| \leqslant B \left(\frac{\rho}{R}\right)^n \frac{R}{R-\rho} \leqslant \frac{\varepsilon}{2}. \tag{5.4}$$

Using an algorithm which will be specified in section 6.2, we next compute an $(\varepsilon/2)$-approximation $\tilde{M}$ for $\|P\|_\rho$, where $P = f_0 + \cdots + f_{n-1} \zeta^{n-1}$. Then $\tilde{M}$ is the desired ε-approximation of M. We have proved:

PROPOSITION 5.5. *Given $f \in \mathbb{A}^{\mathrm{lcom}}$, we may compute an improvement $f^* \in \mathbb{A}^{\mathrm{lcom}}$ of f, such that $[\![f^*_{+\delta}]\!]_\rho = \| f_{+\delta} \|_\rho$ for all $\delta \in \mathbb{P}_f^{\mathrm{com}}$ and $|\rho| < \underline{r}_{f_{+\delta}}$.* □

Another situation which frequently occurs is that the radius of convergence can be improved via the process of analytic continuation and that we want to compute bounds on larger disks, corresponding to the new radii of convergence. This problem may be reformulated by introducing the class $\mathbb{A}^{\mathrm{wlcom}}$ of *weak* locally computable analytic functions. The signatures of $\mathbb{A}^{\mathrm{wlcom}}$ and $\mathbb{A}^{\mathrm{lcom}}$ are the same except that $\mathbb{A}^{\mathrm{wlcom}}$ comes with a second radius function $s_.: \mathbb{A}^{\mathrm{wlcom}} \to \bar{\mathbb{R}}^{\mathrm{com},>}$ with $s_f \leqslant \underline{r}_f$; given $f \in \mathbb{A}^{\mathrm{wlcom}}$, we only require computable bounds $[\![f]\!]_\rho$ for $\rho < s_f$. We have a natural inclusion $\mathbb{A}^{\mathrm{lcom}} \to \mathbb{A}^{\mathrm{wlcom}}$ and the notions of path domain and improvement naturally extend to $\mathbb{A}^{\mathrm{wlcom}}$.

Assume now that $f \in \mathbb{A}^{\mathrm{wlcom}}$ and that we want to compute a bound for $[\![f]\!]$ on $\bar{\mathcal{B}}_\rho$ for a given $\rho < \underline{r}_f$. We have an algorithm for computing a $t_z \in \mathbb{R}^{\mathrm{com},>}$ with $t_z < s_{f_{+z}}$ from $z \in \bar{\mathcal{B}}_\rho^{\mathrm{dig}}$. Consider any computable sequence $z_0, z_1, ... \in \bar{\mathcal{B}}_\rho$ with

$$z_{n+1} \notin \mathcal{B}_{z_0,t_{z_0}} \cup \cdots \cup \mathcal{B}_{z_n,t_{z_n}}.$$

Since $\bar{\mathcal{B}}_\rho$ is compact and the function $z \in \bar{\mathcal{B}}_\rho^{\mathrm{dig}} \mapsto t_z$ is continuous (by theorem 2.3), it follows that there exists an $\varepsilon > 0$ with $t_z \geqslant \varepsilon$ for all $z \in \bar{\mathcal{B}}_\rho^{\mathrm{dig}}$. In particular, the balls $\mathcal{B}_{z_i,t_{z_i}}$ form an open covering of $\bar{\mathcal{B}}_\rho$, whence the sequence $z_0, z_1, ...$ is necessarily finite. Let z_l be its last term. Then

$$[\![f]\!]^*_\rho = \max\left([\![f_{+z_0}]\!]_{t_{z_0}}, ..., [\![f_{+z_l}]\!]_{t_{z_l}}\right)$$

is a computable upper bound for $\| f \|_\rho$. We have proved:

PROPOSITION 5.6. *Given a weak locally computable analytic function $f \in \mathbb{A}^{\mathrm{wlcom}}$, we may compute an improvement $f^* \in \mathbb{A}^{\mathrm{lcom}}$ of f.* □

The bound (5.4) may also be used in order to provide a default analytic continuation method of a computable power series $f \in \mathbb{C}[[z]]^{\mathrm{com}}$ inside a given computable radius of convergence $\underline{r}_f \in \mathbb{R}^{\mathrm{lcom},>}$, assuming that we have an algorithm for the computation of upper bounds $\llbracket f \rrbracket_\rho$, $(\rho < \underline{r}_f)$. Indeed, let $\delta \in \mathbb{C}^{\mathrm{com}}$, $k \in \mathbb{N}$ and $\varepsilon \in \mathbb{R}^{\mathrm{com},>}$ be such that $|\delta| < \underline{r}_f$ and assume that we want to compute an ε-approximation of $(f_{+\delta})_k = f^{(k)}(\delta)/k!$. Now choose ρ, $\rho' \in \mathbb{R}^{\mathrm{com}}$ with $|\delta| < \rho < \rho' < \underline{r}_f$ and let $M' = \llbracket f \rrbracket_{\rho'}$. Then the majoration [vdH05a, vdH03]

$$f \preccurlyeq \frac{M'}{1 - z/\rho'}$$

yields the majoration

$$\frac{f^{(k)}}{k!} \preccurlyeq M' \left(\frac{\rho'}{1 - z/\rho'} \right)^{k+1},$$

so that

$$\| f^{(k)} \|_\rho \leqslant M := M' \left(\frac{(\rho')^2}{\rho' - \rho} \right)^{k+1}.$$

Taking n in a similar way as in (5.3), we thus have

$$\left| \binom{k+n}{k} f_n \delta^n + \binom{k+n+1}{k} f_{n+1} \delta^{n+1} + \cdots \right| \leqslant \frac{\varepsilon}{2}.$$

Let u be an $(\varepsilon/2)$-approximation of $\binom{k}{k} f_0 + \cdots + \binom{k+n-1}{k} f_{n-1} \delta^{n-1}$. Then u is also an ε-approximation of $(f_{+\delta})_k$.

5.3. Globally and incrementally computable analytic functions

Let us now consider an analytic function f on a computable rooted Riemann surface $\mathcal{R}$. We say that f is a *globally computable analytic function* on $\mathcal{R}$, if there exists a computable function $L_f : \mathcal{R}^{\mathrm{com}} \to \mathbb{A}^{\mathrm{lcom}}$, which maps $\zeta \in \mathcal{R}^{\mathrm{com}}$ to a locally computable analytic function $L_f(\zeta) : z \mapsto f(\zeta + z)$, such that

$$\underline{r}_{L_f(\zeta)} = r_\zeta \tag{5.5}$$

$$\llbracket L_f(\zeta) \rrbracket_\rho = \| f_{+\zeta} \|_\rho \tag{5.6}$$

$$L_f(\zeta)_{+\delta} = L_f(\zeta + \delta) \tag{5.7}$$

for all $\zeta \in \mathcal{R}^{\mathrm{com}}$, $\rho \in \mathbb{C}^{\mathrm{com},>,<r_\zeta}$ and $\delta \in \mathcal{B}^{\mathrm{com}}_{r_\zeta}$. We denote by $\mathbb{A}^{\mathrm{com}}_{\mathcal{R}}$ the set of such functions. We also denote by $\mathbb{A}^{\mathrm{com}}$ the set of all computable analytic functions f on *some* connected computable (and computable as a function of f) rooted Riemann surface $\mathcal{R}_f$. In other words, the signature of $\mathbb{A}^{\mathrm{com}}$ is given by

$$\begin{aligned} \mathcal{R}_\cdot : \ & \mathbb{A}^{\mathrm{com}} \to \mathbb{S}^{\mathrm{com}}_\bullet \\ \Lambda : \ & \mathbb{A}^{\mathrm{com}} \to \mathbb{A}^{\mathrm{lcom}} \end{aligned}$$

Here the projection Λ is required to satisfy $\Lambda(f)_{+\delta} = L_f(\delta_{\mathcal{R}})$. The signature for $\mathbb{A}^{\mathrm{com}}$ can also be given in a more intrinsic way:

$$\begin{aligned} \mathsf{series}: \ & \mathbb{A}^{\mathrm{com}} \to \mathbb{C}^{\mathrm{com}}[[z]]^{\mathrm{com}} \\ \underline{r}_\cdot : \ & \mathbb{A}^{\mathrm{com}} \to \bar{\mathbb{R}}^{\mathrm{lcom},>} \\ \| \cdot \|_\cdot : \ & \mathbb{A}^{\mathrm{com}} \times \mathbb{R}^{\mathrm{com},>} \rightharpoonup \mathbb{R}^{\mathrm{com},\geqslant} \\ \cdot_+ : \ & \mathbb{A}^{\mathrm{com}} \times \mathbb{C}^{\mathrm{com}} \rightharpoonup \mathbb{A}^{\mathrm{com}} \\ \mathcal{R}_\cdot : \ & \mathbb{A}^{\mathrm{com}} \to \mathbb{S}^{\mathrm{com}}_\bullet \\ \bullet_\cdot : \ & \mathbb{A}^{\mathrm{com}} \to \mathcal{R}^{\mathrm{com}}_\cdot \end{aligned}$$

Notice that the method $[\![\cdot]\!].$ has been replaced by an exact method $\|\cdot\|.$ with codomain $\mathbb{R}^{\mathrm{com},\geqslant}$, in view of proposition 5.5. For the intrinsic representation, the compatibility conditions become $\underline{r}_f = r_{\bullet_f}$ and $\mathcal{R}_{f_{+\delta}} = \mathcal{R}_{f,+\delta}$, where $\mathcal{R}_{f,+\delta}$ denotes the Riemann surface $\mathcal{R}_f$ with its root moved by δ.

Using analytic continuation, let us now show how to improve a locally computable analytic function $f \in \mathbb{A}^{\mathrm{lcom}}$ into a globally computable analytic function $f^\sharp \in \mathbb{A}^{\mathrm{com}}$.

THEOREM 5.7. *There exists an algorithm which takes $f \in \mathbb{A}^{\mathrm{lcom}}$ on input and produces an improvement $f^\sharp \in \mathbb{A}^{\mathrm{com}}$ of f on output. Given any other improvement $g \in \mathbb{A}^{\mathrm{com}}$ of f, we have $f^\sharp \sqsubseteq g$.*

Proof. For the underlying Riemann surface of $f^\sharp$, we take $\mathcal{R}_{f^\sharp} = \mathcal{O}_{0,\mathbb{P}_f,r}$, with $r_\delta = \underline{r}_{f_{+\delta}}$ for all $\delta \in \mathbb{P}_f^{\mathrm{dig}}$. By the construction of $\mathcal{O}_{0,\mathbb{P}_f,r}$, we may compute a sequence $n \mapsto \mathcal{B}_{c_n,r_n}$ of open balls $\mathcal{B}_{c_n,r_n} \subseteq \mathcal{R}_{f^\sharp}$ with $c_n = (\delta_n)_{\mathcal{R}_{f^\sharp}}$, $\delta_n \in \mathbb{P}_f^{\mathrm{dig}}$ and $\mathbb{R}^{\mathrm{dig},>} \ni r_n \leqslant \underline{r}_{f_{+\delta_n}}$, such that

$$\mathcal{R}_{f^\sharp} = \bigcup_{n \in \mathbb{N}} \mathcal{B}_{c_n,r_n}.$$

Given $\zeta \in \mathcal{R}_{f^\sharp}^{\mathrm{com}}$, we may therefore compute an n with $\zeta \in \mathcal{B}_{\zeta_n,r_n}$ (which may depend on the encoding $\check{\zeta}$) and take

$$\mathsf{series}(L_{f^\sharp}(\zeta)) = \mathsf{series}(f_{+\delta_n+(\pi(\zeta)-\|\delta_n\|)}).$$

Given $\zeta \in \mathcal{R}_{f^\sharp}^{\mathrm{com}}$, we may also compute

$$s_{L_{f^\sharp}(\zeta)} = \max_{n \in \mathbb{N}} \{r_{f_{+\delta_n}} - |\zeta - c_n| : \zeta \in \mathcal{B}_{c_n,r_n}\} \in \mathbb{R}^{\mathrm{lcom},>}.$$

Given $\rho \in \mathbb{R}^{\mathrm{com},>}$ with $\rho < s_{L_{f^\sharp}(\zeta)}$, we may finally compute

$$[\![L_{f^\sharp}(\zeta)]\!]_\rho = \min_{n \in \mathbb{N}} \{[\![f_{+\delta_n}]\!]_{\rho+|\zeta-c_n|} : \zeta \in \mathcal{B}_{c_n,r_n}\} \in \mathbb{R}^{\mathrm{rcom},\geqslant}.$$

By propositions 5.6 and 5.5, we may therefore compute $\|L_{f^\sharp}(\zeta)\|_\rho$ for every $\rho \in \mathbb{R}^{\mathrm{com},>}$ with $|\rho| < r_\zeta$. Since $\mathbb{P}_{f^\sharp}^{\mathrm{dig}} \supseteq \mathbb{P}_f^{\mathrm{dig}}$, proposition 4.14 and its adaptation to $\mathbb{P}_f^{\mathrm{dig}}$ imply $\mathbb{P}_{f^\sharp}^{\mathrm{com}} \supseteq \mathbb{P}_f^{\mathrm{com}}$, whence $f^\sharp \sqsupseteq f$. The universal property of $\mathcal{O}_{0,\mathbb{P}_f,r}$ (proposition 4.17) implies that $f^\sharp \sqsubseteq g$ for any other improvement $g \in \mathbb{A}^{\mathrm{com}}$ of f. □

In practice, it is not very convenient to compute with global computable analytic functions f, because we have no control over the regions where we wish to investigate f first. An alternative formalization relies on the incremental extension of the Riemann surface on which f is known. Consider the class $\mathbb{A}^{\mathrm{icom}}$ with the following signature:

$$\begin{array}{rrcl} \mathcal{R}_\cdot: & \mathbb{A}^{\mathrm{icom}} & \to & \mathbb{S}_\bullet^{\mathrm{com}} \\ \Lambda: & \mathbb{A}^{\mathrm{icom}} & \to & \mathbb{A}^{\mathrm{lcom}} \\ X: & \mathbb{A}^{\mathrm{icom}} \times \mathbb{P}_\cdot^{\mathrm{com}} & \to & \mathbb{A}^{\mathrm{icom}} \end{array}$$

Given $f \in \mathbb{A}^{\mathrm{icom}}$ and $\delta \in \mathbb{P}_f^{\mathrm{com}} \cap \mathbb{P}_{\mathcal{R}_f}^{\mathrm{com}}$, where $\mathbb{P}_f^{\mathrm{com}} = \mathbb{P}_{\Lambda(f)}^{\mathrm{com}}$, the method X returns an extension $\tilde{f} = X_\delta(f)$ of f on a Riemann surface $\mathcal{R}_{\tilde{f}}$ with $\mathbb{P}_{\mathcal{R}_{\tilde{f}}}^{\mathrm{com}} \supseteq \delta + \mathcal{B}_{\underline{r}_{\Lambda(f)_{+\delta}}}^{\mathrm{com}}$ (in particular, there exists a computable rooted covering $\varphi: \mathcal{R}_f \to \mathcal{R}_{\tilde{f}}$). For simplicity, it will be convenient to assume that $\Lambda(\tilde{f}) = \Lambda(f)$. For consistency, we also assume that successive calls of X for paths $\delta_1, \ldots, \delta_l$ and $\delta'_1, \ldots, \delta'_{l'}$ with $\{\delta_1, \ldots, \delta_l\} \subseteq \{\delta'_1, \ldots, \delta'_{l'}\}$ yield extended surfaces $\mathcal{R}_1 = \mathcal{R}_{X_{\delta_l} \circ \cdots \circ X_{\delta_1}(f)}$ and $\mathcal{R}_2 = \mathcal{R}_{X_{\delta'_{l'}} \circ \cdots \circ X_{\delta'_1}(f)}$ for which there exists a rooted covering $\mathcal{R}_1 \to \mathcal{R}_2$. This ensures the following:

PROPOSITION 5.8. *Consider an enumeration $\{\delta_0, \delta_1, \ldots\}$ of $\mathbb{P}^{\mathrm{com}}_f$. Then the limit $\mathcal{R}^\sharp_f$ of the computable rooted covering sequence $\mathcal{R}_f \to \mathcal{R}_{X_{\delta_0}(f)} \to \mathcal{R}_{X_{\delta_1} \circ X_{\delta_0}(f)} \to \cdots$ does not depend on the particular ordering of the enumeration (up to isomorphism).*

COROLLARY 5.9. *There exists an algorithm which takes $f \in \mathbb{A}^{\mathrm{icom}}$ on input and produces an improvement $f^\sharp \in \mathbb{A}^{\mathrm{com}}$ of f on output. Given any other improvement $g \in \mathbb{A}^{\mathrm{com}}$ of f, we have $f^\sharp \sqsubseteq g$.*

Any locally computable analytic function $f \in \mathbb{A}^{\mathrm{lcom}}$ naturally determines an incrementally computable analytic function $f^{\mathrm{inc}} \in \mathbb{A}^{\mathrm{icom}}$: starting with $\mathcal{R}_f = \mathcal{B}_{r_{\Lambda(f)}}$, each successive call of $X_\delta(f)$ joins a ball with radius $r_{\Lambda(f)_{+\delta}}$ to $\mathcal{R}_f$ at the end of $\delta_{\mathcal{R}_f}$, just like in the construction of organic Riemann surfaces. However, as we will see in the next section, the method X may also be used to identify identical branches in the Riemann surface of a function.

5.4. Operations on computable analytic functions

In this section, we improve the implementations of the operations (5.1) so as to identify branches of the underlying computable Riemann surface of an analytic function f, whenever we know that f takes the same values on both branches. We will also consider several other operations on computable analytic functions.

Constructors. The inclusion $\iota\colon \mathbb{C}^{\mathrm{com}} \to \mathbb{A}^{\mathrm{icom}}$ and identity $z\colon \mathbb{A}^{\mathrm{icom}}$ are easy to implement, since it suffices to take $\mathbb{C}$ for the Riemann surface and return $X_\delta(f) = f$ for all $\delta \in \mathbb{P}^{\mathrm{com}}$.

Entire functions. Let us now consider the case of addition $f + g$ for $f, g \in \mathbb{A}^{\mathrm{icom}}$. We take

$$\mathcal{R}_{f+g} = \mathcal{R}_f \between^\bullet \mathcal{R}_g,$$

where $\between^\bullet$ stands for the rooted covering product, i.e. $\mathcal{R}_f \between^\bullet \mathcal{R}_g$ is the connected component of the root of $\mathcal{R}_f \between \mathcal{R}_g$. This root is computed by applying the universal property of computable covering products to the immersions of a small ball $\mathcal{B}_{\pi(\bullet_{\mathcal{R}}),\varepsilon}$ into neighbourhoods of the roots of $\mathcal{R}_f$ and $\mathcal{R}_g$. As to the method X, we may simply take

$$X_\delta(f + g) = X_\delta(f) + X_\delta(g).$$

The consistency condition for successive applications of X is naturally met, because of the universal property of covering products. The cases of subtraction, multiplication, exponentiation and precomposition with any other computable entire functions in several variables can be dealt with in a similar way.

Multiplicative inverses. Assume now that we want to compute f^{-1} for $f \in \mathbb{A}^{\mathrm{icom}}$ with $f(\bullet_f) \neq 0$. Clearly, we may take

$$\begin{aligned} \mathcal{R}_{f^{-1}} &= \mathcal{R}_f^{\neq} := \{\zeta \in \mathcal{R}_f\colon f(\zeta) \neq 0\} \\ X_\delta(f^{-1}) &= X_\delta(f)^{-1} \end{aligned}$$

It remains to be shown that $\mathcal{R}_f^{\neq}$ is a computable rooted Riemann surface. It suffices to show this in the case when $\mathcal{R}_f$ is a digital Riemann surface. Indeed,

$$\mathcal{R}_f = \lim \mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots \implies \mathcal{R}_f^{\neq} = \lim \mathcal{R}_0^{\neq} \xrightarrow{\varphi_0} \mathcal{R}_1^{\neq} \xrightarrow{\varphi_1} \cdots.$$

Now for every point $\zeta \in \mathcal{R}_f$ above $\mathbb{C}^{\text{dig}}$, we will show in section 6.1 how to compute a maximal disk $\mathcal{B}_{\zeta,r_\zeta}$ on which f does not vanish. For the n-th approximation $\mathcal{R}_n^{\neq}$ of $\mathcal{R}_f^{\neq}$, it suffices to take the union of all $\mathcal{B}_{\zeta,r_\zeta}$ with $\zeta \in \mathbb{Z}[\mathrm{i}]/2^n$ (starting with an n for which $\mathcal{R}_n^{\neq}$ contains the root of $\mathcal{R}_f$).

Differentiation. Given $f \in \mathbb{A}^{\text{icom}}$, we may take

$$\begin{aligned} \mathcal{R}_{f'} &= \mathcal{R}_f \\ X_\delta(f') &= X_\delta(f)'. \end{aligned}$$

Integration. Given $f \in \mathbb{A}^{\text{icom}}$ and $c \in \mathbb{C}^{\text{com}}$, let

$$g(\xi) = I(f,c) = \int_\bullet^\zeta f(\xi)\,\mathrm{d}\xi.$$

Let $\mathcal{R}_f$ be the limit of a covering sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$ of digital Riemann surfaces. Given $n \in \mathbb{N}$, we have sketched in remark 4.27 how to compute generators $\gamma_1, ..., \gamma_g$ for the homotopy group $\pi_1(\mathcal{R}_n)$ of $\mathcal{R}_n$. The relations $\gamma_i\gamma_j = \gamma_j\gamma_i$ induce a computable equivalence relation $\sim_n$ on $\mathcal{R}_n^\sharp$. Setting $\mathcal{R}_n^I = \mathcal{R}_n^\sharp/\!\sim_n$, the covering φ_n gives rise to a natural covering $\varphi_n^I\colon \mathcal{R}_n^I \to \mathcal{R}_{n+1}^I$. We take

$$\begin{aligned} \mathcal{R}_{I(f,c)} &= \lim \mathcal{R}_0^I \xrightarrow{\varphi_0^I} \mathcal{R}_1^I \xrightarrow{\varphi_1^I} \cdots \\ X_\delta(I(f,c)) &= I(X_\delta(f), c). \end{aligned}$$

Logarithm. Given $f \in \mathbb{A}^{\text{icom}}$ and $c \in \mathbb{C}^{\text{com}}$ with $f(\bullet) = \mathrm{e}^c$, we may take

$$\log(f,c) = I(f'/f, c).$$

However, in this particular case, the integrals of f'/f over the above generators γ_i are always multiples of $2\pi\mathrm{i}$, so they can be computed exactly. More precisely, let $\mathcal{R}_f^{\neq} = \mathcal{R}_{f'/f}$ be the limit of $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$. We now replace $\mathcal{R}_n^I$ by $\mathcal{R}_n^{\log} = \mathcal{R}_n^\sharp/\!\sim$, where $\sim$ is the equivalence relation defined by

$$\zeta \sim \xi \Leftrightarrow \zeta^\flat = \xi^\flat \wedge |\log(f(\zeta)) - \log(f(\xi))| < \pi.$$

Given ζ, $\xi \in \mathcal{R}_n^\sharp$, we may check whether $|\log(f(\zeta)) - \log(f(\xi))| < \pi$, by computing 1-approximations ℓ_1 and ℓ_2 of $\log(f(\zeta))$ resp. $\log(f(\xi))$ and testing whether $|\ell_1 - \ell_2| \leqslant 2$. The covering φ_n induces a natural covering $\varphi_n^{\log}\colon \mathcal{R}_n^{\log} \to \mathcal{R}_{n+1}^{\log}$ and we take

$$\begin{aligned} \mathcal{R}_{\log(f,c)} &= \lim \mathcal{R}_0^{\log} \xrightarrow{\varphi_0^{\log}} \mathcal{R}_1^{\log} \xrightarrow{\varphi_1^{\log}} \cdots \\ X_\delta(\log(f,c)) &= \log(X_\delta(f), c). \end{aligned}$$

Solving algebraic equations. Let P_{d-1}, ..., $P_0 \in \mathbb{A}^{\text{icom}}$ be such that the polynomial $P = F^d + P_{d-1}F^{d-1} + \cdots + P_0$ is square-free. Let

$$\begin{aligned} \mathcal{R} &= \mathcal{R}_{P_{d-1}} \leftthreetimes \cdots \leftthreetimes \mathcal{R}_{P_0} \\ &= \lim \mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots. \end{aligned}$$

Let $\mathcal{S}_n$ be the digital Riemann surface with

$$\begin{aligned} \lambda_{\mathcal{S}_n} &= \lambda_{\mathcal{R}_n} \\ A_{\mathcal{S}_n} &= \{1, ..., d\} \times A_{\mathcal{R}_n} \\ \pi(i,a) &= \pi(a) \end{aligned}$$

and with an adjacency relation $\boxplus$ defined as follows. Solving the equation $P(f) = 0$ at the center c_a of $a \in A_{\mathcal{R}_n}$ yields d solutions $f_{a,1}, ..., f_{a,d}$ which we attach arbitrarily to the $(i, a) \in A_{\mathcal{S}_n}$ with $i \in \{1, ..., d\}$. We set $(i, a) \boxplus (j, b)$ if $a \boxplus b$ and if the analytic continuation of $f_{a,i}$ from $\pi(c_a)$ to $\pi(c_b)$ coincides with $f_{b,j}$. This can be tested effectively, since there are no multiple roots, whence all branches are bounded away from each other when computing with a sufficient precision. By a similar argument, the root $\bullet_{\mathcal{R}_n}$ of $\mathcal{R}_n$ may be lifted to $\mathcal{S}_n$, if $f(\bullet_{\mathcal{R}})$ has a prescribed value $c \in \mathbb{C}^{\mathrm{com}}$, and the rooted covering φ_n may be lifted to a rooted covering $\psi_n : \mathcal{S}_n \to \mathcal{S}_{n+1}$. We now take

$$\mathcal{R}_f = \lim \mathcal{S}_0 \xrightarrow{\psi_0} \mathcal{S}_1 \xrightarrow{\psi_1} \cdots.$$

Denoting $f = \mathsf{solve}(P, c)$, we also take

$$X_\delta(f) = \mathsf{solve}(F^d + X_\delta(F_{d-1})\, F^{d-1} + \cdots + X_\delta(F_0), f(\delta)).$$

Integral equations. Consider an equation of the form

$$f(z) = I + \int_0^z \Phi(f(t))\, \mathrm{d}t, \tag{5.8}$$

where $f = (f_1, ..., f_d)$ is a vector of indeterminates, Φ a polynomial in $f_1, ..., f_d$ and $I \in (\mathbb{C}^{\mathrm{com}})^d$. Any algebraic differential equation can be rewritten in this form. In section 6.3 below, we will discuss techniques for computing the power series solution to (5.8) at the origin, as well as bounds $\underline{r}_{f_i}$ and $\llbracket f_i \rrbracket_\rho$. Given $\delta \in \mathbb{C}^{\mathrm{com}}$ with $|\delta| < \underline{r}_{f_i}$ for all i, we have

$$\begin{aligned} f_{+\delta}(z) &= I + \int_0^\delta \Phi(f(t))\, \mathrm{d}t + \int_0^z \Phi(f_{+\delta}(t))\, \mathrm{d}t \\ &= f(\delta) + \int_0^z \Phi(f_{+\delta}(t))\, \mathrm{d}t. \end{aligned} \tag{5.9}$$

By what has been said at the end of section 5.2, we may compute $f(\delta) \in (\mathbb{C}^{\mathrm{com}})^d$. Consequently, the equations (5.8) and (5.9) have the same form, and the analytic continuation process may be used recursively, so as to yield a solution $f \in (\mathbb{A}^{\mathrm{lcom}})^n$. Since we do not have any *a priori* knowledge about identical branches in the Riemann surfaces of the f_i, we simply solve (5.8) in $\mathbb{A}^{\mathrm{icom}}$ by taking $f^{\mathrm{inc}} = (f_1^{\mathrm{inc}}, ..., f_d^{\mathrm{inc}}) \in (\mathbb{A}^{\mathrm{icom}})^n$. Notice that the decomposition of the integral in (5.9) may be used for more general implicit equations involving integration, even if they are not given in normal form (5.8).

Composition. Let us first show how to compute $g \circ f \in \mathbb{A}^{\mathrm{lcom}}$ for given $f, g \in \mathbb{A}^{\mathrm{lcom}}$ with $f(0) = 0$. Assuming by induction over $|\delta|$ that $\delta \in \mathbb{P}^{\mathrm{com}}_{g \circ f} \subseteq \mathbb{P}^{\mathrm{com}}_f$, we denote

$$\begin{aligned} f(\delta) &= (f(\delta_1) - f(\epsilon), ..., f(\delta) - f(\delta_1, ..., \delta_{l-1})) \\ M_\rho &= \| f'_{+\delta} \|_\rho \qquad (\rho \in \mathbb{R}^{\mathrm{com},>}, \rho < \underline{r}_{f_{+\delta}}) \end{aligned}$$

and set

$$\begin{aligned} \underline{r}_{(g \circ f)_{+\delta}} &= \sup \{ \rho \in \mathbb{R}^{\mathrm{com},>} : \rho < \underline{r}_{\Lambda(f)_{+\delta}} \wedge M_\rho < \underline{r}_{g_{+f(\delta)}} \}, \\ \llbracket g \circ f \rrbracket_\rho &= \llbracket \Lambda(g)_{+f(\delta)} \rrbracket_{M_\rho}. \end{aligned}$$

In section 6.2, we will show how to compute $M_\rho \in \mathbb{R}^{\mathrm{com}}$, so that $\underline{r}_{(g \circ f)_{+\delta}} \in \mathbb{R}^{\mathrm{lcom},>}$ and $\llbracket g \circ f \rrbracket_\rho \in \mathbb{R}^{\mathrm{rcom},\geqslant}$.

Assume now that $f, g \in \mathbb{A}^{\mathrm{icom}}$ are such that $f(\bullet_{\mathcal{R}_f}) = \pi(\bullet_{\mathcal{R}_g})$, so that $\Lambda(g \circ f) = \Lambda(g) \circ \Lambda(f) \in \mathbb{A}^{\mathrm{lcom}}$ is well-defined by what precedes. Let $\mathcal{R}_{\Lambda(g\circ f)^\sharp}$ be the canonical Riemann surface of $\Lambda(g \circ f)$, as in theorem 5.7, and let $\mathcal{S}$ be the subspace induced by paths $\delta \in \mathcal{R}_f$ for which $\Lambda(f)(\delta) \in \mathcal{R}_g$. A digital folding η on $\mathcal{R}_g$ is said to be a *digital homotopy* between the paths associated to $\eta(0, \cdot)$ and $\eta(|\eta|_1, \cdot)$ if $\eta(\cdot, 0) = \bullet_{\mathcal{R}_g}$ and $\eta(\cdot, |\eta|_2)$ is constant. The set of pairs $(\delta, \delta') \in \mathbb{P}^{\mathrm{dig}}_{\Lambda(g\circ f)} \cap \mathbb{P}^{\mathrm{dig}}_{\mathcal{S}}$ such that $\Lambda(f)(\delta), \Lambda(f)(\delta') \in \mathbb{P}^{\mathrm{dig}}_{g}$ determine digitally homotopic paths on $\mathcal{R}_g$ is enumerable. We take $\mathcal{R}_{g\circ f} = \mathcal{S}/\!\sim$, where $\sim$ stands for digital homotopy on $\mathcal{R}_g$. We also take

$$X_\delta(g \circ f) = X_{\Lambda(f)(\delta)}(g) \circ X_\delta(f).$$

Remark 5.10. With a bit more effort, the computation of the Riemann surface of $g \circ f$ can be done more efficiently, by directly working with digital approximations and using corollary 4.26.

Heuristic continuation. Assume that we are given a computable convergent power series $\tilde{f}$ at the origin. Sometimes, it is interesting to associate a function $f \in \mathbb{A}^{\mathrm{icom}}$ to $\tilde{f}$ by determining $\underline{r}_{f_{+\delta}}$ and $[\![f_{+\delta}]\!]_\rho$ in a heuristic way. For instance, given the first n coefficients $f_0, \ldots, f_{n-1}$ of f, the radius of convergence may be determined heuristically, by looking at the convex hull of $(i, \log|f_i| + \mathbb{R}^{\leqslant})$ in $\mathbb{R}^2$ and considering the edge from (i, α) to (j, β) with $i \leqslant \lfloor 2\,n/3 \rfloor < j$. Then

$$\log \underline{r}_f \approx -\frac{\beta - \alpha}{j - i}.$$

In order to determine $[\![f]\!]_\rho$, it suffices to compute f_0, $f_1\,\rho$, ... until several successive values $f_n\,\rho^n$ are small with respect to $\max\{|f_0|, |f_1|\,\rho, \ldots, |f_{n-1}|\,\rho^{n-1}\}$ and approximate $f \approx f_0 + \cdots + f_{n-1} z^{n-1}$. A similar approximation may be used for the analytic continuation to a point δ with $|\delta| = \rho$. Finally, one may determine $\mathcal{R}_f$ by heuristically identifying branches in $\mathcal{R}_f^\sharp$ where the germs of f above the same point coincide up to a given order and at a given accuracy.

Remark 5.11. Even if one may not want to crucially depend on heuristic computations, so as to obtain only certified answers, one may still use them as a complement to theoretically correct computations, in order to obtain an idea about the quality of a bound or to guide other computations. For instance, given $f \in \mathbb{A}^{\mathrm{lcom}}$, assume that we want to obtain a lower bound for r_f with "expected relative error" ε. Then we may keep producing better and better lower bounds r_n and heuristic bounds $\tilde{r}_n$ (at expansion order n), until $|r_n/\tilde{r}_n - 1| < \varepsilon$.

5.5. Convolution products

Let $f, g \in \mathbb{A}^{\mathrm{lcom}}$. The convolution product of f and g is locally defined by

$$(f * g)(z) = \int_0^z f(u)\, g(z - u)\, \mathrm{d}u. \tag{5.10}$$

If we want to evaluate $f * g$ up to many digits at a small $z \in \mathbb{C}^{\mathrm{com}}$, then we may simply compute the Taylor series expansions of $f(u)$ and $g(z - u)$ at $u = 0$ and evaluate the primitive of $f(u)\, g(z - u)$. Assuming that the series expansions of $f(u)$ and $g(z - u)$ are given, this algorithm requires a time $O(n^2 \log n \log\log n)$ for the computation of a 2^{-n}-approximation of $(f * g)(z)$. More generally, if $\delta = (\delta_1, \ldots, \delta_l) \in \mathbb{P}^{\mathrm{com}}_f$ is a broken-line path with $\delta^! = (\delta_l, \ldots, \delta_1) \in \mathbb{P}^{\mathrm{com}}_g$, then

$$(f * g)(\delta) = (f * g_{+\delta_l + \cdots + \delta_2})(\delta_1) + (f_{+\delta_1} * g_{+\delta_l + \cdots + \delta_3})(\delta_2) + \cdots + (f_{+\delta_1 + \cdots + \delta_{l-1}} * g)(\delta_l).$$

Modulo the replacement of each δ_i by δ_i/k_i, ..., δ_i/k_i for a sufficiently large $k_i \in \mathbb{N}^{\neq}$, we may thus compute a 2^{-n}-approximation of $(f*g)(\delta)$ using the above method, in time $O(n^2 \log n \log\log n)$.

In order to obtain a complete power series expansion of $f*g$ at 0 or δ, it is convenient to consider the Borel and Laplace transforms

$$\begin{aligned}\mathcal{B}\colon f = \sum_{n=1}^{\infty} f_n z^n &\longmapsto \sum_{n=1}^{\infty} \frac{1}{(n-1)!} f_n z^{n-1}\\ \mathcal{L}\colon f = \sum_{n=0}^{\infty} f_n z^n &\longmapsto \sum_{n=0}^{\infty} n!\, f_n z^{n+1}\end{aligned}$$

Then we have

$$f*g = \mathcal{B}(\mathcal{L}(f)\,\mathcal{L}(g)) \tag{5.11}$$

$$(f*g)_i = \sum_{j=1}^{i} \left[j \binom{i}{j} \right]^{-1} f_{i-1}\, g_{i-j}. \tag{5.12}$$

These formula allow for the efficient (and possibly relaxed) computation of the coefficients $(f*g)_i$, since the Borel and Laplace transforms can be computed in essentially linear time.

More precisely, let $\varepsilon = 2^{-n}$, $k = O(n)$ and $h = f*g$. Assume that $|f_i| \leqslant 1$ and $|g_i| \leqslant 1$ for all i and that we are given ε-approximations $\tilde{f}_0$, ..., $\tilde{f}_{k-1}$ of f_0, ..., f_{k-1} and ε-approximations $\tilde{g}_0$, ..., $\tilde{g}_{k-1}$ of g_0, ..., g_{k-1}. Then the naive evaluation of the formula (5.12) using interval arithmetic yields ε-approximations $\tilde{h}_0, \ldots, \tilde{h}_{k-1}$ of $h_0, \ldots, h_{k-1}$. In order to use (5.11) and fast multiplication methods based on the FFT, one may subdivide the multiplication of $\mathcal{L}(f)$ and $\mathcal{L}(g)$ into squares like in relaxed multiplication method from [vdH02b, Figure 3]. For each square, one may then apply the scaling technique from [vdH02b, Section 6.2.2], so as to allow for FFT-multiplication without precision loss. This yields an $O(n^2 \log^2 n \log\log n)$ algorithm for the computation of ε-approximations for $\tilde{h}_0, \ldots, \tilde{h}_{k-1}$. Notice that this algorithm is relaxed.

If we want the power series expansion of $f*g$ at a path $\delta = \mathbb{P}_f^{\mathrm{com}}$ with $\delta^! \in \mathbb{P}_g^{\mathrm{com}}$, then consider the formula

$$(f*g)(\delta+\varepsilon) = (f*g_{+\varepsilon+\delta_l+\cdots+\delta_2})(\delta_1) + \cdots + (f_{+\delta_1+\cdots+\delta_l}*g)(\varepsilon) \tag{5.13}$$

Assuming that the δ_i and ε are sufficiently small, we also have

$$\begin{aligned}(f_{+\delta_1+\cdots+\delta_{i-1}}*g_{+\varepsilon+\delta_l+\cdots+\delta_{i+1}})(\delta_i) &= (f_{+\delta_1+\cdots+\delta_{i-1}}*g_{+\delta_l+\cdots+\delta_{i+1}})(\delta_i+\varepsilon) -\\ &\quad (f_{+\delta_1+\cdots+\delta_i}*g_{+\delta_l+\cdots+\delta_{i+1}})(\varepsilon),\end{aligned} \tag{5.14}$$

for all $i \in \{1, \ldots, l\}$. Now if δ_i is sufficiently small, we may compute the series expansion of $f_{+\delta_1+\cdots+\delta_{i-1}}*g_{+\delta_l+\cdots+\delta_{i+1}}$ at δ_i as a function of the series expansion of the same function at the origin, using a variant of [vdH02b, Section 3.4.1]. This yields n-digit expansions for $O(n)$ coefficients of $f*g$ at δ in time $O(n^2 \log^2 n \log\log n)$.

Let us now define $\underline{r}_{(f*g)_{+\delta}}$ and $\llbracket (f*g)_{+\delta} \rrbracket_\rho$ by induction over $|\delta|$, in such a way that $\delta \in \mathbb{P}_{f*g}^{\mathrm{com}}$ implies $\delta = \mathbb{P}_f^{\mathrm{com}}$ and $\delta^! \in \mathbb{P}_g^{\mathrm{com}}$. Assuming that $\delta = (\delta_1, \ldots, \delta_l) \in \mathbb{P}_{f*g}^{\mathrm{com}}$, we take

$$\underline{r}_{(f*g)_{+\delta}} = \min\{\underline{r}_{f_{+\delta}}, \underline{r}_g - |\delta_l|, \underline{r}_{g_{+\delta_l}} - |\delta_{l-1}|, \ldots, \underline{r}_{g_{+\delta_l+\cdots+\delta_2}} - |\delta_1|, \underline{r}_{g_{+\delta^!}}\}. \tag{5.15}$$

Clearly, for $\varepsilon \in \mathbb{C}^{\mathrm{com}}$ with $|\varepsilon| < \underline{r}_{(f*g)_{+\delta}}$, we have $\delta+\varepsilon \in \mathbb{P}_f^{\mathrm{com}}$ and $(\delta+\varepsilon)^! \in \mathbb{P}_g^{\mathrm{com}}$. Given $\rho < \underline{r}_{(f*g)_{+\delta}}$, we take

$$\begin{aligned}\llbracket (f*g)_{+\delta} \rrbracket_\rho &= \llbracket f \rrbracket_{\delta_1} \llbracket g_{+\delta_l+\cdots+\delta_2} \rrbracket_{\delta_1+\rho} |\delta_1| + \cdots +\\ &\quad \llbracket f_{+\delta_1+\cdots+\delta_{l-1}} \rrbracket_{\delta_l} \llbracket g \rrbracket_{\delta_l+\rho} |\delta_l| +\\ &\quad \llbracket f_{+\delta} \rrbracket_\rho \llbracket g \rrbracket_\rho \rho\end{aligned} \tag{5.16}$$

This completes the induction and the construction of $f*g \in \mathbb{A}^{\mathrm{lcom}}$. If $f, g \in \mathbb{A}^{\mathrm{gcom}}$, then we have $\mathcal{R}_{(f*g)^\sharp} = \mathcal{R}_f * \mathcal{R}_g$, since (5.15) reduces to (4.3). If $f, g \in \mathbb{A}^{\mathrm{lcom}}$, then we suspect that $\mathcal{R}_{(f*g)^\sharp} = \mathcal{R}_{f^\sharp} * \mathcal{R}_{g^\sharp}$, but we have not tried to check this in detail.

In practice, if we want to analytically continue $f*g$ along a path $\delta \in \mathbb{P}^{\mathrm{com}}$ which is known to belong to $\mathbb{P}^{\mathrm{com}}_{(f*g)^\sharp}$, it can be quite expensive to "randomly" compute a part of $\mathbb{P}^{\mathrm{com}}_{(f*g)^\sharp}$ which contains δ. During the analytic continuation of $f*g$ along δ, it is therefore recommended to progressively compute equivalent paths for (δ_1), (δ_1, δ_2), ..., $(\delta_1, ..., \delta_{|\delta|})$ which avoid singularities as well as possible. These paths may then be used for the computation of better bounds and in order to accelerate the computation of a part of $\mathbb{P}^{\mathrm{com}}_{(f*g)^\sharp}$ which contains δ.

More precisely, let $f, g \in \mathbb{P}^{\mathrm{icom}}$ and assume that $\delta \in \mathbb{P}^{\mathrm{com}}_{f*g}$ is fixed. Let

$$h(\zeta) = f(\zeta)\, g_{+\delta}(-\zeta).$$

By construction, we have $\delta \in \mathcal{R}_h$ and $\mathcal{R}_h$ is the limit of a sequence $\mathcal{R}_0 \xrightarrow{\varphi_0} \mathcal{R}_1 \xrightarrow{\varphi_1} \cdots$ with $\lambda_{\mathcal{R}_0} > \lambda_{\mathcal{R}_1} > \cdots$. Let $n \in \mathbb{N}$ be fixed and consider the set $\mathcal{P}$ of all paths $\varepsilon = (\varepsilon_1, ..., \varepsilon_l) \in \mathbb{P}^{\mathrm{dig}}_{\mathcal{R}_n}$ with $\varepsilon_1, ..., \varepsilon_l \in \lambda_{\mathcal{R}_n} \mathbb{Z}[\mathrm{i}]$. Given $\varepsilon \in \mathbb{P}^{\mathrm{com}}_{\mathcal{R}_n}$, let

$$\ell_\varepsilon = \int_{\phi_{\varepsilon,\mathcal{R}_n}} \left(1 + \frac{1}{r_\zeta^2}\right) \mathrm{d}\zeta.$$

Here r_ζ denotes the distance between ζ and the border of $\mathcal{R}_n$ and we recall that

$$\phi_{\varepsilon,\mathcal{R}_n} \colon [0,1] \to \mathcal{R}_n$$

stands for the continuous path on $\mathcal{R}_n$ associated to ε. Using Dijkstra's shortest path algorithm, we may enumerate $\mathcal{P} = \{\varepsilon^0, \varepsilon^1, ...\}$ such that $\ell_{\varepsilon^0} \leqslant \ell_{\varepsilon^1} \leqslant \cdots$. As soon as we find an ε^i with

$$\| \phi_{\varepsilon^i,\mathcal{R}_n^\sharp} - \phi_{\delta,\mathcal{R}_n^\sharp} \| < \lambda_{\mathcal{R}_n},$$

then we stop (this condition can be checked ultimately by computing a sufficiently precise digital approximation of $\mathcal{R}_n^\sharp$ using the techniques from section 4.7). If $\lambda_{\mathcal{R}_n}$ is small enough, this yields a path $\varepsilon = \varepsilon^i + (\|\delta\| - \|\varepsilon^i\|)$ which is homotopic to δ on $\mathcal{R}_n$ and for which ℓ_ε is small. The idea is now to replace δ by ε in the right-hand side of (5.15) resp. (5.16), if this yields a better bound.

The above approach raises several subtle problems. First of all, the computed path depends on the number n. When computing a k-th approximation for $\underline{r}_{(f*g)_{+\delta}}$, one possibility is to take $n = k$. A second problem is that the choice of ε depends on $\mathcal{R}_f$ and $\mathcal{R}_g$, so we no longer have $\Lambda(X_\delta(f*g)) = \Lambda(f*g)$. Nevertheless, it should be possible to adapt the theory to the weaker condition that $(X_{\delta^k} \circ \cdots \circ X_{\delta^1})(f*g) \sqsubseteq (X_{\varepsilon^l} \circ \cdots \circ X_{\varepsilon^1})(f*g)$ whenever $\{\delta^1, ..., \delta^k\} \subseteq \{\varepsilon^1, ..., \varepsilon^l\}$, where we notice that our change can only lead to improved bounds. Finally, if $\lambda_{\mathcal{R}_n}$ becomes small, then the shortest path algorithm may become inefficient. One approach to this problem would be to use the shortest path at a larger scale for an accelerated computation of the shortest path at a smaller scale. As a first approximation, one may also try to continuously deform ε as a function of δ. We wish to come back to these issues in a forthcoming paper.

6. Bound computations

For actual implementations of computable analytic functions it is very important that bound computations (i.e. lower bounds for convergence radii and upper bounds for the norm on compact disks) can be carried out both accurately and efficiently.

A first problem is to find a good balance between efficiency and accuracy: when bounds are needed during intermediate computations, rough bounds are often sufficient and faster to obtain. However, bad bounds may lead to pessimistic estimates and the computation of more terms in power series expansions in order to achieve a given precision for the end-result. Therefore, it is important that cheap bounds are also reasonably accurate.

Another point is that it is usually a good idea to use different algorithms for rough and high precision bound computations. Indeed, only when sufficient knowledge is gathered about the function using rough bound computations, it is usually possible to fulfill the conditions for applying a high precision method, such as Newton's method. Furthermore, such asymptotically fast methods may only be more efficient when large precisions are required, which requires the study of the trade-off between different methods.

In this section, we will present several techniques for efficient and/or accurate bound computations. Some of the algorithms have been implemented in MMXLIB. However, the topic of bound computations deserves a lot of further study.

6.1. Lower bounds for the smallest zero of an analytic function

Let $f \in \mathbb{A}^{\mathrm{lcom}}$ with $f_0 \neq 0$ and $r = \underline{r}_f$. The problem of computing a lower bound for the radius of convergence of f^{-1} reduces to the computation of a ρ such that f has no zeros on $\mathcal{B}_\rho$. We may start with the simpler problem of computing a lower bound for

$$s = \max \{ s \leqslant \rho : \forall z \in \mathcal{B}_s, f(z) \neq 0 \},$$

where $\rho \in \mathbb{R}^{\mathrm{com},>}$ with $\rho < r$ has been fixed. A natural approach is to approximate the problem by a root finding problem of complex polynomials.

More precisely, we may approximate real and complex numbers by elements of the sets $\mathbb{I}$ and $\mathbb{B}$ of real intervals with endpoints in $\mathbb{R}^{\mathrm{dig}}$ resp. complex balls with centers in $\mathbb{C}^{\mathrm{dig}}$ and radii in $\mathbb{R}^{\mathrm{dig},\geqslant}$ [vdH06b]. Let $M = \| f \|_R$ for some $R \in \mathbb{R}^{\mathrm{com}}$ with $\rho < R < r$. We start by picking $n \in \mathbb{N}$, and the computation of complex ball approximations $\tilde{f}_0, \tilde{f}_1, \ldots, \tilde{f}_{n-1} \in \mathbb{B}$ for $f_0, f_1, \ldots, f_{n-1}$, as well as a bound for the remainder

$$| f_n z^n + f_{n+1} z^{n+1} + \cdots | \leqslant \eta = \frac{M}{1 - \rho/R} \left(\frac{\rho}{R} \right)^n.$$

The bound η may be integrated into the constant coefficient $\tilde{f}_0$ by setting $\tilde{f}_0 := \tilde{f}_0 + \bar{\mathcal{B}}_\eta$. Now we compute a lower bound for the norm of the smallest root of the polynomial

$$P(z) = \tilde{f}_0 + \tilde{f}_1 z + \cdots + \tilde{f}_{n-1} z^{n-1} \in \mathbb{B}[z],$$

using some classical numerical method and interval/ball arithmetic. The result will then be presented as an interval $\tilde{s} = [\underline{s}, \overline{s}] \in \mathbb{I}$ and $\underline{s}$ yields the desired lower bound for s.

We have implemented two experimental versions of the above method for the two numerical methods from [Car96] and a variant of [Pan96, Appendix A]. The first method is based on repeated squaring in the ring $\mathbb{B}[z^n]/P(z)$. However, it is cumbersome to adapt to the case when there exist almost multiple roots. Also, we observed a lot of precision loss in our context of certified computations with complex balls. This might be due to the divisions. The second method is based on Graeffe transforms and rapidly provided us with rough lower bounds for s of an acceptable quality. Let us quickly explain this method.

First of all, we recall that Graeffe's transform sends a polynomial $P(z) = P_n z^n + \cdots + P_0$ of degree n with roots $\alpha_1, \ldots, \alpha_n$ to another polynomial $P^{②}$ with roots $\alpha_1^2, \ldots, \alpha_n^2$. Such a polynomial can be computed efficiently using FFT-squaring:

$$\begin{aligned} P(z) &= P_{\mathrm{odd}}(z^2)\, z + P_{\mathrm{even}}(z^2); \\ P^{②}(z) &= P_{\mathrm{odd}}(z)^2\, z - P_{\mathrm{even}}(z)^2. \end{aligned}$$

Given a monic polynomial $P(z) = z^n + P_{n-1}\,z^{n-1} + \cdots + P_0$ with $\max(|P_{n-1}|, ..., |P_0|) = 1$, we also observe that the norm of the largest root of P lies in the interval $[1/n, 2]$. Indeed, if $|z| > 2$, then $|(P(z) - z^n)/z^n| = |P_{n-1}/z + \cdots + P_0/z^n| < 1$, whence $|P(z)/z^n| > 0$. Similarly, if $P(z) = (z - \alpha_1) \cdots (z - \alpha_n)$ is such that $|\alpha_i| < 1/n$ for all i, then $|P_{n-i}| < \binom{n}{i}/n^i \leqslant 1$ for all $i \in \{1, ..., n\}$.

Now let $P \in \mathbb{B}[z]$ be a polynomial of degree n and assume that we want an upper bound for the largest root of P with a relative accuracy $\varepsilon > 0$. If we rather want a lower bound, then we replace $P(z) = P_0 + \cdots + P_n\,z^n$ by $P(z) = P_0\,z^n + \cdots + P_n$. We start by making P monic by setting $P := P/P_n$. We next let $p \in \mathbb{N}$ be smallest such that $|[1/n, 2]^{1/2^p} - 1| < \varepsilon/2$. Starting with $s := 1$ and $k := 1$, we now repeat the following:

1. Compute $\lambda = [\underline{\lambda}, \overline{\lambda}] := 1/\max(|P_{n-1}|, |P_{n-2}|^{1/2} ..., |P|_0^{1/n}) \in \mathbb{I}$.
2. Scale $P(z) := z^n\,(1 + P_{n-1}\,(\underline{\lambda}/z) + \cdots + P_0\,(\underline{\lambda}/z)^n)$.
3. Replace $s := s/\underline{\lambda}^{1/k}$.
4. If $k = 2^p$, then return $s\,[1/n, 2]^{1/k}\,[\underline{\lambda}/\overline{\lambda}\,, 1]^{1/k}$.
5. Set $P := P^{②}$ and $k := 2\,k$.

Consider the factorizations $P^* = (z - \alpha_1^*) \cdots (z - \alpha_n^*)$ and $P = (z - \alpha_1) \cdots (z - \alpha_n)$, where P^* denotes the original. Then we observe that $\{\alpha_1^*, ..., \alpha_n^*\} = \{s\,\alpha_1^k, ..., s\,\alpha_n^k\}$, each time when we arrive at step 4. When the approximations $P_0, ..., P_n$ were sufficiently precise, it follows that we obtain an ε-approximation of the largest root of P^* on exit.

Remark 6.1. Notice that we simplified the method from [Pan96, Appendix A], since we do not need Turan's proximity test. Instead, we use a variant of bound (B.7) mentioned in Appendix B, by rescaling at each step. Notice that FFT-multiplication leads to huge precision loss when applied to polynomials which have not been scaled.

Remark 6.2. If there exists a unique and simple root α_1 of maximal modulus, then after a few steps, we have $P \approx z^n - \omega\,z^{n-1}$, with $|\omega| = 1$, whence a good approximation of $\alpha_1^{2^k}$ can be read off from P. Now if $P^{②}(\beta) \approx 0$, then either $P(-\sqrt{\beta}) \approx 0$ or $P(\sqrt{\beta}) \approx 0$. Going the way back up, we may thus compute a good approximation of α_1. At a second stage, this approximation may be further improved using Newton's method.

Remark 6.3. The worst case for the above algorithm is when P admits a single root α of multiplicity n. In that case, each iteration typically gives rise to a precision loss of $\log_2 \binom{n}{n/2} = O(n)$ binary digits, when using a fast algorithm for multiplication.

Let us now come back to the original problem of computing a lower bound for the radius $\underline{r}_{f^{-1}}$ of convergence of f^{-1}. Given $n \in \mathbb{N}$, we thus have to find an n-th lower approximation $s_n \in \mathbb{R}^{\mathrm{dig},\geqslant}$ for $\underline{r}_{f^{-1}}$ with $s_0 \leqslant s_1 \leqslant \cdots$ and $\lim_{n\to\infty} s_n = \underline{r}_{f^{-1}}$. We start by computing the n-th lower approximation r_n of r. For ρ, we may now take $(s_{n-1} + r_n)/2$ if $n > 0$ and $r_0/2$ otherwise (alternatively, one may choose ρ as a function of a heuristic approximation of the radius of convergence of f^{-1}; see remark 5.11). Using the above algorithm, we may now compute a lower bound s for $\underline{r}_{f^{-1}}$, using an expansion of f at order n (or an order like $\sqrt{n}$ which makes the total computation time more or less proportional to n) and $\varepsilon = 1/(n+1)$. We may then take $s_n = \max(s_{n-1}, s)$ if $n > 0$ and $s_0 = s$ otherwise.

6.2. Computing extremal values on compact disks

Let $f \in \mathbb{A}^{\mathrm{lcom}}$ and $\rho \in \mathbb{R}^{\mathrm{com},>}$ be such that $\rho < \underline{r}_f$. By definition, we have a method for computing an upper bound $[\![f]\!]_{+\rho}$ for $M = \| f \|_\rho$. Since this bound may be pessimistic, we will now show how to compute better approximations for M.

We start by computing an $R \in \mathbb{R}^{\mathrm{com}}$ with $\rho < R < r_\zeta$ and picking an expansion order $n \in \mathbb{N}$. If we want an ε-approximation of M, then we may take n as in (5.3) and (5.4). We next compute approximations $\tilde{f}_0$, $\tilde{f}_1$, ..., $\tilde{f}_{n-1}$ for the first n coefficients f_0, $f_1\,\rho$, ..., $f_{n-1}\,\rho^{n-1}$ of the series $f(\rho\,z)$ and set $P(z) = \tilde{f}_0 + \tilde{f}_1\,z + \cdots + \tilde{f}_{n-1}\,z^{n-1}$. We now have to approximate

$$\tilde{M} \;=\; \max\{|P(z)|\colon |z| = 1\}.$$

Let $N \in 2^{\mathbb{N}}$ be a power of two larger with $10\,n \leqslant N = O(n)$ and $\omega = \mathrm{e}^{2\pi\mathrm{i}/N}$. We may efficiently approximate the vector $v_0 = (P(1), P(\omega), ..., P(\omega^{N-1}))$ using the FFT and compute

$$V \;=\; \|v_0\|_\infty = \max\{|v_{0,0}|, ..., |v_{0,N-1}|\}.$$

More generally, we may efficiently approximate $v_k = \frac{1}{k!}\,(P^{(k)}(1), P^{(k)}(\omega), ..., P^{(k)}(\omega^{N-1}))$ using the FFT for small values of k. Let $\delta = |\mathrm{e}^{\pi\mathrm{i}/N} - 1| \sim \pi/N$. Then

$$|P(z) - P(\omega^i)| \leqslant |P'(\omega^i)|\,\delta + \cdots + |P^{(k-1)}(\omega^i)|\,\frac{\delta^{k-1}}{(k-1)!} + \|P^{(k)}\|_1\,\frac{\delta^k}{k!},$$

for $|z - \omega^i| \leqslant \delta$, and where $\|Q\|_1 = |Q_0| + \cdots + |Q_{n-1}|$ for polynomials Q of degree $<n$. In other words,

$$|\tilde{M} - V| \leqslant \left\| v_1\,\delta + \cdots + v_{k-1}\,\frac{\delta^{k-1}}{(k-1)!} \right\|_\infty + \|P^{(k)}\|_1\,\frac{\delta^k}{k!}. \tag{6.1}$$

We also have

$$\|v_k\|_\infty\,\frac{\delta^k}{k!} \leqslant \|v_0\|\,\frac{(\delta\,N)^k}{k!},$$

where $\delta\,N < 1/3$. We may thus compute an approximation $|\tilde{M} - V| \leqslant V/2$ using one FFT of order $O(n)$. More generally, for a fixed $\varepsilon > 0$, and modulo choosing a larger $N = O(n)$, we may compute an approximation $|\tilde{M} - V| \leqslant \varepsilon\,V$ using one FFT of order $O(n)$.

In practice, the above method is more powerful. Indeed, if P is a truncated power series, then the right-hand side of (6.1) is usually of the order $O(\|v_0\|/n)$ for a small $k = O(1)$. Also, in the favorable but frequent case when the maximal value of $|P(z)|$ is obtained near a unit ω^i which "clearly dominates the others" (this case typically occurs when we approach an isolated singularity), one may consider the shifted polynomial $P(\omega^i + z)$ and apply Newton's method near ω^i in order to efficiently find high precision approximations of $\tilde{M}$. If the upper bound for $[\![f]\!]_\rho$ was pessimistic, one may also directly recompute the Taylor expansion of $f_{+\rho\omega^i}$ at order n and apply Newton's method for this series. This allows us to use a much sharper bound for the tail of the expansion of $f_{+\rho\omega^i}$ on $\bar{\mathcal{B}}_{\rho\delta}$ than (5.4). Alternatively, one may investigate the use of a steepest descent method. Notice that the method may still be applied in the slightly less favorable case of a small number of units ω^i which dominate the others.

Remark 6.4. One feature of the above method is that it can easily be applied to the computation of approximations of

$$\begin{aligned} M^{\min} &= \min\{|f(z)|\colon z \in \bar{\mathcal{B}}_\rho\};\\ M^{\mathrm{real}} &= \max\{\Re\, f(z)\colon z \in \bar{\mathcal{B}}_\rho\}. \end{aligned}$$

Indeed, it suffices to replace $\tilde{M}$ and V by the corresponding $\tilde{M}^{\min}$, $\tilde{M}^{\mathrm{real}}$ and $V^{\min}$, V^{real}. The efficient computation of $M^{\min}$ and M^{real} is interesting in order to compute upper bounds for f^{-1} resp. $\exp f$ on compact disks. In the case of $M^{\min}$, one needs to require that f has no roots on $\bar{\mathcal{B}}_\rho$, so that $M^{\min} > 0$.

Remark 6.5. The previous remark actually generalizes to extrema of the form

$$M^g \;=\; \|g \circ f\|_\rho,$$

where g is a more general continuous and real-valued function which can be evaluated efficiently. However, suitable analogues of (6.1) are harder to obtain in that case.

6.3. Relaxed Taylor series and bounds for the remainders

In sections 6.1 and 6.2, an important ingredient of the algorithms is the computation of a bound $[\![f_{n;}]\!]_\rho$ for the tail $f_{n;} = f_n z^n + f_{n+1} z^{n+1} + \cdots$ of the power series expansion of f on a compact disk $\bar{\mathcal{B}}_\rho$. Until now, sharp bounds for the tail were obtained by computing a rough bound $[\![f]\!]_R$ on a slightly larger disk and using Cauchy's formula. However, if $[\![f]\!]_R$ is pessimistic, then we will have to choose n quite large in order to reduce the bound for $|f_{n;}|$. This raises the questing of finding more direct ways for bounding $|f_{n;}|$ on $\bar{\mathcal{B}}_\rho$. In this section, we will see how to adapt the strategies of lazy and relaxed computations with formal power series in order to directly take into account error bounds for the tails.

Notations. Given a power series $f \in \mathbb{C}[[z]]$ and $k < n \in \mathbb{N}$, we will denote:

$$\begin{aligned} f_{;n} &= f_0 + \cdots + f_{n-1} z^{n-1} \\ f_{n;} &= f_n z^n + f_{n+1} z^{n+1} + \cdots \\ f_{k;n} &= f_k z^k + \cdots + f_{n-1} z^{n-1} \end{aligned}$$

Assuming algorithms for the computation of bounds $[\![f_{;n}]\!]_\rho$ and $[\![f_{n;}]\!]_\rho$ for $f_{;n}$ resp. $f_{n;}$ on $\bar{\mathcal{B}}_\rho$, we will also denote by $[\![f_{;n;}]\!]_\rho = [\![f_{;n}]\!]_\rho + [\![f_{n;}]\!]_\rho$ the resulting bound for $|f|$ on $\bar{\mathcal{B}}_\rho$. Finally, in the case when $\rho = 1$, then we will abbreviate $[\![f_{;n}]\!]_1$, $[\![f_{n;}]\!]_1$, etc. by $[\![f_{;n}]\!]$, $[\![f_{n;}]\!]$ and so on.

Relaxed power series. We recall that the technique of lazy computations with formal power series relies on the observation that solutions to implicit equations usually can be put into a form which expresses the n-th coefficient of a solution in terms of the previous ones. For instance, if $g = \exp f$ with $f_0 = 0$, then the formula $g = \int f' g$ yields a way to compute the coefficients of g using

$$g_n = \frac{1}{n} (f' g)_{n-1} = \sum_{k=0}^{n-1} \frac{k+1}{n} f_{k+1} g_{n-1-k}.$$

In the case of relaxed computation [vdH02b], additional tricks are used so as to accelerate these computations using FFT-multiplication. This enables us to compute n coefficients in time $O(M(n) \log n)$, where $M(n)$ corresponds to the complexity of multiplication of polynomials of degree n. The lazy and relaxed strategies have the big advantage that the resolution of a functional equation can be done in approximately the same time as the evaluation of the defining implicit equation.

One disadvantage of FFT-multiplication is that it increases numerical instability in the case when the coefficients f_n do not have the same orders of magnitude. Using transformations of the kind $f(z) \mapsto f(r z)$, where r is the "numerical" radius of convergence of f, it has been shown in [vdH02b, Section 6.2] how to reduce this numerical instability. In our case, we are rather interested in the computation of ε-approximations of $f(z)$ for $z \in \bar{\mathcal{B}}_\rho$. Assume that f is the solution of some implicit equation using the operations $+$, $-$, $\times$, $/$, $\mathrm{d}/\mathrm{d}z$, $\int$ and $\circ$. Using the rules

$$\begin{aligned} (f \,\square\, g)(\rho z) &= f(\rho z) \,\square\, g(\rho z) \qquad (\square \in \{+, -, \times, /\}) \\ (f')(\rho z) &= (f(\rho z))'/\rho \\ (\textstyle\int f)(\rho z) &= \rho \textstyle\int f(\rho z) \\ (f \circ g)(\rho z) &= f(\rho z) \circ (g(\rho z)/\rho) \end{aligned}$$

we may then construct an implicit equation for $f(\rho z)$ which can be evaluated as efficiently as f itself. Without loss of generality, we may thus assume that $\rho = 1$ and compute ε'-approximations for the coefficients f_k for an $\varepsilon' < \varepsilon$ which does not depend on k. If we need n coefficients, $\varepsilon' \approx \varepsilon/n$ usually suffices. This trick therefore reduces the general case to fixed point arithmetic and FFT-multiplication of degree n polynomials only accounts for a precision loss of $O(\log n)$ digits.

Bounds for the remainders. Having computed $f_0, \ldots, f_{n-1}$, we have seen in the previous section how to compute a bound $\llbracket f_{;n} \rrbracket \in \mathbb{R}^{\mathrm{dig},\geqslant}$ for $\| f_{;n} \|$. The next question is to compute a bound $\llbracket f_{n;} \rrbracket \in \mathbb{R}^{\mathrm{dig},\geqslant}$ for $\| f_{n;} \|$. Clearly, we may take

$$\begin{aligned}
\llbracket (f+g)_{n;} \rrbracket &= \llbracket f_{n;} \rrbracket + \llbracket g_{n;} \rrbracket && (6.2)\\
\llbracket (f\,g)_{n;} \rrbracket &= \llbracket f_{n;} \rrbracket \, \llbracket g_{;n} \rrbracket + \llbracket f_{;n} \rrbracket \, \llbracket g_{n;} \rrbracket + \llbracket (f_{;n}\, g_{;n})_{n;} \rrbracket && (6.3)\\
\llbracket (\textstyle\int f)_{n;} \rrbracket &= \frac{1}{n+1} \llbracket f_{n;} \rrbracket && (6.4)
\end{aligned}$$

where

$$\llbracket (f_{;n}\, g_{;n})_{n;} \rrbracket \leqslant \sum_{k=0}^{n-1} |f_k| \left(\sum_{l=n-k}^{n-1} |g_l| \right)$$

can be computed in time $O(n)$. One may also compute a bound $\llbracket f'_{n;} \rrbracket$ for $\| f'_{n;} \|$ using automatic differentiation. For especially nice postcompositions, one may take:

$$\begin{aligned}
\llbracket (f \circ (\alpha\, z))_{n;} \rrbracket &= \llbracket f_{n;} \rrbracket \, |\alpha|^n \quad (|\alpha| \leqslant 1); && (6.5)\\
\llbracket (f \circ z^p)_{n;} \rrbracket &= \llbracket f_{\lceil n/p \rceil;n} \rrbracket + \llbracket f_{n;} \rrbracket \quad (p \in \mathbb{N}^{>}). && (6.6)
\end{aligned}$$

For more general postcompositions with g, with $g_0 = 0$, $g_1 \neq 0$ and $\| g \| \leqslant \alpha \leqslant 1$, one may use

$$\llbracket (f \circ g)_{n;} \rrbracket = \llbracket (f_0 + \cdots + f_{n-1}\, g^{n-1})_{n;} \rrbracket + \llbracket f_{n;} \rrbracket \, |\alpha|^n.$$

The case of convolution products will be discussed below.

Implicit equations. Let us now show how to deal with implicit equations. We start with the case when $f = \Phi(f)$ for some expression which involves operations for which we can compute bounds of the type (6.2–6.6). When making the hypothesis that $\llbracket f_{n;} \rrbracket = \lambda$ for some $\lambda \in \mathbb{R}^{\mathrm{com},\geqslant}$, we may formally compute the bound $\varphi(\lambda) = \llbracket \Phi(f)_{n;} \rrbracket$. If $\varphi(\lambda) \leqslant \lambda$, then we claim that the hypothesis was correct and that we may indeed take $\llbracket f_{n;} \rrbracket = \lambda$. Indeed, since the formulas (6.2–6.6) are positive and real analytic, the function $\varphi : \lambda \mapsto \varphi(\lambda)$ is real analytic with a power series expansion which is positive at the origin. Therefore, $0, \Phi(0), \Phi(\Phi(0)), \ldots$ forms a sequence of analytic functions on $\bar{\mathcal{B}}_1$ which converges uniformly to f and such that $\| \Phi^{(i)}_{n;} \| \leqslant \lambda$ for all i. By continuity, it follows that $\| f_{n;} \| \leqslant \lambda$.

In order to find the smallest fixed-point λ_{fix} of φ, we may use the secant method:

$$\begin{aligned}
\lambda_0 &:= 0\\
\lambda_1 &:= \varphi(\lambda_0)\\
\lambda_{k+2} &:= \lambda_k + \frac{\varphi(\lambda_k) - \lambda_k}{\lambda_{k+1} - \varphi(\lambda_{k+1}) + \varphi(\lambda_k) - \lambda_k} (\lambda_{k+1} - \lambda_k)
\end{aligned}$$

If $\lambda_{k+1} < \lambda_k$ for some k or if k exceeds a given threshold, then the method fails and we set $\llbracket f_{n;} \rrbracket = +\infty$. Otherwise, λ_k converges quadratically to λ_{fix}. As soon as $|\lambda_{k+1}/\lambda_k - 1| < \varepsilon$, for some given $\varepsilon > 0$, we check whether $\varphi(\tilde{\lambda}_{\mathrm{fix}}) \leqslant \tilde{\lambda}_{\mathrm{fix}}$ for $\tilde{\lambda}_{\mathrm{fix}} = 2\,\lambda_{k+1} - \lambda_k$, in which case we stop. The resulting $\tilde{\lambda}_{\mathrm{fix}}$ is an approximation of λ_{fix} with relative accuracy $\varepsilon > 0$.

The above technique generalizes to systems $f=(f_1,\ldots,f_d)=\Phi(f)$ of implicit equations. In this case, the hypothesis $\lambda=\VERT f_{n;}\VERT$ and the bound $\varphi(\lambda)=\VERT\Phi(f)_{n;}\VERT$ are vectors and the secant method becomes:

$$\begin{aligned}\lambda_0 &:= 0\\ \lambda_{2k+1} &:= \varphi(\lambda_{2k})\\ \lambda_{2k+2} &:= \lambda_{2k}+\min(\mu_1,\ldots,\mu_d)\,(\lambda_{2k+1}-\lambda_{2k}),\end{aligned}$$

where

$$\mu_i=\frac{\varphi_i(\lambda_{2k})-\lambda_{2k,i}}{\lambda_{2k+1,i}-\varphi_i(\lambda_{2k+1})+\varphi_i(\lambda_{2k})-\lambda_{2k,i}}.$$

We may also consider systems $f=\Phi(f)$ such that Φ is recursively built up using the standard operations $+$, $-$, $\times$, $\int$, etc., together with extra operations like $/$ and exp which involve the recursive resolution of other systems of implicit equations. Indeed, theoretically speaking, such a system may be rewritten as one big system $g=\Psi(g)$ of the above kind. In practice however, we also want to preserve the lazy computation paradigm, which can be achieved by storing the hypotheses $\lambda_i=\VERT(g_i)_{n;}\VERT$ and the corresponding bounds $\lambda(g)_i$ in a hash table, which is passed as a reference to the bound computation method.

Lower bounds for the radius of convergence. Let $\rho\in\mathbb{R}^{\mathrm{eff},>}$ be arbitrary. Modulo a transformation of the type $f(z)\mapsto f(z/\rho)$, the above algorithms can be used in order to compute a possibly infinite upper bound $\VERT f_{;n;}\VERT_\rho$ for $\|f\|_\rho$. In particular, when applying this method for different values of ρ, we obtain an algorithm for computing a lower bound for $\underline{r}_f$. Indeed, we keep decreasing or increasing ρ depending on whether $\VERT f\VERT_\rho=\infty$ resp. $\VERT f\VERT_\rho<\infty$. More precisely, assuming that $\rho\in[\rho_0/\sigma_0,\rho_0\sigma_0]$ for a starting approximation ρ_0 and $\sigma_0>1$, we keep setting $\sigma_{k+1}=\sqrt{\sigma_k}$ and $\rho_{k+1}:=\rho_k\,\sigma_{k+1}^{\pm1}$ at each iteration, until we obtain an adequate precision. When a starting approximation is not beforehand, one may use a second iteration $\rho'_k=2^k$ resp. $\rho'_k=2^{-k}$ in order to obtain a reasonable value for ρ_0, while taking $\sigma_0=2$.

Let us now consider the dependence of the computation of $\VERT f_{n;}\VERT_\rho$ for a solution to $f=\Phi(f)$ as a function of ρ (assuming that we perform the necessary scalings for each ρ). When the implicit equation was constructed using $+$, $-$, $\times$, $\int$ and recursive solutions to implicit equations of the same kind, then it can be checked that

$$\varphi(\lambda)=O(\rho^n)+O(\rho)\,\lambda+O(\lambda^2) \tag{6.7}$$

for $\rho\to0$. Consequently, the function φ indeed does have a fixed point for sufficiently small ρ, and our algorithm yields a computable lower bound for $\underline{r}_f$. In particular, our technique can be used as an alternative for the classical majorant method [vK75, vdH03]. Moreover, it easily adapts to slightly more general functional equations, which involve composition or other operations: it suffices to check that (6.7) holds for $\rho\to0$.

Assuming that lower bounds for radii of convergence are computed as above, we claim that $\mathcal{R}_{f^\sharp}$ coincides with the largest theoretical simply connected Riemann surface $\tilde{\mathcal{R}}$ on which f and $\Phi(f)$ are defined. In order to see this, we first observe that the algorithm for computing $\VERT f_{+\delta}\VERT_\rho$ may theoretically be applied to arbitrary paths $\delta\in\mathbb{P}_{\tilde{\mathcal{R}}}$ and $\rho\in\mathbb{R}^>$ with $|\rho|<r_{\delta_{\tilde{\mathcal{R}}}}$. Since Φ was constructed using the common operations $+$, $-$, $\times$, $\int$, etc., we have $\VERT f_{+\delta'}\VERT_\rho=\VERT f_{+\delta}\VERT_\rho$ whenever $\delta'_{\tilde{\mathcal{R}}}=\delta_{\tilde{\mathcal{R}}}$ and $\VERT f_{+\delta}\VERT_\rho$ depends continuously on $\delta_{\tilde{\mathcal{R}}}$ and ρ. Consequently, the supremum

$$\underline{r}_\zeta=\sup\{\rho>0\colon\VERT f_{+\delta}\VERT_\rho<\infty,\zeta=\delta_{\tilde{\mathcal{R}}}\}>0$$

is lower continuous in ζ. Now assume for contradiction that $\mathcal{R}_{f^\sharp} \subsetneq \tilde{\mathcal{R}}$ and take

$$\zeta \in (\tilde{\mathcal{R}} \cap \partial \mathcal{R}_{f^\sharp}) \setminus \mathcal{R}_{f^\sharp}.$$

Setting $\varepsilon = \underline{r}_\zeta / 2 > 0$, there exists a neighbourhood $\mathcal{U} \subseteq \tilde{\mathcal{R}}$ of ζ with $\underline{r}_\xi > \varepsilon$ for all $\xi \in \mathcal{U}$. Taking $\xi \in \mathcal{U} \cap \mathcal{R}_{f^\sharp}^{\mathrm{com}}$ with $|\xi - \zeta| < \varepsilon$, we thus obtain $\zeta \in \mathcal{B}_{\xi,\varepsilon} \subseteq \mathcal{R}_{f^\sharp}$. This contradiction completes the proof of our claim. Notice the analogy with [vdH05a, Theorem 3].

Composition equations. The case of implicit equations which involve compositions has to be treated with additional care. For instance, consider an equation of the type

$$f = \Phi(f, f \circ g_1, \ldots, f \circ g_p). \tag{6.8}$$

Assuming that the equation admits a solution at the origin, its analytic continuation to ζ requires the prior analytic continuation of f to $g_{i_1} \circ \cdots \circ g_{i_k}(\zeta)$ for any $i_1, \ldots, i_k \in \{1, \ldots, p\}$ and $k \geqslant 1$. Naive implementations may therefore lead to infinite loops.

One solution to this problem is to introduce a "freezing" operator $\dashv$. Given $f \in \mathbb{A}^{\mathrm{icom}}$, the function $f^\dashv$ is the restriction of f to its current Riemann surface $\mathcal{R}_f$. In particular, $\underline{r}_{f^\dashv + \delta} = r_{\delta_{\mathcal{R}_f}}$ for all $\delta \in \mathbb{P}_{\mathcal{R}_f}^{\mathrm{com}}$. Then we may replace (6.8) by

$$f = \Phi(f, f^\dashv \circ g_1, \ldots, f^\dashv \circ g_p).$$

This approach avoids infinite loops, by handing over to the user the responsibility of ensuring that all values $f(g_{i_1} \circ \cdots \circ g_{i_k}(\zeta))$ with $k \geqslant 1$ are already defined. Of course, this may be automatized by trying brutal continuations in all directions. One may also consider delayed freezing operators $\dashv_n$, which only freeze f after n postcompositions.

In the very particular case when the g_i generate a finite group $\mathcal{G}$ for the composition operator, we notice that (6.8) may be rewritten as a system of card $\mathcal{G}$ equations in the unknowns $f \circ g$ with $g \in \mathcal{G}$. After a local resolution at the origin, these equations do no longer involve composition. A particularly important special case of this situation is when $k = 1$ and $g_1 = q\, z$ with $q^n = 1$.

Convolution equations. The power series expansion of the analytic continuation $(f * g)_{+\delta}$ of a convolution product may be computed using (5.13) and (5.14). Unfortunately, the translation of a power series by a small δ is not very convenient for relaxed computations, which naturally occur if f and g are unknowns in a convolution equation [É85], such as

$$f = (1 - z)^{-1} + f * f.$$

Nevertheless, in the equation (5.14), the functions $f_{+\delta_1 + \cdots + \delta_{i-1}}$ and $g_{+\delta_l + \cdots + \delta_{i+1}}$ are known except when $i = 1$ resp. $i = l$. Modulo one subdivision of the path, we may also assume without loss of generality that $l \geqslant 2$. This reduces the resolution of the convolution equation to the problem of determining the coefficients of $f * g$ at a small δ as a function of the coefficients of f at δ in a relaxed manner, assuming that the coefficients of g at δ are already known. Now we may again write

$$(f * g)(\delta + \varepsilon) = (f_{+\delta} * g)(\varepsilon) + (f * g_{+\varepsilon})(\delta). \tag{6.9}$$

The coefficients of $f_{+\delta} * g$ may be computed in a relaxed manner by what precedes. The second member may be expanded in ε using

$$(f * g_{+\varepsilon})(\delta) = (f * g)(\delta) + (f * g')(\delta)\, \varepsilon + \frac{1}{2} (f * g'')(\delta)\, \varepsilon^2 + \cdots. \tag{6.10}$$

However, the evaluation of each $(f*g^{(i)})(\delta)/i!$ at a precision of n digits still requires a time $O(n^2 \log n \log\log n)$, which is not very convenient if we want to evaluate up to order $i \leqslant n$. On the other hand, if the power series expansion of $(f*g)(\varepsilon)$ has convergence radius r, then the translated expansion of $(f*g)(\delta+\varepsilon)$ still has convergence radius $r-\delta$. The idea is now to use (6.9) and (6.10) for the computation of good bounds $\llbracket((f*g)_{+\delta})_{n;}\rrbracket_\rho$ and not for the expansion of $(f*g)_{+\delta}$ itself, using the formulas

$$\begin{aligned}
\llbracket(f*g)_{n;}\rrbracket_\rho &= \llbracket(f_{;n}*g_{;n})_{n;}\rrbracket_\rho + \\
&\quad \frac{1}{n+1}\left(\llbracket f_{;n}\rrbracket_\rho\, \llbracket g_{n;}\rrbracket_\rho + \llbracket f_{n;}\rrbracket_\rho\, \llbracket g_{;n}\rrbracket_\rho\right) + \\
&\quad \frac{1}{2\,n+1}\,\llbracket f_{n;}\rrbracket_\rho\, \llbracket g_{n;}\rrbracket_\rho \\
\llbracket(f*g_{+\cdot})(\delta)_{n;}\rrbracket_\rho &= \frac{1}{n!}\,\llbracket f_{;n;}\rrbracket_\delta\, \llbracket g_{n;}^{(n)}\rrbracket_{\delta+\rho}
\end{aligned}$$

If $|\delta|$ is close to r, then $\llbracket((f*g)_{+\delta})_{n;}\rrbracket_\rho$ may typically remain finite even for $\rho > r-|\delta|$. In that case, we have a method to analytically continue $f*g$ beyond $\bar{\mathcal{B}}_r$.

Remark 6.6. With the above method, in order to obtain an order n expansion of the solution f to a convolution equation at a path $\delta=(\delta_1,\ldots,\delta_l)$, one generally needs an order $k\,n$ expansion of f at the origin, where k is more or less proportional to $|\delta_1|+\cdots+|\delta_l|$ (it also depends on the positions of the singularities of f). It remains an interesting question whether the order $k\,n$ can be reduced.

6.4. Improved bounds for remainders of Taylor series

Division. The error bounds computed in section 6.3 are not optimal in the case of division

$$f=\frac{1}{1-\varepsilon}=1+\varepsilon\,f \qquad (\varepsilon_0=0) \tag{6.11}$$

Indeed, the fixed-point method yields

$$\llbracket f_{n;}\rrbracket=\begin{cases}\dfrac{\llbracket f_{;n}\rrbracket\,\llbracket\varepsilon_{n;}\rrbracket+\llbracket(f_{;n}\,\varepsilon_{;n})_{n;}\rrbracket}{1-\llbracket\varepsilon_{;n;}\rrbracket} & \text{if } \llbracket\varepsilon_{;n;}\rrbracket<1\\ +\infty & \text{otherwise}\end{cases}$$

The denominator $1-\llbracket\varepsilon_{;n;}\rrbracket$ is unnecessarily pessimistic: even if $\|\varepsilon\|$ exceeds 1, the function ε itself might be bounded away from 1. This is particularly annoying in the case when $\varepsilon=\mathrm{e}^{\alpha z}-1$ for large values of α. Indeed, when using the fixed-point method in a direct way on this example, the computable radius of convergence of f would be $O(\alpha^{-1})$ instead of $+\infty$.

For this reason, it is good to treat the case of division (6.11) in an *ad hoc* manner. When rewriting (6.11) in terms of $f_{n;}$, we obtain the solution

$$f_{n;}=\frac{1+\varepsilon\,f_{;n}-f_{;n}}{1-\varepsilon}.$$

Now we may compute a lower bound M for $1-\varepsilon=1-\varepsilon_{;n}+\bar{\mathcal{B}}_{\llbracket\varepsilon_{n;}\rrbracket}$ on $\bar{\mathcal{B}}_1$ using the technique from section 6.2. Consequently, we may take

$$\llbracket f_{n;}\rrbracket=\frac{\llbracket(1+\varepsilon\,f_{;n}-f_{;n})_{n;}\rrbracket}{M}.$$

Exponentiation. Similarly, when applying the technique from the previous section to the case of exponentiation

$$f=\mathrm{e}^g=\int g'\,f, \tag{6.12}$$

we obtain a bound

$$\llbracket f_{n;}\rrbracket = \begin{cases} \frac{\llbracket f_{;n}\rrbracket\,\llbracket g'_{n;}\rrbracket + \llbracket (f_{;n}\,g'_{;n})_{n;}\rrbracket}{n+1-\llbracket g'_{n;}\rrbracket} & \text{if } \llbracket g'_{;n;}\rrbracket < n+1 \\ +\infty & \text{otherwise} \end{cases}$$

Although this bound is a bit better than in the bound for division (roughly speaking, we effectively "see" the part of f with $|f(z)| \leqslant \mathrm{e}^{O(n^2)}$), we again obtain a better *ad hoc* bound by solving (6.12) in terms of $f_{n;}$:

$$f_{n;} = \mathrm{e}^g \int (f_{;n}\,g' - f'_{;n})\,\mathrm{e}^{-g}.$$

Section 6.2 again yields an efficient algorithm for computing order n bounds $M_>$ and $M_<$ for $|\mathrm{e}^g|$ and $|\mathrm{e}^{-g}|$ on $\bar{\mathcal{B}}_1$. We may then take

$$\llbracket f_{n;}\rrbracket = M_>\,M_<\,\llbracket (f_{;n}\,g' - f'_{;n})_{n;}\rrbracket.$$

Implicit equations. Let us now return to the case of a general implicit equation $f = \Phi(f)$ and again consider the decomposition $f = f_{;n} + f_{n;}$. We may rewrite each subexpression $g = \Psi(f)$ of $\Phi(f)$ as $g = g^\circ + g^*\,f_{n;}$, where g° and g^* are new expressions in $f_{n;}$, such that g^* corresponds to the "coefficient of $f_{n;}$" in $\Psi(f)$:

$$\begin{aligned} f^\circ &= f_{;n} & f^* &= 1 \\ (g+h)^\circ &= g^\circ + h^\circ & (g \pm h)^* &= g^* \pm h^* \\ (g\,h)^\circ &= g^\circ\,h^\circ + g^*\,h^*\,f_{n;}^2 & (g\,h)^* &= g^*\,h^\circ + g^\circ\,h^* \\ (\textstyle\int g)^\circ &= \textstyle\int (g^\circ + g^*\,f_{n;}) & (\textstyle\int g)^* &= 0 \end{aligned}$$

Composition is treated in a similar way as integration. Applying the above rules to $\Phi(f)$, we obtain

$$\begin{aligned} f_{n;} &= \Phi(f) - f_{n;} \\ &= (\Phi(f)^\circ - f_{n;}) + \Phi(f)^*\,f_{n;} \\ &= \Xi_0(f_{n;}) + \Xi_1(f_{n;})^*\,f_{n;}. \end{aligned}$$

We now replace the equation $f = \Phi(f)$ by

$$f_{n;} = \frac{\Xi_0(f_{n;})}{1 - \Xi_1(f_{n;})}$$

and compute bounds $\llbracket (f_{n;})_{;n}\rrbracket = 0$ and $\llbracket (f_{n;})_{n;}\rrbracket$ as in the previous section with the above improvement for the final division by $1 - \Xi_1(f_{n;})$. In the case of possibly nested systems of implicit equations $f = (f_1, \ldots, f_d) = \Phi(f)$, subexpressions $g = \Psi(f)$ are decomposed as

$$g = g^\circ + g^* \cdot f_{n;},$$

where g^* is a vector and $\cdot$ stands for the vector product.

Example 6.7. Consider the implicit equation

$$f = z + \int z\,f + f^2. \tag{6.13}$$

For $n \geqslant 2$, we have

$$\begin{aligned} \Phi(f)^\circ &= z + \textstyle\int z\,(f^\circ + f_{n;}) + (f^\circ)^2 + f_{n;}^2 \\ \Phi(f)^* &= 2\,f^\circ \end{aligned}$$

and

$$\Phi(f)^\circ - f^\circ = P(z) + \textstyle\int z\,f_{n;} + f_{n;}^2$$

for the polynomial $P = z + \int z\, f^\circ + (f^\circ)^2$ with $P_{n;} = 0$. Then (6.13) is equivalent to

$$f_{n;} = \frac{P(z) + \int z\, f_{n;} + f_{n;}^2}{1 - 2\, f^\circ}.$$

Dynamical systems. Instead of taking $(\int g)^* = 0$ in the above case of implicit equations, it would be nice to rather extract the linear part of $\Phi(f)$ in f. Unfortunately, the resulting linear equation in $f_{n;}$ is often not so easy to solve. Nevertheless, for implicit equations of a particular shape, such a resolution may be feasible. For instance, consider the case of an ordinary differential equation

$$f = \int \Phi(f), \tag{6.14}$$

where $\Phi(f)$ is an expression which is also a power series in f. We may then rewrite (6.14) as

$$\begin{aligned} f_{n;} &= -f^\circ + \int (\Phi(f)^\circ + \Phi(f)^*\, f_{n;}) \\ &= \Xi_0(f_{n;}) + \int \Xi_1(f_{n;})\, f_{n;}. \end{aligned} \tag{6.15}$$

We next set

$$\begin{aligned} \Xi_0(f_{n;}) &= P_0(z) + \bar{\mathcal{B}}_{\lambda_0}; \\ \Xi_1(f_{n;}) &= P_1(z) + \bar{\mathcal{B}}_{\lambda_1}, \end{aligned}$$

for polynomials $P_0 = 0, P_1$ of degree $< n$ and numbers λ_0 and λ_1 which are approximated at successive stages using the secant method. Then (6.15) admits an explicit solution

$$f_{n;} = \mathrm{e}^{\int P_1(z) + \bar{\mathcal{B}}_{\lambda_1}} \int \bar{\mathcal{B}}_{\eta_0}\, \mathrm{e}^{-\int P_1(z) + \bar{\mathcal{B}}_{\lambda_1}}.$$

Now order n upper bounds for $M_> = \|\mathrm{e}^{\int P_1(z) + \bar{\mathcal{B}}_{\lambda_1}}\|$ and $M_< = \|\mathrm{e}^{-\int P_1(z) + \bar{\mathcal{B}}_{\lambda_1}}\|$ can be computed using the method from section 6.2. Then we may take

$$[\![f_{n;}]\!] = \lambda_0\, M_>\, M_<.$$

With some more work, this method can be adapted to the case of systems of ordinary differential equations (6.14), with $f = (f_1, \ldots, f_d)$ and $\Phi = (\Phi_1, \ldots, \Phi_d)$. The case when Φ is polynomial can also be treated with the majorant technique [vdH03, Section 5].

6.5. Approaches for limiting the precision loss

Computing in the jet space. Consider the solution f to some ordinary differential equation $\Phi(f, f', \ldots, f^{(r)}) = 0$ with given initial conditions $(f(0), \ldots, f^{(r-1)}(0)) = (\lambda_0, \ldots, \lambda_{r-1})$ at the origin. Assuming that $\lambda_0, \ldots, \lambda_{r-1}$ are given by complex ball representations $\bar{\mathcal{B}}_{\lambda_0^*, \rho_0}$, ..., $\bar{\mathcal{B}}_{\lambda_{r-1}^*, \rho_{r-1}}$, we may in principle compute coefficients of f using complex ball arithmetic. However, this may lead to overestimation of the error due to the fact that we do not keep track of possible cancellations between the errors in $\lambda_0, \ldots, \lambda_{r-1}$ during the computation.

One approach to this problem is to use Taylor models [MB96, Ber98] in which we consider $\lambda_0 = \lambda_0^* + \varepsilon_0$, ..., $\lambda_{r-1} = \lambda_{r-1}^* + \varepsilon_{r-1}$ as formal parameters, with $\varepsilon_0 \in \bar{\mathcal{B}}_{\rho_0}$, ..., $\varepsilon_{r-1} \in \bar{\mathcal{B}}_{\rho_{r-1}}$. Instead of computing with coefficients in $\mathbb{B}$, we now compute with coefficients in the jet space

$$\begin{aligned} \mathbb{B}[[\varepsilon_0, \ldots, \varepsilon_{r-1}]]_d &= \{x = \textstyle\sum_{|i| < d} x_i\, \varepsilon^i\} \\ |i| &= i_0 + \cdots + i_{r-1} \\ \varepsilon^i &= \varepsilon_0^{\alpha_0} \cdots \varepsilon_{r-1}^{\alpha_{r-1}} \end{aligned}$$

For i and j with $|i+j| \geqslant d$, we take $\varepsilon^i \varepsilon^j = \bar{\mathcal{B}}_{\rho^{i+j}}$. Given $x \in \mathbb{B}[[\varepsilon_0, ..., \varepsilon_{r-1}]]_d$, the constant coefficient x_0 is stored at a precision which is one or a few words higher than the precision of the λ_k.

Taylor models can be used in many variants. For instance, each of the coefficients x_i with $|i| \neq 0$ may be taken to be finite precision floating point numbers instead of balls, in which case rounding errors are incorporated into the error of x_0. If $d = 2$, then one may also take $\varepsilon^i \varepsilon^j = \bar{\mathcal{B}}_{\rho^i} \varepsilon^j + \bar{\mathcal{B}}_{\rho^j} \varepsilon^i$ ($|i| = |j| = 1$), which allows for the computations of bounds for the derivatives in the parameters ε_i. If f is continued analytically from 0 to z, then we also notice that the initial conditions $\lambda'_0, ..., \lambda'_{r-1}$ at z may again be taken in the jet space $\mathbb{B}[[\varepsilon_0, ..., \varepsilon_{r-1}]]_d$ for the errors $\varepsilon_0, ..., \varepsilon_{r-1}$ at 0. This is useful for the computation of return maps and limit cycles. When the constant coefficients of such jets become less precise than $\rho_0, ..., \rho_{r-1}$, it may sometimes be useful to *unjettify* $\lambda'_0, ..., \lambda'_{r-1}$ and replace each $x = \lambda'_k$ by $\sum_{|i|<d} x_i \bar{\mathcal{B}}_{\rho^i}$. We next *rejettify* the vector λ' by replacing each $\lambda'_k = \bar{\mathcal{B}}_{(\lambda'_k)^*, \rho'_k}$ by $\lambda'_k = (\lambda'_k)^* + \varepsilon'_k$.

Remark 6.8. The jet-space technique can also be used for studying the dependence of the analytic continuation of f on initial conditions. For instance, return maps for limit cycles may be computed in such a way.

Remark 6.9. Clearly, the technique of jettification is not limited to differential equations: it applies to more general functional equations whose local expansions are determined by the equation in terms of a finite number of initial conditions.

The wrapping effect. A well known problem with certified integration of dynamical systems using interval methods is the wrapping effect [Moo66, Loh01, MB04]. Consider a simple equation like

$$f'' + f = 0.$$

Given an initial condition

$$\begin{pmatrix} f(t_0) \\ f'(t_0) \end{pmatrix} = F_{t_0}$$

at t_0, integration of the equation from t_0 to $t_1 = t_0 + \pi/4$ yields $F_{t_1} = \Delta_{t_0 \to t_1} F_{t_0}$ with

$$\Delta_{t_0 \to t_1} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

Now if F_{t_0} is given by an enclosing rectangle, then left multiplication by $\Delta_{t_0 \to t_1}$ turns this rectangle by $\pi/4$, so that we lose $1/2$ bit of precision when enclosing the result by a new rectangle.

Now a similar problem is encountered when using complex interval arithmetic in order to compute the n-th power of $1+\mathrm{i}$ using $(1+\mathrm{i})^n = (1+\mathrm{i})\,(1+\mathrm{i})^{n-1}$. Therefore, one possible remedy is to adapt ball arithmetic to matrices and represent transition matrices such as $\Delta_{t_0 \to t_1}$ by balls $\Delta_{t_0 \to t_1} = M + \bar{\mathcal{B}}_\varepsilon$, where M is an exact matrix and $\bar{\mathcal{B}}_\varepsilon$ denotes the space of matrices E with norm $\|E\| \leqslant \varepsilon$ (i.e. $\|EV\| \leqslant \varepsilon \|V\|$ for all vectors V). In this representation, taking the (naive) n-th power of the matrix $\Delta_{t_0 \to t_1}$ only gives rise to a precision loss of $O(\log n)$ bits. Although the above idea applies well to matrices whose eigenvalues are of approximately the same order of magnitude, the error bounds may again be pessimistic in other cases. In addition, one may therefore adapt the norms in an incremental manner using numerical preconditioning. Equivalently, one may adapt the coordinate system as a function of $\Delta_{t_0 \to t_i}$ [Loh88, MB04].

We notice that the divide and conquer technique may also be used to the wrapping effect. Indeed, in the case of linear equations, the transition matrices verify the relation

$$\Delta_{t_0\to t_n} = \Delta_{t_{n-1}\to t_n} \cdots \Delta_{t_0\to t_1}.$$

Instead of computing $\Delta_{t_0\to t_n}$ in the naive way, using

$$\Delta_{t_0\to t_1} = \Delta_{t_{n-1}\to t_n} (\cdots \Delta_{t_2\to t_3} (\Delta_{t_1\to t_2} (\Delta_{t_0\to t_1}))),$$

one may use binary splitting:

$$\Delta_{t_i\to t_j} = \Delta_{t_{\lfloor (i+j)/2\rfloor}\to t_j} \Delta_{t_i\to t_{\lfloor (i+j)/2\rfloor}}.$$

Even in the case when ordinary interval arithmetic is used in order to represent the matrices $\Delta_{t_i\to t_j}$, the computation of $\Delta_{t_0\to t_n}$ using this technique gives rise to a precision loss of only $O(\log n)$ bits. With some more work, this technique also applies to non-linear equations.

Preservation laws. In the case when a dynamical system is subject to preservation laws or symmetries, then one may project the bounding region for the numerical solution on the variety of actual solutions, after each numerical integration step. In lucky cases, this may help to further reduce overestimation and precision loss.

Acknowledgment. The author would like to thank the anonymous referees for their careful reading an corrections.

GLOSSARY

Bibliography

[AH83] G. Alefeld and J. Herzberger. *Introduction to interval analysis*. Academic Press, 1983.

[Alb80] O. Alberth. *Computable analysis*. McGraw-Hill, 1980.

[BB56] C.A. Briot and J.C. Bouquet. Propriétés des fonctions définies par des équations différentielles. *Journal de l'École Polytechnique*, 36:133–198, 1856.

[BB85] E. Bishop and D. Bridges. *Foundations of constructive analysis*. Die Grundlehren der mathematische Wissenschaften. Springer, Berlin, 1985.

[BBH01] J. Blanck, V. Brattka, and P. Hertling, editors. *Computability and complexity in analysis*, volume 2064 of *Lect. Notes in Comp. Sc.* Springer, 2001.

[Ber98] M. Berz. Cosy infinity version 8 reference manual. Technical Report MSUCL-1088, Michigan State University, 1998.

[BK75] R.P. Brent and H.T. Kung. $O((n \log n)^{3/2})$ algorithms for composition and reversion of power series. In J.F. Traub, editor, *Analytic Computational Complexity*, 1975. Proc. of a symposium on analytic computational complexity held by Carnegie-Mellon University.

[BK78] R.P. Brent and H.T. Kung. Fast algorithms for manipulating formal power series. *Journal of the ACM*, 25:581–595, 1978.

[Bla02] J. Blanck. General purpose exact real arithmetic. Technical Report CSR 21-200, Luleå University of Technology, Sweden, 2002. `http://www.sm.luth.se/~jens/`.

[BPR03] S. Basu, R. Pollack, and M.F. Roy. *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and computation in mathematics*. Springer-Verlag, 2003.

[Bre76a] R.P. Brent. The complexity of multiprecision arithmetic. In *Complexity of Computational problem solving*, 1976.

[Bre76b] R.P. Brent. Fast multiple-precision evaluation of elementary functions. *Journal of the ACM*, 23(2):242–251, 1976.

[Car96] J. P. Cardinal. On two iterative methods for approximating the roots of a polynomial. In J. Renegar, M. Shub, and S. Smale, editors, *Proc. AMS-SIAM Summer Seminar on Math. of Numerical Analysis*, volume 32 of *Lectures in Applied Math.*, pages 165–188, Providence, 1996. American Mathematical Society Press. Park City, Utah.

[CC90] D.V. Chudnovsky and G.V. Chudnovsky. Computer algebra in the service of mathematical physics and number theory (computers in mathematics, stanford, ca, 1986). In *Lect. Notes in Pure and Applied Math.*, volume 125, pages 109–232, New-York, 1990. Dekker.

[CT65] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Computat.*, 19:297–301, 1965.

[DL89] J. Denef and L. Lipshitz. Decision problems for differential equations. *The Journ. of Symb. Logic*, 54(3):941–950, 1989.

[É85] J. Écalle. *Les fonctions résurgentes I–III*. Publ. Math. d'Orsay 1981 and 1985, 1985.

[FHL+05] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. Mpfr: A multiple-precision binary floating-point library with correct rounding. Technical Report RR-5753, INRIA, 2005.

[GPR03] M. Grimmer, K. Petras, and N. Revol. Multiple precision interval packages: Comparing different approaches. Technical Report RR 2003-32, LIP, École Normale Supérieure de Lyon, 2003.

[Grz55] A. Grzegorczyk. Computable functionals. *Fund. Math.*, 42:168–202, 1955.

[Grz57] A. Grzegorczyk. On the definitions of computable real continuous functions. *Fund. Math.*, 44:61–71, 1957.

[JKDW01] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied interval analysis*. Springer, London, 2001.

[Kar91] E.A. Karatsuba. Fast evaluation of transcendental functions. *Problems of Information Transmission*, 27:339–360, 1991.

[KO63] A. Karatsuba and J. Ofman. Multiplication of multidigit numbers on automata. *Soviet Physics Doklady*, 7:595–596, 1963.

[Lam06] B. Lambov. The RealLib project. `http://www.brics.dk/~barnie/RealLib`, 2001–2006.

[Loh88] R. Lohner. *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendugen*. PhD thesis, Universität Karlsruhe, 1988.

[Loh01] R. Lohner. On the ubiquity of the wrapping effect in the computation of error bounds. In U. Kulisch, R. Lohner, and A. Facius, editors, *Perspectives of enclosure methods*, pages 201–217. Springer, 2001.

[MB96] K. Makino and M. Berz. Remainder differential algebras and their applications. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational differentiation: techniques, applications and tools*, pages 63–74, SIAM, Philadelphia, 1996.

[MB04] K. Makino and M. Berz. Suppression of the wrapping effect by taylor model-based validated integrators. Technical Report MSU Report MSUHEP 40910, Michigan State University, 2004.

[Mül00] N. Müller. iRRAM, exact arithmetic in C++. `http://www.informatik.uni-trier.de/iRRAM/`, 2000.

[MM96] V. Ménissier-Morain. Arbitrary precision real arithmetic: design and algorithms. `http://www-calfor.lip6.fr/~vmm/documents/submission_JSC.ps.gz, 1996.`

[Moo66] R.E. Moore. *Interval analysis*. Prentice Hall, Englewood Cliffs, N.J., 1966.

[MP00] M. N. Minh and M. Petitot. Lyndon words, polylogarithms and the Riemann ζ function. *Discrete Maths*, 217(1–3):273–292, 2000.

[Neu90] A. Neumaier. *Interval methods for systems of equations*. Cambridge university press, Cambridge, 1990.

[O'C05] R. O'Connor. A monadic, functional implementation of real numbers. Technical report, Institute for Computing and Information Science, Radboud University Nijmegen, 2005.

[Pan96] Victor Y. Pan. On approximating complex polynomial zeros: Modified quadtree (Weyl's) construction and improved Newton's iteration. Technical Report RR-2894, INRIA Sophia, 1996.

[Ric92] D. Richardson. The elementary constant problem. In *Proc. ISSAC '92*, pages 108–116, 1992.

[Ric97] D. Richardson. How to recognise zero. *JSC*, 24:627–645, 1997.

[SS71] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing*, 7:281–292, 1971.

[Tur36] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Maths. Soc.*, 2(42):230–265, 1936.

[vdH97] J. van der Hoeven. *Automatic asymptotics*. PhD thesis, École polytechnique, France, 1997.

[vdH99a] J. van der Hoeven. Fast evaluation of holonomic functions. *TCS*, 210:199–215, 1999.

[vdH99b] J. van der Hoeven. GMPX, a C-extension library for gmp. `http://www.math.u-psud.fr/~vdhoeven/`, 1999. No longer maintained.

[vdH01a] J. van der Hoeven. Fast evaluation of holonomic functions near and in singularities. *JSC*, 31:717–743, 2001.

[vdH01b] J. van der Hoeven. Zero-testing, witness conjectures and differential diophantine approximation. Technical Report 2001-62, Prépublications d'Orsay, 2001.

[vdH02a] J. van der Hoeven. A new zero-test for formal power series. In Teo Mora, editor, *Proc. ISSAC '02*, pages 117–122, Lille, France, July 2002.

[vdH02b] J. van der Hoeven. Relax, but don't be too lazy. *JSC*, 34:479–542, 2002.

[vdH03] J. van der Hoeven. Majorants for formal power series. Technical Report 2003-15, Université Paris-Sud, Orsay, France, 2003.

[vdH05a] J. van der Hoeven. Effective complex analysis. *JSC*, 39:433–449, 2005.

[vdH05b] J. van der Hoeven. Efficient accelero-summation of holonomic functions. Technical Report 2005-54, Univ. Paris-Sud, Orsay, France, 2005. Accepted for JSC.

[vdH06a] J. van der Hoeven. Computations with effective real numbers. *TCS*, 351:52–60, 2006.

[vdH06b] J. van der Hoeven. Effective real numbers in Mmxlib. In D. Saunders, editor, *Proc. ISSAC '06*, Genova, Italy, July 2006.

[vdH07] J. van der Hoeven. Around the numeric-symbolic computation of differential Galois groups. *JSC*, 42(1–2):236–264, 2007.

[vK75] S. von Kowalevsky. Zur Theorie der partiellen Differentialgleichungen. *J. Reine und Angew. Math.*, 80:1–32, 1875.

[Wei00] K. Weihrauch. *Computable analysis*. Springer-Verlag, Berlin/Heidelberg, 2000.

[Zei90] D. Zeilberger. A holonomic systems approach to special functions identities. *Journal of Comp. and Appl. Math.*, 32:321–368, 1990.