University of London
Imperial College of Science, Technology and Medicine
Department of Computing

Exact Real Arithmetic using Möbius Transformations

by

Peter John Potts

A thesis submitted for the degree of
Doctor of Philosophy of the University of London
and the Diploma of Imperial College.

July 1998

Abstract

In this thesis, we develop a domain theoretic and computationally feasible framework for exact real arithmetic. We present a formal account of incremental digit representations born out of domain theory, which includes the redundant binary representation and continued fraction representation. The generalization of both these fundamental representations for the real numbers leads to the notion of a general normal product constructed using Möbius transformations. In this thesis, we develop the work of Vuillemin, Nielsen and Kornerup, and show that incrementality and efficiency can be simultaneously achieved in exact real arithmetic. We examine a specialization of general normal products called exact floating point with elegant mathematical properties on the one-point compactification of the real line. Real functions are captured by the composition of 2-dimensional Möbius transformations, leading to the notion of expression trees. Various reduction rules and a lazy form of information flow analysis is used to allow expression trees to be converted efficiently into the exact floating point representation. Algorithms for the basic arithmetic operations and the transcendental functions are presented using the redundant if statement for range reduction and various expression trees derived from the theory of continued fractions. Finally, we present a practical implementation in the functional programming language called Miranda and examine the mathematical properties of two theoretical languages with an exact real number data type.

Acknowledgements

I would like to thank my parents without whose support and encouragement I would never have returned to academia after six years in industry.

I must also thank my supervisor Abbas Edalat for his endless enthusiasm and availability. In particular, we spent many hours trying to come up with a sound theoretical representation for the real numbers using Möbius transformations. This early work culminated in the notions of general normal product and exact floating point, which formed the basis of a joint paper accepted by the Mathematical Foundations of Programming Semantics (MFPS) [20]. Later, Abbas formed a Real Group at Imperial College and arranged weekly meetings. Many ideas were formed, tried and tested in these meetings. The Real Group included Lindsay Errington, Martín Escardó, Reinhold Heckmann, Luis Lamb, Valerie Ménissier-Morain and Philipp Sünderhauf. Lindsay and I discussed some of the more practical aspects of the algorithms. Martín helped me understand the important role of theoretical languages and together with Abbas we had a paper accepted by Logic in Computer Science (LICS) [63]. Reinhold examined the complexity of many of my algorithms and managed to improve some and come with up some new ones [32, 34, 33]. Luis helped me by proof reading this thesis and making me aware of many interesting articles. Valerie also proof read this thesis. She also installed a version of the functional programming language called CAML with arbitrary integer arithmetic. This allowed me to produce my fastest implementation of exact real arithmetic in a functional programming language. Philipp made many contributions to discussions especially during the latter days. He also helped Abbas proof read this thesis. Philipp also edited a joint paper together with Abbas and myself accepted by the International Conference on Functional Programming [21].

Finally, I would like to thank the Engineering and Physical Sciences Research Council for funding me.

Some of the algorithms in this thesis were derived with the help of Mathematica [78] and tested in CAML [76] with BigNum [35].

This document was written using Scientific Workplace, Maple, Mathematica, Metafile Companion and Paul Taylor's Commutative Diagrams package.

Contents

Chapter 1

Introduction

Real numbers are usually represented by finite strings of decimal digits $\{0,1,2,3,4,5,6,7,8,9\}$ and a decimal point $\{.\}$. This is called the decimal representation. In a computer, it is more usual to consider finite strings of binary digits $\{0,1\}$. In both cases, the representation specifies a function that maps finite strings to real numbers. However, finite strings of digits can only represent a limited subset of the real numbers exactly. This means that most real numbers are represented by nearby real numbers or enclosing real intervals with distinct end points giving rise to the notion of round off errors. This is generally accepted for a wide range of applications. However, it is well known that the accumulation of round off errors due to a large number of calculations can produce grossly inaccurate or even incorrect results.

Interval analysis [48] has been used to partially circumvent this problem by maintaining a pair of bounding numbers that is guaranteed to contain the real number or interval in question. However, this interval can get unjustifiably large and thereby convey very little information.

Alternatively, by allowing infinite strings of digits all the real numbers can be represented exactly. There have been a number of theoretical and practical attempts to find a viable framework for exact real arithmetic. Broadly speaking they fall into three categories:

1. Infinite sequences of linear maps proposed by Avizienis [2] and appeared in the work of Watanuki and Ercegovac [74], Boehm and Cartwright [5], Di Gianantonio [10, 13, 12], Escardó [23], Nielsen and Kornerup [51] and Ménissier-Morain [47].

2. Continued fraction expansions proposed by Gosper [28], developed by Peyton-Jones [54] and Vuillemin [71], implemented by Lester [45] and advanced more recently by Kornerup and Matula [39, 40, 41, 42].

3. **Infinite composition of Möbius transformations** generalizes the other two frameworks as demonstrated by Vuillemin [71]. Nielsen and Kornerup [51] showed that this framework can be used to represent quasi-normalized floating point [74].

In this thesis, we explore and develop techniques for computing real numbers on-line with infinite precision. The emphasis has been to improve efficiency, while at the same time maintaining an elegant framework and a sound theoretical basis. We start by noting that every real number can be represented by a sequence of nested closed intervals whose lengths converge to zero. The intersection of these intervals is a singleton set whose element is the real number being represented. Using a computability argument, we justify an extension of the real numbers with infinity and bottom.

In 1972, Gosper [28] presented techniques for computing with exact real numbers using continued fractions and Möbius transformations. Vuillemin [71], Lester [45], Nielsen and Kornerup [51] developed these techniques in the intervening years. In this thesis, we present a sound theoretical basis for these techniques utilizing domain theory and recursion theory. Domain theory enables classical spaces, such as the set of real numbers, to be embedded onto the set of maximal elements of a suitable continuous (or algebraic) domain, which provides a good computational model for these spaces [15, 14, 58, 60, 16, 18, 23, 19, 22].

In this thesis, we formalize the notion of a digital representation of the real numbers and present the notion of a general normal product based on the infinite composition of Möbius transformations (also known as homographies and linear fractional transformations). We then explore the relevant properties of 2-dimensional Möbius transformations, first used by Gosper [28] and Vuillemin [71], to perform the basic arithmetic operations on redundant continued fractions. We show that the standard real number representations such as the decimal, continued fractions and redundant binary fit into this framework for the real numbers. In fact, we show that the standard real number representations are just special cases of general normal products. We then go on to emphasize the role of redundancy, including the redundant if operator, in order to achieve effective representations of the real numbers and real functions. Watanuki and Ercegovac [74] devised a redundant version of floating point and then Nielsen and Kornerup [51] extended this representation to infinite precision using a representation based on the infinite product of matrices each with an associated binary state. We present this technique in our sound theoretical setting and introduce our own simplified variation called (unbiased) exact floating point [61, 62, 20]. We explore the theoretical basis for exact floating point and show that it has fundamental mathematical properties on the one-point compactification of the real line. We highlight the crucial features of this representation in an attempt to answer the challenge by Vuillemin

[71] to find a rational choice for a normal form. For this reason, much of this thesis revolves around developing algorithms for exact floating point.

We then extend the notion of a general normal product to expression trees. Expression trees allow real functions and real expressions to be represented elegantly and provide a link back to domain theory. New algorithms [59] are presented for the elementary functions derived from the theory of continued fractions. A new algorithm is presented for pi derived from a formula by Ramanujan [30]. This algorithm is particularly exciting because it can be easily implemented using numerous parallel processes. We then quantize information and use this idea to analyze the flow of information around an expression tree in order to improve temporal efficiency. We also examine the storage requirements of an expression tree in order to improve spatial efficiency. We then bring these ideas together and present an efficient algorithm for converting an expression tree into the exact floating point representation.

In chapter 12, we describe an actual implementation of exact real arithmetic written in the functional programming language called Miranda [36], which is similar to Haskell.

Finally, chapter 13 of this thesis is devoted to the presentation of theoretical languages based on the Programnming Language for Computable Functions [55, 29] incorporating data types for infinite precision real numbers represented by general normal products. We present models for these languages and show that they are sound and adequate with respect to their reduction rules. Consequently, this confirms the correctness of some of the fundamental aspects of the more practical algorithm presented earlier in this thesis including the role of the redundant if operator.

Chapter 2

Notation and Background

In this chapter, we outline the background knowledge and the notation used in the remaining chapters of this thesis.

2.1 Sets and Functions

A *function* (or *map*) $f : X \to Y$ is a relation between the *domain* (or *source*) X and the *codomain* (or *target*) Y such that for each $x \in X$, there is a unique $y \in Y$ such that $(x, y) \in f$ and we say that f maps x to y, denoted $f(x)$. The set of all $y \in Y$ such that $f(x) = y$ for some $x \in X$ is called the *image* (or *range*) of f. Given X and Y, the set of all functions with domain X and codomain Y is written Y^X.

A *partial function* $f : X \rightsquigarrow Y$ is a function from a subset of X to Y. The function $f : X \rightsquigarrow Y$ is a *total function* if the domain of f is X. The *lift* $X_\perp$ of a set X is the disjoint union of X and the singleton of *bottom* $\perp$. The *lift* of a partial function $f : X \rightsquigarrow Y$ is the total function $f_\perp : X \to Y_\perp$ such that for every element x in the domain of f, $f_\perp$ maps x to $f(x)$ and for each element x not in the domain of f, $f_\perp$ maps x to $\perp$. For convenience, the subscript $\perp$ will be dropped whenever it is clear to do so.

The *powerset* $\mathcal{P}(X)$ of a set X is the set of all subsets of X. Let $\mathcal{P}_f(X)$ and $\mathcal{P}^*(X)$ denote the set of finite and non-empty subsets of X respectively. The *canonical extension* of a function $f : X \to Y$ is the total function $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$ such that $\mathcal{P}(f)(Z) = \{f(z) | \, z \in Z\}$. The *canonical extension* of a partial function $f : X \rightsquigarrow Y$ is the function $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$ such that

$$\mathcal{P}(f)(Z) = \begin{cases} \{f(z) | \, z \in Z\} & \text{if } f(z) \text{ is defined for all } z \in Z \\ Y & \text{otherwise.} \end{cases}$$

For convenience, the functor $\mathcal{P}$ will be dropped whenever it is clear to do so.

We will use the *barred arrow notation* $\mapsto$ to provide an anonymous notation for functions. For example, the function that squares its input can be denoted $x \mapsto x^2$. Also, we will interpret a subscript as function application. For instance, given a function $f : X \to Y$

$$f_x = f(x).$$

Let

$$\begin{aligned}\mathbb{N}(n) &= \{0, 1, 2, \ldots, n-1\} \\ \mathbb{Z}(n) &= \{1-n, 2-n, \ldots, 0, 1, 2, \ldots, n-2, n-1\}\end{aligned}$$

for $n \in \mathbb{N} - \{0\}$.

2.2 Sequences

An infinite sequence $x_0, x_1, x_2, \ldots \in X$, written $\langle x_n \rangle_{n=0}^{\infty}$, corresponds to the function $x : \mathbb{N} \to X$. A finite sequence of elements $x_0, x_1, x_2, \ldots, x_m \in X$, written $\langle x_n \rangle_{n=0}^{m}$, corresponds to the partial function $n \mapsto \begin{cases} x_n & \text{if } n \le m \\ \perp & \text{otherwise} \end{cases} : \mathbb{N} \to X$. In other words, the elements of a sequence taken from a set X can be specified using a partial function $\mathbb{N} \rightsquigarrow X$.

2.3 Semigroups, Monoids and Groups

A *semigroup* is a set endowed with an associative binary operation. A *monoid* is a semigroup with an identity element. A *group[9]* is a monoid with inverse elements. The order of a group G, denoted by $|G|$, is the number of elements in G. If H is a subgroup of G, denoted by $H \le G$, then

$$Ha = \{ha \,|h \in H\}$$

is called a *right coset* and

$$aH = \{ah \,|h \in H\}$$

is called a *left coset*. The index of H in G, denoted by $|G : H|$, is the number of distinct cosets. A subgroup N of G is called a *normal subgroup*, denoted by $N \trianglelefteq G$, if

$$\forall a \in G \cdot Na = aN.$$

For a normal subgroup N, the set of cosets

$$G/N = \{Ng \,|g \in G\}$$

is a group under multiplication defined by

$$(aN)(bN) = (ab)N.$$

The group G/N is called the *quotient group* of G by the normal subgroup N. A *homomorphism* between groups is a mapping $\theta : G \to H$ such that $\theta(ab) = \theta(a)\theta(b)$. An *isomorphism* is a bijective homomorphism and two groups are said to be *isomorphic*, denoted by $\cong$, if there exists an isomorphism between them. The *image* $\mathsf{image}(\theta)$ of a homomorphism $\theta : G \to H$ is given by

$$\mathsf{image}(\theta) = \{\theta(g) \mid g \in G\}.$$

The *kernel* $\mathsf{kernel}(\theta)$ of a homomorphism $\theta : G \to H$ is given by

$$\mathsf{kernel}(\theta) = \{g \in G \mid \theta(g) = 1\}.$$

An *action* of a group G on a set Ω is a mapping $\mu : \Omega \times G \to \Omega$ such that

$$\begin{aligned} \mu(\omega, 1) &= \omega \\ \mu(\mu(\omega, g), h) &= \mu(\omega, gh). \end{aligned}$$

Theorem 1 (First Isomorphism Theorem) *If $\theta : G \to H$ is a homomorphism then*

$$\mathsf{image}(\theta) \cong G/\mathsf{kernel}(\theta).$$

The element $x^{-1}gx$ is called the *conjugate* of g by x. The set of all conjugates of g

$$\{x^{-1}gx \mid x \in G\}$$

is the *conjugacy class* of g in G.

Let $GL(n, F)$ denote the group of $n \times n$ non-singular matrices with coefficients taken from the field F, known as the *n-dimensional general linear group* over F[50]. Let I denote the identity matrix. Let F^* denote the multiplicative group of non-zero elements of the field F. It can be shown that the set of scalar matrices is a normal subgroup of $GL(n, F)$. The *projective linear group* $GL(n, F)$ is defined to be the quotient group $GL(n, F)/F^*$.

2.4 Topological Spaces

A *topological space* $(A, \mathcal{B})$ consists of a non-empty set A together with a collection $\mathcal{B}$ of subsets of A such that the set A and the *empty set* $\emptyset$ are members of $\mathcal{B}$ and $\mathcal{B}$ is closed under finite intersections and arbitrary unions [68]. A *cover* for a set

A is a collection $\mathcal{B}$ of sets such that $A \subseteq \bigcup_{B \in \mathcal{B}} B$. A topological space is *compact* if every cover by open subsets has a finite subcover. We will consider an *interval* in a topological space X to be any connected subspace of X. Any closed bounded interval $[a, b]$ in $\mathbb{R}$ is compact. Let $\mathbb{I}X$, $\mathbb{I}^{\mathrm{O}}X$, $\mathbb{I}^{\mathrm{C}}X$ and $\mathbb{I}^{\mathrm{F}}X$ denote the set of all, open, closed and compact intervals of X. For any compact interval $[a, b]$ in the set of real numbers $\mathbb{R}$, let

$$\begin{aligned} \underline{[a,b]} &= a \\ \overline{[a,b]} &= b \\ \mathsf{width}\left([a,b]\right) &= b - a. \end{aligned}$$

2.5 Trees

A *tree* is an acyclic, connected graph with one node designated as the *root* of the tree [27]. A tree can be constructed recursively. A single node is a tree. If $T_0, T_1, \ldots, T_n$ are trees with roots $r_0, r_1, \ldots, r_n$ then the graph formed by attaching a new node r by a single arc to each of $r_0, r_1, \ldots, r_n$ is a tree with root r. The nodes $r_0, r_1, \ldots, r_n$ are *children* of r, and r is a *parent* of $r_0, r_1, \ldots, r_n$. A *binary tree* is a tree, where each node has at most two children. The *depth* of a node in a tree is the length of the path from the root to the node; the root itself has depth 0.

2.6 Automata

The tuple (S, I, O, f_S, f_O) is an *automaton* if S is a set of *states*, I is a set of input symbols (the *input alphabet*), O is a set of output symbols (the *output alphabet*), $f_S : S \times I \to S$ is a *state transition function* and $f_O : S \to O$ is an *output function [27]*. The automaton is always initialized to begin in a fixed starting state $s_0 \in S$. A sequence of input symbols $\langle i_n \rangle_{n=0}^{\infty}$ generates a sequence of output symbols $\langle o_n \rangle_{n=0}^{\infty}$ according to the prescription

$$\begin{aligned} s_{n+1} &= f_S\left(s_n, i_n\right) \\ o_n &= f_O\left(s_n\right). \end{aligned}$$

2.7 Category Theory

Definition 2 *A* category $\mathcal{C}$ *is specified by*

- *a class of* objects $\mathsf{ob}(\mathcal{C})$,

- *a class of* morphisms $\mathsf{mor}(\mathcal{C})$,
- *a* source *function* $\mathsf{src}:\mathsf{mor}(\mathcal{C}) \rightarrow \mathsf{ob}(\mathcal{C})$,
- *a* target *function* $\mathsf{tar}:\mathsf{mor}(\mathcal{C}) \rightarrow \mathsf{ob}(\mathcal{C})$,
- *an associative* composition *function* $\circ : \mathsf{mor}(\mathcal{C}) \times \mathsf{mor}(\mathcal{C}) \rightsquigarrow \mathsf{mor}(\mathcal{C})$ *where* $f \circ g$ *is defined if* $\mathsf{tar}(g) = \mathsf{src}(f)$ *with* $\mathsf{src}(f \circ g) = \mathsf{src}(g)$ *and* $\mathsf{tar}(f \circ g) = \mathsf{tar}(f)$ *and*
- *an* identity *morphism* $\mathsf{id}_x : x \rightarrow x$ *exists for each object* $x \in \mathsf{ob}(\mathcal{C})$ *which is* unitary*; namely* $\mathsf{id}_{\mathsf{tar}(f)} \circ f = f$ *and* $f \circ \mathsf{id}_{\mathsf{src}(f)} = f$.

A *functor* $F : \mathcal{C} \rightarrow \mathcal{D}$ between categories $\mathcal{C}$ and $\mathcal{D}$ is specified by an operation $F : \mathsf{ob}(\mathcal{C}) \rightarrow \mathsf{ob}(\mathcal{D})$ and an operation $F : \mathsf{mor}(\mathcal{C}) \rightarrow \mathsf{mor}(\mathcal{D})$such that $F(\mathsf{id}_X) = \mathsf{id}_{FX}$ and $F(f \circ g) = F(f) \circ F(g)$.

A *monad* (T, η, μ) in a category $\mathcal{C}$ consisting of a functor $T : \mathcal{C} \rightarrow \mathcal{C}$, a *unit* operator $\eta_X : X \rightarrow T(X)$ and a *multiplication* operator $\mu_X : T(T(X)) \rightarrow T(X)$ such that η and μ are natural transformations

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\eta_X"'] & Y \arrow[d, "\eta_Y"] \\
T(X) \arrow[r, "T(f)"'] & T(Y)
\end{tikzcd}
\qquad
\begin{tikzcd}
T^2(X) \arrow[r, "T^2(f)"] \arrow[d, "\mu_X"'] & T^2(Y) \arrow[d, "\mu_Y"] \\
T(X) \arrow[r, "T(f)"'] & T(Y)
\end{tikzcd}$$

and the following diagrams commute

$$\begin{tikzcd}
T(X) \arrow[r, "\eta_{T(X)}"] \arrow[dr, "\mathsf{id}_{T(X)}"'] & T^2(X) \arrow[d, "\mu_X"] & T(X) \arrow[l, "T(\eta_X)"'] \arrow[dl, "\mathsf{id}_{T(X)}"] \\
 & T(X) &
\end{tikzcd}
\qquad
\begin{tikzcd}
T^3(X) \arrow[r, "T(\mu_X)"] \arrow[d, "\mu_{T(X)}"'] & T^2(X) \arrow[d, "\mu_X"] \\
T^2(X) \arrow[r, "\mu_X"'] & T(X).
\end{tikzcd}$$

2.8 Domain Theory

Most programming languages allow recursive definitions in which the name of the entity being defined can "recur" in its own definition. The mathematical theory known as "domain theory" provides the appropriate semantic constructions to interpret such definitions. Domain theory was introduced by Dana Scott in 1970 as a mathematical theory of programming languages [67], and for nearly a quarter of a century developed almost exclusively in connection with denotational semantics in computer science.

In the denotational semantics of programming languages, the meaning of a program is taken to be an element of a domain. A domain is a mathematical structure consisting of a set of values and an ordering relation, denoted $\sqsubseteq$, on those values. Domain theory is the study of such structures equipped with a notion of completion and approximation [1, 17].

The fundamental idea of a domain is encapsulated by the notion of a partial order. A *partial ordered set* (or *poset*) $(D, \sqsubseteq)$ is a set D with a binary relation $\sqsubseteq$ that is reflexive

$$\forall x \in D \cdot x \sqsubseteq x,$$

transitive

$$\forall x, y, z \in D \cdot x \sqsubseteq y \wedge y \sqsubseteq z \Rightarrow x \sqsubseteq z$$

and antisymmetric

$$\forall x, y \in D \cdot x \sqsubseteq y \wedge y \sqsubseteq x \Rightarrow x = y.$$

A partial order can be thought of as possible states of a computer ordered by stored information. A *preorder* $(D, \sqsubseteq)$ is a set D with a binary relation $\sqsubseteq$ that is reflexive and transitive only. A subset A of a poset D is an *upper set* if $x \in A$ implies that $y \in A$ for all $x \sqsubseteq y$. The set of all elements above some element in A is denoted by $\uparrow A$. The dual notions are *lower set* and $\downarrow A$. A subset A of a poset D is *directed* if it is non-empty and each pair of elements of A has an upper bound in A. A poset D is *pointed* if it has a least element and we call this element *bottom* $\perp$. Different bottoms will not be distinguished explicitly. The *least upper bound* (or *supremum* or *join*) and the *greatest lower bound* (or *infimum* or *meet*) of a subset A of a poset D are denoted by $\sqcup A$ and $\sqcap A$ respectively. Directed lower sets are called *ideals*. Ideals of the form $\downarrow \{x\}$ are called *principal ideals*. A function f from a poset D to a poset E is *monotone* if $f(x) \sqsubseteq f(y)$ whenever $x \sqsubseteq y$ for all $x, y \in D$. The *monotone function space* between a poset D and a poset E is the poset consisting of the set of monotone functions from D to E ordered pointwise.

A poset D in which every directed subset has a supremum is called a *directed complete partial order* (or *dcpo* or *domain*). The *flat domain* $D_\perp$ of a set D is the dcpo $(D \cup \{\perp\}, \sqsubseteq)$ where $x \sqsubseteq y$ if $x = y \vee x = \perp$. The *lift* of a dcpo $(D, \sqsubseteq)$, consisting of the set $D_\perp = D \cup \{\perp\}$ together with the partial ordering $\sqsubseteq_\perp$ where $x \sqsubseteq_\perp y$ if $x \sqsubseteq y \vee x = \perp$, is a dcpo. The *product* $D \times E$ of a dcpo D with a dcpo E, together with coordinatewise order, is a dcpo. The *smash product* $D \otimes E$ of a dcpo D with a dcpo E, consisting of the set

$$\{(x, y) \in D \times E | x \neq \perp \wedge y \neq \perp\} \cup \{\perp\}$$

together with coordinatewise order, is a dcpo. A function f from a dcpo D to a dcpo E is (*Scott*) *continuous* if it is monotone and $f\left(\bigsqcup A\right) = \bigsqcup f(A)$ for all directed subsets A of D. The *function space* $[D \to E]$ between a dcpo D and a dcpo E is the dcpo consisting of the set of continuous functions from D to E ordered pointwise. Every continuous function f on a pointed dcpo has a least fixed point given by $\bigsqcup_{n=0}^{\infty} f^n(\bot)$. Define

- the down function from a dcpo $D_\bot$ to a dcpo D by

$$\begin{aligned} \mathsf{down} &: D_\bot \to D \\ \mathsf{down}(x) &= \begin{cases} x & \text{if } x \in D \\ \bot & \text{if } x = \bot, \end{cases} \end{aligned}$$

- the up function from a dcpo D to a dcpo $D_\bot$ by

$$\begin{aligned} \mathsf{up} &: D \to D_\bot \\ \mathsf{up}(x) &= x, \end{aligned}$$

- the smash function from a product dcpo $D \times E$ to a smash product dcpo $D \otimes E$ by

$$\begin{aligned} \mathsf{smash} &: D \times E \to D \otimes E \\ \mathsf{smash}(x, y) &= \begin{cases} (x, y) & \text{if } x \neq \bot \wedge y \neq \bot \\ \bot & \text{otherwise}, \end{cases} \end{aligned}$$

- the $\mathsf{unsmash}$ function from a smash product dcpo $D \otimes E$ to a product dcpo $D \times E$ by

$$\begin{aligned} \mathsf{unsmash} &: D \otimes E \to D \times E \\ \mathsf{unsmash}(z) &= \begin{cases} z & \text{if } z = (x, y) \\ (\bot, \bot) & \text{if } z = \bot \end{cases} \end{aligned}$$

- and the fix function from a dcpo of continuous functions $[D \to D]$ to a dcpo D by

$$\mathsf{fix}(f) = \bigsqcup_{n=0}^{\infty} f^n(\bot).$$

Let x and y be elements of a dcpo D. We say that x is *way below* or *approximates* y, denoted $x \ll y$, if for all directed subsets A of D, $y \sqsubseteq \sqcup A$ implies $\exists z \in A$

such that $x \sqsubseteq z$. We say that x is *compact* (or *finite*) if it approximates itself. Let $K(D)$ be the set of compact elements and

$$\begin{aligned} \uparrow x &= \{y \in D \mid x \sqsubseteq y\} \\ \downarrow x &= \{y \in D \mid y \sqsubseteq x\} \\ \twoheaduparrow x &= \{y \in D \mid x \ll y\} \\ \twoheaddownarrow x &= \{y \in D \mid y \ll x\}. \end{aligned}$$

A subset B of a dcpo D is a *basis* for D if $\twoheaddownarrow x \cap B$ is directed and $\sqcup(\twoheaddownarrow x \cap B) = x$ for all $x \in D$. A *continuous domain* is a dcpo with a basis. It can be shown, in this case, that if $x \ll y$ then $\exists z \in D$ such that $x \ll z$ and $z \ll y$. An *algebraic domain* is a dcpo with a basis of compact elements. A *ω-continuous domain* is a dcpo with a countable basis. An *ω-algebraic domain* is a dcpo with a countable basis of compact elements. The Scott topology of a dcpo $(D, \sqsubseteq)$ consists of all the Scott open subsets A of D which are upward closed

$$\forall x \in A \cdot x \sqsubseteq y \Rightarrow y \in A$$

and inaccessible by least upper bounds of directed subsets

$$\forall \text{ directed } B \subseteq A \cdot \sqcup B \in A \Rightarrow A \cap B \neq \emptyset.$$

Dually, a Scott closed set is downward closed and contains all the least upper bounds of directed subsets. In particular, the set $\downarrow x$ is Scott closed.

For a continuous domain $(D, \sqsubseteq)$, the sets $\twoheaduparrow x$ for all $x \in D$ are Scott open and form a basis for the Scott topology.

A dcpo is *bounded complete* if every bounded subset has a least upper bound. Note that a bounded complete dcpo is trivially pointed.

A *Scott domain* is a bounded complete ω-algebraic domain. Also, the dcpo $[D \to E]$ is a Scott domain if D and E are Scott domains. The *step function* $d \searrow e$ from a Scott domain D to a Scott domain E is given by

$$(d \searrow e)(x) = \begin{cases} e & \text{if } d \sqsubseteq x \\ \bot & \text{otherwise} \end{cases}$$

for any $d \in K(D)$ and $e \in K(E)$.

A *continuous Scott domain* is a bounded complete ω-continuous domain. Also, the dcpo $[D \to E]$ is a continuous Scott domain if D and E are continuous Scott domains. The *step function* $d \searrow e$ from a continuous Scott domain D to a continuous Scott domain E is given by

$$(d \searrow e)(x) = \begin{cases} e & \text{if } d \ll x \\ \bot & \text{otherwise} \end{cases}$$

for any $d \in D$ and $e \in E$. These step functions form a basis for the $[D \to E]$.

An *abstract basis* $(B, \prec)$ consists of a set B together with a transitive binary relation $\prec$ such that

$$A \prec x \Rightarrow \exists y \in B \cdot A \prec y \prec x$$

for all $x \in B$ and $A \in \mathcal{P}_f(B)$, where $A \prec x$ denotes $\forall a \in A \cdot a \prec x$. Given an *abstract basis* $(B, \prec)$, $A \subseteq B$ is called an *ideal* if it is *downward closed*

$$a \in A \wedge b \prec a \Rightarrow b \in A$$

and *directed*

$$\forall a, b \in A \cdot \exists c \in A \cdot a \prec c \wedge b \prec c.$$

The set of ideals is denoted by $\overline{(B, \prec)}$. The *ideal completion* of $(B, \prec)$ is the set of ideals ordered by set inclusion. The ideal completion of a preorder is an algebraic domain. The compact elements are the principal ideals. For any algebraic domain D, the ideal completion of $(K(D), \sqsubseteq)$ is isomorphic to D. For any continuous domain D with basis B, the ideal completion of $(B, \ll)$, where $\ll$ is the restriction of the way below relation of D to B, is isomorphic to D.

Definition 3 *Given an abstract basis $(B, \prec)$, define the relations $\prec^{\#}$, $\prec^{\flat}$ and $\prec^{\natural}$ on the finite non-empty subsets $\mathcal{P}^*_f(B)$ of B by*

$$\begin{array}{lcll} X & \prec^{\#} & Y & \textit{iff } \forall y \in Y \cdot \exists x \in X \cdot x \prec y \\ X & \prec^{\flat} & Y & \textit{iff } \forall x \in X \cdot \exists y \in Y \cdot x \prec y \\ X & \prec^{\natural} & Y & \textit{iff } X \prec^{\#} Y \wedge X \prec^{\flat} Y. \end{array}$$

It can be shown that $\left(\mathcal{P}^*_f(B), \prec^{\#}\right)$, $\left(\mathcal{P}^*_f(B), \prec^{\flat}\right)$ and $\left(\mathcal{P}^*_f(B), \prec^{\natural}\right)$ are abstract bases. Given an algebraic domain $(D, \sqsubseteq)$, the ideal completions

$$\begin{array}{rcl} \mathrm{UPPER}(D) & = & \left(\overline{\left(\mathcal{P}^*_f(K(D)), \sqsubseteq^{\#}\right)}, \subseteq\right) \\ \mathrm{LOWER}(D) & = & \left(\overline{\left(\mathcal{P}^*_f(K(D)), \sqsubseteq^{\flat}\right)}, \subseteq\right) \\ \mathrm{CONVEX}(D) & = & \left(\overline{\left(\mathcal{P}^*_f(K(D)), \sqsubseteq^{\natural}\right)}, \subseteq\right) \end{array}$$

are algebraic domains and given a continuous domain $(D, \sqsubseteq)$ with basis B, the ideal completions

$$\begin{array}{rcl} \mathrm{UPPER}(D) & = & \left(\overline{\left(\mathcal{P}^*_f(B), \ll^{\#}\right)}, \subseteq\right) \\ \mathrm{LOWER}(D) & = & \left(\overline{\left(\mathcal{P}^*_f(B), \ll^{\flat}\right)}, \subseteq\right) \\ \mathrm{CONVEX}(D) & = & \left(\overline{\left(\mathcal{P}^*_f(B), \ll^{\natural}\right)}, \subseteq\right) \end{array}$$

are continuous domains (independent of the basis chosen). These domains are known as the *upper* (or *Smyth*), the *lower* (or *Hoare*) and the *convex* (or *Plotkin* or *Vietoris*) *powerdomains* of D respectively. A powerdomain is the domain-theoretic analog to the powerset. They were introduced as a tool for modeling the semantics of non-deterministic programs [29]. We will be using the lower powerdomain in chapter 13 together with the following theorem [1].

Theorem 4 *The lower powerdomain of a continuous domain* $(D, \sqsubseteq)$ *is isomorphic to the lattice of all non-empty Scott closed subsets of* D.

$$\bigsqcup X = \mathsf{closure}\left(\bigcup X\right)$$

2.9 Recursion Theory

Recursion theory attempts to formalize the notion of an algorithm [66]. Informally, an algorithm is a partial function f from a set of inputs X to a set of outputs Y together with a finite set of instructions for evaluating $f(x) \in Y$ for each $x \in X$. Such a partial function $f : X \rightsquigarrow Y$ is said to be *effectively computable* and the set of instructions is said to be an *effective procedure* for computing this partial function. In recursion theory, an algorithm, such as this, is broken in three smaller algorithms;

1. an algorithm from the set of inputs X to the natural numbers $\mathbb{N}$,
2. an algorithm from the natural numbers $\mathbb{N}$ to the natural numbers $\mathbb{N}$ and
3. an algorithm from the natural numbers $\mathbb{N}$ to the set of outputs Y.

2.9.1 Partial Recursive Functions

Recursion theory provides a means for formalizing algorithms from the natural numbers $\mathbb{N}$ to the natural numbers $\mathbb{N}$. Firstly, a function $f : \mathbb{N}^n \times \mathbb{N} \rightsquigarrow \mathbb{N}$ is said to be defined by *primitive recursion* from $g : \mathbb{N}^n \rightsquigarrow \mathbb{N}$ and $h : \mathbb{N}^n \times \mathbb{N} \times \mathbb{N} \rightsquigarrow \mathbb{N}$ if

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}) \\ f(\vec{x}, y+1) &= h(\vec{x}, y, f(\vec{x}, y)) \end{aligned}$$

where $\vec{x} : \mathbb{N}^n$, $n \in \mathbb{N}$ and $y : \mathbb{N}$. Secondly, a function $f : \mathbb{N}^n \to \mathbb{N}$ is said to be defined by *restricted μ-recursion* from $g : \mathbb{N}^n \times \mathbb{N} \rightsquigarrow \mathbb{N}$ if

$$\begin{aligned} \forall \vec{x} \cdot \exists y \cdot g(\vec{x}, y) &= 0 \\ f(\vec{x}) &= \mu y . g(\vec{x}, y) = 0 \end{aligned}$$

where $\vec{x} : \mathbb{N}^n$, $n \in \mathbb{N}$, $y : \mathbb{N}$ and $\mu y.g(\vec{x}, y) = 0$ is defined as the least natural number y such that $g(\vec{x}, y) = 0$. Thirdly, let us define the *initial functions* as the zero function $n \mapsto 0$, the successor function $n \mapsto n + 1$ and the projection functions $(x_1, x_2, \ldots, x_n) \mapsto x_i$ for $1 \leq i \leq n$. Finally, using these definitions, the class of *recursive functions* is defined as the smallest class of functions

- containing the initial functions and
- closed under composition, primitive recursion and restricted μ-recursion.

An important part of recursion theory is Church's Thesis, which claims that every effectively computable function is a recursive function. Clearly, every recursive function is an effectively computable function. However, there is a problem with the definition of a recursive function. Given a recursive function $g(\vec{x}, y)$, there is no algorithm for determining whether $\forall \vec{x} \cdot \exists y \cdot g(\vec{x}, y) = 0$ is true, as required in the definition of μ-recursion. Thankfully, this problem goes away if we consider partial functions instead of total functions. A function f is said to be defined by *unrestricted μ-recursion* if

$$f(\vec{x}) = \mu y. (\forall z < y \cdot g(\vec{x}, z) \neq \bot) \wedge g(\vec{x}, y) = 0.$$

The class of *partial recursive functions* is the smallest class of functions

- containing the initial functions and
- closed under composition, primitive recursion and unrestricted μ-recursion.

A subset of $\mathbb{N}$ is said to be *recursively enumerable* (abbreviated r.e.) if it is the domain of a partial recursive function, or equivalently, if it is the range of a partial recursive function. In fact, an infinite r.e. set is the range of a one-one recursive function. A subset of $\mathbb{N}$ is said to be *recursive* if it is empty or it is the range of a non-decreasing recursive function. In fact, an infinite recursive set is the range of an increasing recursive function [66].

The partial recursive functions are synonymous with the algorithms from the natural numbers to the natural numbers. However, the original informal notion of an algorithm concerned partial functions with more general domains and codomains. This gap is bridged by considering some fixed one-one function from the domain or codomain to the natural numbers. Such a mapping is called a *coding*. The coding is chosen so that it is itself given by an informal algorithm and it is invertible.

2.9.2 Partial Recursive Functionals

In this section, we consider the generalization of recursiveness by considering effective procedures that act not on numbers, but on functions as well. If $\vec{f}$ is a set of partial functions, the class of functions *partially recursive in* $\vec{f}$ is the smallest class of functions

- containing the initial functions and $\vec{f}$ and
- closed under composition, primitive recursion and unrestricted μ-recursion.

The functional $F : (\mathbb{N} \rightsquigarrow \mathbb{N}) \times \mathbb{N} \rightsquigarrow \mathbb{N}$ is a *partial recursive functional* if it can be obtained from the initial functions and the functions $f : \mathbb{N} \rightsquigarrow \mathbb{N}$ by composition, primitive recursion and unrestricted μ-recursion. It follows that $\lambda x.F(f, x)$ is a function partially recursive in the functions f. However, the reverse is not true.

Recall that a partial function $f : \mathbb{N} \rightsquigarrow \mathbb{N}$ can be interpreted as the set $f = \{(x, f(x)) | x \in \mathbb{N} \wedge f(x) \text{ is defined}\}$. A partial functional $F(f, x) : (\mathbb{N} \rightsquigarrow \mathbb{N}) \times \mathbb{N} \rightsquigarrow \mathbb{N}$ is said to be *compact* if

$$\forall f, x \cdot \exists \text{ finite } g \subseteq f \cdot F(f, x) = F(g, x)$$

and it is said to be *monotone* if

$$\forall f, x \cdot \forall g \supseteq f \cdot F(f, x) = F(g, x).$$

It can be shown that if $F(f, x) : (\mathbb{N} \rightsquigarrow \mathbb{N}) \times \mathbb{N} \rightsquigarrow \mathbb{N}$ is a partial recursive functional then it is compact and monotone.

The *First Recursion Theorem* states that every partial recursive functional $F : (\mathbb{N} \rightsquigarrow \mathbb{N}) \rightsquigarrow (\mathbb{N} \rightsquigarrow \mathbb{N})$ admits a least fixed-point, which is a partial recursive function, given by

$$\bigcup_{n=0}^{\infty} F^n(n \mapsto \perp).$$

A generalization of Church's Thesis claims that every effectively computable functional is a recursive functional. This means that effectively computable functionals can be applied to non-computable functions (sometimes referred to as an oracle).

The set of partial functions $\mathbb{N} \rightsquigarrow \mathbb{N}$ can be viewed as a topological space with basic open sets

$$\hat{f} = \{g : f \subseteq g\}$$

where f is a finite partial function. In other words, the set of finite partial functions form a countable basis for the set of partial functions. It can be shown that if O is an open subset of $\mathbb{N} \rightsquigarrow \mathbb{N}$ then

- $f \in O$ implies $\exists$ finite $g \in O$ such that $g \subseteq f$
- $f \in O$ and $f \subseteq g$ implies $g \in O$.

We call O *effectively open* if the set of finite partial functions belonging to it is r.e. This requires a coding for the finite partial functions (i.e. a function from the finite partial functions to the natural numbers).

An equivalent way of looking at the set of partial functions $\mathbb{N} \rightsquigarrow \mathbb{N}$ is by considering it to be a partially ordered set, under the ordering relation $\subseteq$. In this case $\mathbb{N} \rightsquigarrow \mathbb{N}$ is a directed complete partial order, in the sense that every directed set has a least upper bound (which is just the union of the set). Note that $g : \mathbb{N} \rightsquigarrow \mathbb{N}$ is compact iff it is finite and it is Scott continuous iff it is monotone and continuous.

It can be shown that a partial functional $F : (\mathbb{N} \rightsquigarrow \mathbb{N}) \rightsquigarrow (\mathbb{N} \rightsquigarrow \mathbb{N})$ is continuous iff it is compact and monotone. This means that the behavior of a continuous partial functional is completely determined by its behavior on finite partial functions. We call a partial functional effectively continuous if its behavior on finite partial functions, suitably coded, is partial recursive. It can be shown that the partial recursive functionals are effectively continuous, but not vice versa. It can be shown that the topological and domain theoretic notions of continuity with respect to the partial functionals $(\mathbb{N} \rightsquigarrow \mathbb{N}) \rightsquigarrow (\mathbb{N} \rightsquigarrow \mathbb{N})$ coincide. Also, if a partial functional $F : (\mathbb{N} \rightsquigarrow \mathbb{N}) \rightsquigarrow (\mathbb{N} \rightsquigarrow \mathbb{N})$ is effectively continuous then its least fixed point is partial recursive.

Chapter 3

The Real World

This small chapter introduces the real numbers and the closely related concepts that will be used in the later chapters.

3.1 The Real Line and the Complex Plane

There are two standard methods for constructing the *set of real numbers* $\mathbb{R}$, also known as the *real line*, from the set of rational numbers $\mathbb{Q}$. In one method, the real line is defined as the set of *Dedekind cuts* endowed with addition $+$, multiplication $\times$ and order $<$. A Dedekind cut is a subset of the set of rational numbers satisfying certain conditions. The essential condition being that each member of a cut must be greater than all rational numbers that are not in the cut. The Dedekind method results in a very elegant mathematical structure and is of historical importance as being the first satisfactory definition of a real number. In the other method, one considers the set of all sequences of rational numbers $\langle q_n \rangle_{n=0}^{\infty}$ that satisfy the *Cauchy condition*. The Cauchy condition states that for any rational number $\epsilon > 0$ there is an integer N such that $|q_n - q_m| < \epsilon$ for all integers n and m exceeding N. Two sequences are said to be equivalent if their difference converges to zero. The resulting collection of equivalence classes endowed with addition, multiplication and order is the real line. The appeal of this definition stems from the fact that it can be generalized easily to other mathematical settings. We classify the *basic arithmetic operations* as negation, reciprocal, addition and multiplication.

It can be shown that the real line defined by Dedekind and Cauchy satisfy the set of axioms for a complete ordered field. In fact, these axioms completely characterize the real line because there is only one complete ordered field up to isomorphism. The real line together with the *Euclidean metric* $d_{\mathrm{E}}(x, y) = |x - y|$ forms a metric space. In general, the Euclidean metric on $\mathbb{R}^n$ is given by

$$d_{\mathrm{E}} : \mathbb{R}^n \times \mathbb{R}^n \quad \rightarrow \quad \mathbb{R}$$

$$d_{\mathrm{E}}\left(\left(x_1, x_2, \ldots, x_n\right), \left(y_1, y_2, \ldots, y_n\right)\right) = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2}.$$

The real line together with the set of open sets induced by the Euclidean metric forms a topological space. Note that only the bounded closed intervals in $\mathbb{R}$ are compact sets. In particular, $\mathbb{R}$ itself is not compact.

Classically, there are three ways to define an integer "close" to a given real number $x \in \mathbb{R}$; namely the *floor* $\lfloor x \rfloor$, *ceiling* $\lceil x \rceil$ and *round* $\lfloor x \rceil$ satisfying

$$\begin{aligned} 0 &\leq x - \lfloor x \rfloor < 1 && (3.1)\\ -1 &< x - \lceil x \rceil \leq 0 && (3.2)\\ -\frac{1}{2} &< x - \lfloor x \rceil \leq \frac{1}{2}. && (3.3) \end{aligned}$$

It is convenient to extend the system of real numbers with infinity ∞ so that division by zero can be handled. This enlarged set is called the *set of extended real numbers* or the *extended real line*

$$\mathbb{R}^{\infty} = \mathbb{R} \cup \{\infty\}.$$

In general, $\mathbb{F}^{\infty}$ will be used to denote $\mathbb{F} \cup \{\infty\}$. The *one-point compactification* of the real line is the topological space

$$\left(\mathbb{R}^{\infty}, \mathsf{open}\left(\mathbb{R}\right) \cup \mathsf{complement}\left(\mathsf{closed}\left(\mathbb{R}\right)\right)\right)$$

where $\mathsf{open}\left(\mathbb{R}\right)$ is the set of open sets in $\mathbb{R}$, $\mathsf{closed}\left(\mathbb{R}\right)$ is the set of closed sets in $\mathbb{R}$ and $\mathsf{complement}\left(A\right)$ is the set of complements of the elements in A. We do not consider the two-point compactification $\mathbb{R} \cup \{\pm\infty\}$ of the real line because it does not behave as elegantly with the various representations of the real numbers that we will be presenting in the rest of this thesis. The definition of order, addition and multiplication are extended in the obvious manner by defining

$$\begin{aligned} x + \infty &= \infty\\ \infty + x &= \infty \end{aligned}$$

for all real numbers x and by defining

$$\begin{aligned} y \times \infty &= \infty\\ \infty \times y &= \infty\\ \infty \times \infty &= \infty \end{aligned}$$

for all non-zero real numbers y. Note that order, addition and multiplication are partial functions on the set of extended real numbers because

$$\begin{array}{lcl} x & < & \infty \\ \infty & < & x \\ \infty & < & \infty \\ \infty & + & \infty \\ 0 & \times & \infty \\ \infty & \times & 0 \end{array}$$

are undefined for all real numbers x. This essentially introduces the concept of an "undefined number" or "not a number" denoted by NaN in the floating point community. The extended real line $\mathbb{R}^{\infty}$ is usually represented by the stereographic projection $\sigma : \mathbb{R}^{\infty} \longrightarrow \mathbb{R}^2$

$$\sigma(x) = \left(\frac{2x}{x^2+1}, \frac{x^2-1}{x^2+1} \right)$$

of the extended real line $\mathbb{R}^{\infty}$ onto the unit circle in Euclidean space $\mathbb{R}^2$ as illustrated in figure 3.1. The usual chordal metric $d_{\mathrm{C}} : \mathbb{R}^{\infty} \times \mathbb{R}^{\infty} \longrightarrow [0,2]$ on the extended real line is the chordal distance given by

$$d_{\mathrm{C}}(x,y) = d_{\mathrm{E}}(\sigma(x), \sigma(y)) = \frac{2|x-y|}{\sqrt{1+x^2}\sqrt{1+y^2}} \tag{3.4}$$

as illustrated in figure 3.2. An alternative metric on the extended real line $d_{\mathrm{P}} : \mathbb{R}^{\infty} \times \mathbb{R}^{\infty} \longrightarrow [0,2]$ is given by

$$d_{\mathrm{P}}(x,y) = \frac{2|x-y|}{(1+|x|)(1+|y|)} \tag{3.5}$$

as illustrated in figure 3.3.

Proposition 5 *The two metrics $d_{\mathrm{C}}(x,y)$ and $d_{\mathrm{P}}(x,y)$ are topologically equivalent.*

Proof:

$$\tfrac{1}{2} d_{\mathrm{P}}(x,y) \le d_{\mathrm{C}}(x,y) \le 2 d_{\mathrm{P}}(x,y) \ \blacksquare$$

A closed interval in $\mathbb{R}^{\infty}$ is either the set $\mathbb{R}^{\infty}$ itself, a singleton set $\{a\}$ for some $a \in \mathbb{R}^{\infty}$ or a set $[a,b]$ represented by the arc from $\sigma(a)$ to $\sigma(b)$ anti-clockwise in figure 3.1 for some $a \neq b$. Let $\underline{[a,b]}$ denote a and $\overline{[a,b]}$ denote b.

Let $\mathbb{I}[\mathbb{F}]$, $\mathbb{I}^{\mathrm{O}}[\mathbb{F}]$, $\mathbb{I}^{\mathrm{C}}[\mathbb{F}]$ and $\mathbb{I}^{\mathrm{F}}[\mathbb{F}]$ denote the set of all, open, closed and compact intervals in the extended real numbers $\mathbb{R}^{\infty}$ with end-points restricted to a subset $\mathbb{F}$

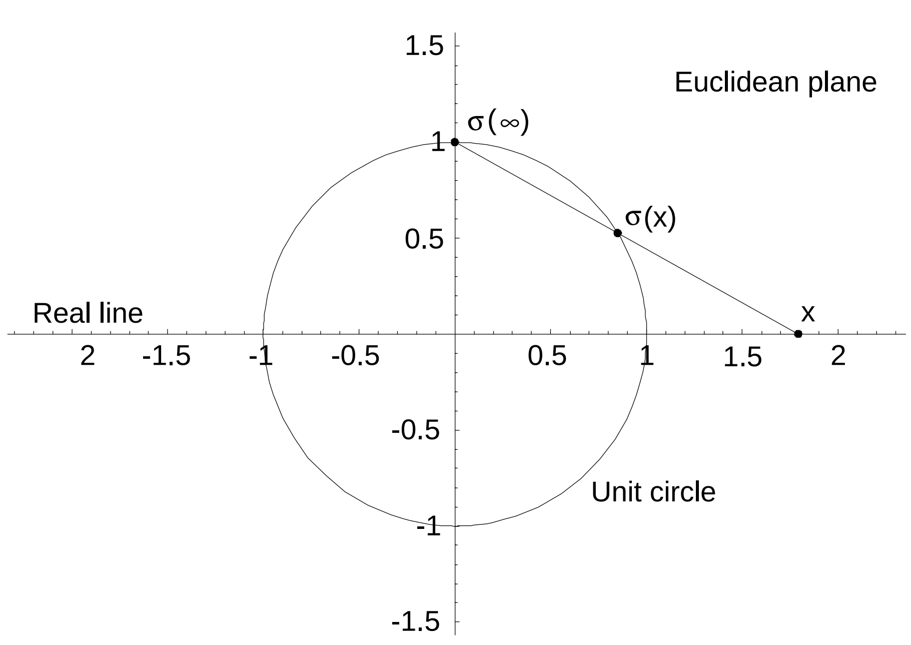


Figure 3.1: Stereographic projection σ of the extended real line $\mathbb{R}^{\infty}$ onto the unit circle in Euclidean space $\mathbb{R}^{2}$.

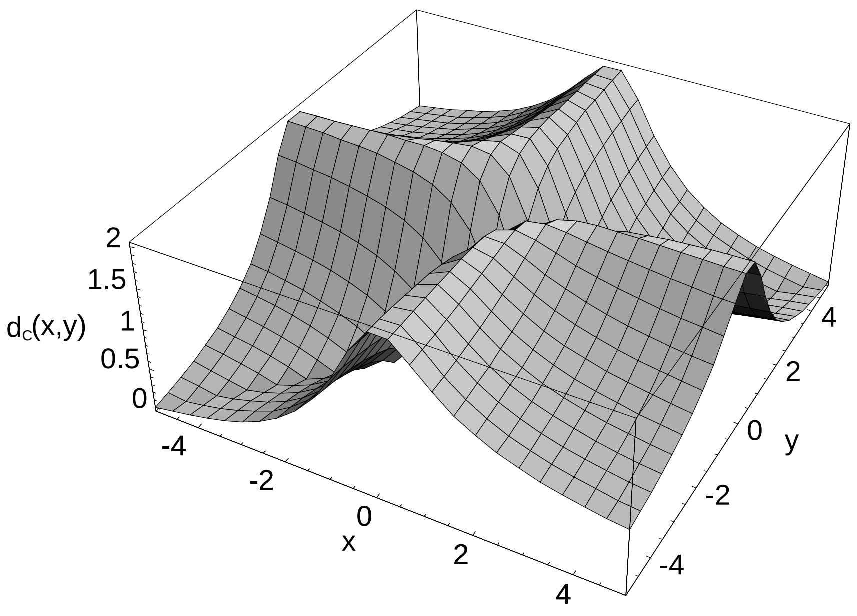


Figure 3.2: The chordal metric $d_{\mathrm{C}}(x, y)$ on the extended real line.

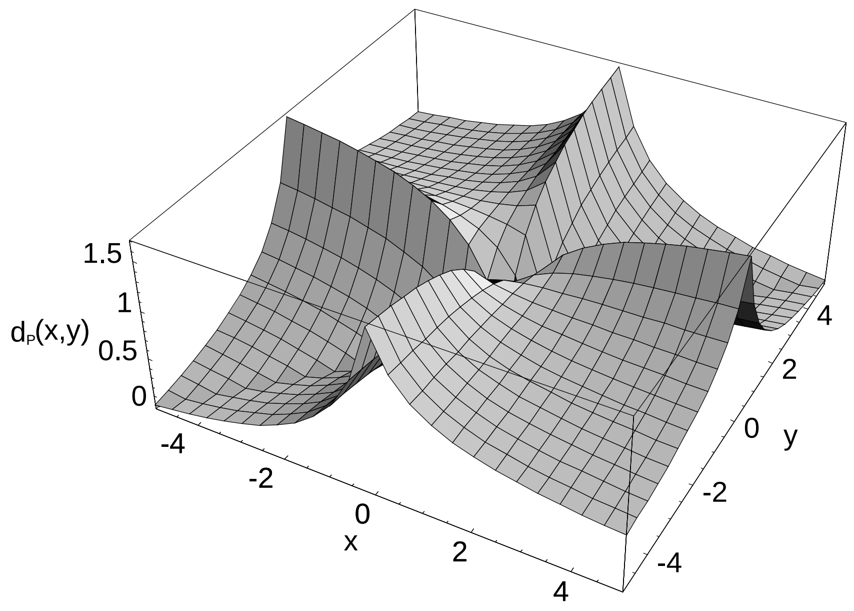


Figure 3.3: The metric $d_{\mathrm{P}}\left(x, y\right)$ on the extended real line.

of $\mathbb{R}$. The extended real line is also known as the *one-point compactification* of the real line because the set $\mathbb{R}^{\infty}$ itself is compact whereas $\mathbb{R}$ is not. *Interval arithmetic* on $\mathbb{R}^{\infty}$ is defined as the canonical extension of the basic arithmetic operations on $\mathbb{R}^{\infty}$.

A complex number z is specified by a pair of real numbers (x, y); written $z = x + iy$, where i is a fixed symbol. The *set of complex numbers* or the *complex plane* is endowed with addition $+$ and multiplication $\times$ in such a way that $i^2 = -1$. The complex plane is isomorphic to $\mathbb{R}^2$. Therefore, the complex plane together with the Euclidean metric forms a metric space. The complex plane together with the set of opens sets induced by the Euclidean metric forms a topological space. Note that only the bounded closed intervals in $\mathbb{C}$ are compact sets. In particular, $\mathbb{C}$ itself is not compact. It is convenient to extend the system of complex numbers with infinity ∞. This enlarged set is called the *set of extended complex numbers* or the *extended complex plane* $\mathbb{C}^{\infty}$. The extended complex plane $\mathbb{C}^{\infty}$ is usually represented by the stereographic projection $\sigma : \mathbb{C}^{\infty} \longrightarrow \mathbb{R}^3$

$$\sigma\left(x + iy\right) = \left(\frac{2x}{x^2 + y^2 + 1}, \frac{2y}{x^2 + y^2 + 1}, \frac{x^2 + y^2 - 1}{x^2 + y^2 + 1}\right)$$

of the extended complex plane $\mathbb{C}^{\infty}$ onto the Riemann sphere in Euclidean space $\mathbb{R}^3$ as illustrated in figure 3.4 [50]. The usual chordal metric $d_{\mathrm{C}} : \mathbb{C}^{\infty} \times \mathbb{C}^{\infty} \longrightarrow [0, 2]$

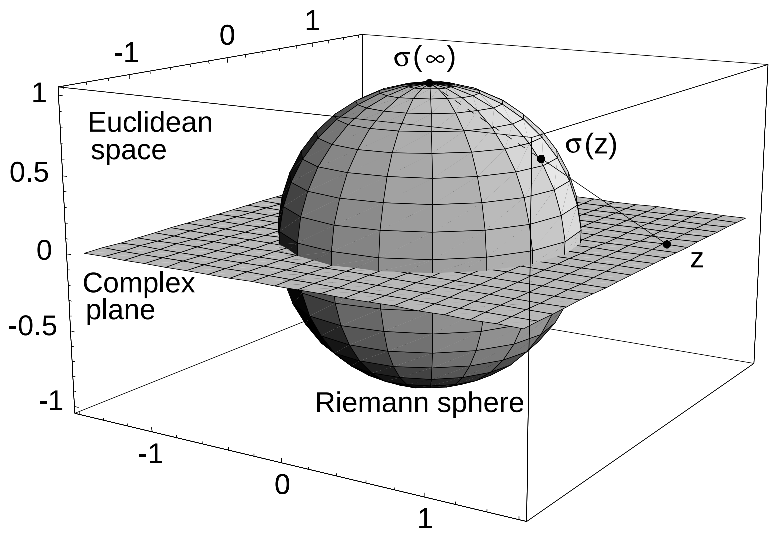


Figure 3.4: Stereographic projection $\sigma(z)$ of the extended complex plane $\mathbb{C}^\infty$ onto the Riemann sphere in Euclidean space $\mathbb{R}^3$.

on the extended complex plane is the chordal distance given by

$$\begin{aligned} d_{\mathrm{C}}(a+ib, c+id) &= d_{\mathrm{E}}(\sigma(a+ib), \sigma(c+id)) \\ &= \frac{2\sqrt{(a-c)^2+(b-d)^2}}{\sqrt{1+a^2+b^2}\sqrt{1+c^2+d^2}}. \end{aligned} \tag{3.6}$$

3.2 Complex Functions and Power Series

In this section, we examine various classes of complex functions believed to interesting and relevant with respect to the representations discussed later in this thesis, although no direct reference is made directly to them.

A function which is differentiable at every point of an open set O is said to be *holomorphic* in O [64]. The set of functions holomorphic in O is denoted by $H(O)$. A function is holomorphic in an open set O if and only if it is *analytic*, that is, locally representable by a power series. Let us define the open disc with centre $c \in \mathbb{C}$ and radius $0 < r \leq \infty$ to be

$$D(c; r) = \{z \in \mathbb{C} : |z - c| < r\}.$$

Theorem 6 (Taylor's Theorem) *Let* $f \in H(D(c; R))$. *Then there exists* *unique complex numbers a_n such that*

$$f(z) = \sum_{n=0}^{\infty} a_n (z-c)^n$$

for all $z \in D(c;R)$. The complex numbers a_n are given by

$$a_n = \frac{f^{(n)}(c)}{n!}.$$

The series defined in the theorem is called the *Taylor series* for f at c. In practice, the Taylor series of a function can be used to approximate the function by taking longer and longer polynomials. However, this approach has undesirable limitations. Often, the Taylor series of a function only converges in a small region of the complex plane. This problem can be circumvented by considering *rational functions*, that is, quotients of polynomials. However, rational functions often fail to be holomorphic at isolated points. So, let us define the open annulus with centre c and radii $0 \leq r < s \leq \infty$ to be

$$A(c;r,s) = \{z \in \mathbb{C} : r < |z-c| < s\}.$$

Theorem 7 (Laurent's Theorem) *Let $f \in H(A(c;R,S))$. Then there exists unique complex numbers a_n such that*

$$f(z) = \sum_{n=-\infty}^{\infty} a_n (z-c)^n$$

for all $z \in A(c;R,S)$.

A function f is said to have a *pole* of order $m \geq 1$ at c if

$$f(z) = \sum_{n=-m}^{\infty} a_n (z-c)^n$$

for some $0 < r \leq \infty$ and for all $z \in A(c;0,r)$ such that $a_{-m} \neq 0$. A function f is said to have a *zero* of order $m \geq 1$ at c if

$$f(z) = \sum_{n=m}^{\infty} a_n (z-c)^n$$

for some $0 < r \leq \infty$ and for all $z \in A(c;0,r)$ such that $a_m \neq 0$.

The above ideas can be extended to the extended complex plane by replacing the Euclidean metric by the chordal metric; namely redefining the open disc by

$$D(c;r) = \{z \in \mathbb{C}^{\infty} : d_{\mathrm{C}}(z,c) < r\}$$

and the open annulus by

$$A(c;r,s) = \{z \in \mathbb{C}^{\infty} : r < d_{\mathrm{C}}(z,c) < s\}.$$

A function f is said to be differentiable, have a pole or have a zero at ∞ if $f\left(\frac{1}{z}\right)$ is differentiable, has a pole or has a zero at 0 respectively. A function which is *holomorphic* in O except possibly for poles is said to be *meromorphic* [64].

Theorem 8

- *If f is holomorphic in $\mathbb{C}^{\infty}$ then f is constant.*
- *If f is meromorphic in $\mathbb{C}^{\infty}$ then f is a rational function, where a rational function is a polynomial fractional transformation.*

Suppose that we have a function that is meromorphic on a collection of open sets that cover $\mathbb{C}^{\infty}$ then there is a finite subcollection that cover $\mathbb{C}^{\infty}$. This means that a power series can be derived for each open set in the finite subcollection with respect to different points. These power series coincide on the intersections. This is known as analytic continuation and we shall be using this property in order to extend algorithms based on sequences of rational functions for the elementary functions over the extend real line.

Chapter 4

Approximate Digital Representations

Numbers are stored in the memory of digital computers as a sequence of binary digits. The most common number system used to represent integers is the *two's complement system.* Many engineering and scientific applications require a larger range for numbers which are represented in a *floating point* format. This is the binary equivalent of scientific notation using a *mantissa* and an *exponent* to represent a number. In this chapter, we discuss the various digital formats that are widely used to represent the set of real numbers in the world of computing today.

4.1 The Fixed Point Representation

A natural number $N \in \mathbb{N}$ in base or radix r is written in *positional notation* as

$$N_r = (d_{m-1}d_{m-2}\cdots d_0)_r \tag{4.1}$$

where each digit d_i has one of the distinct values $\mathbb{N}(r) = \{0, 1, 2, \ldots, r-1\}$ and m is the number of digits required to represent the number. Thus the natural number 741 could be written as 741_{10} with $r = 10$, $m = 3$, $d_0 = 1$, $d_1 = 4$ and $d_2 = 7$ in equation (4.1). The position of the digit relative to the rightmost digit represents a power of the base r; that is, the value of the number is calculated as

$$N = \sum_{i=0}^{m-1} d_i r^i \tag{4.2}$$

in which the digits are restricted in value such that $d_i \in \mathbb{N}(r)$. The arithmetic operations in equation (4.2) could be carried out in any number base and this equation is frequently used to determine the decimal equivalent of a number in another base. The decimal system is, of course, used primarily for ordinary arithmetic by human beings, and the binary system ($r = 2$) is used for computer arithmetic. Octal ($r = 8$) and hexadecimal ($r = 16$) representations are convenient for writing long binary numbers. Note that the largest m-digit positive integer in positional notation

$$N_r = ((r-1)(r-1)\cdots(r-1))_r$$

has the value

$$N = (r-1)\sum_{i=0}^{m-1} r^i = r^m - 1.$$

The positional representation defined by equation (4.1) is valid for natural numbers only. If a fraction is to be represented, a radix point is used to separate the integer part from the fractional part of the number. The radix point is called the *binary point* in base 2 and the *decimal point* in base 10. In general, an m-digit positive fraction is written with a leading radix point as

$$.n_r = .\,(d_{-1}d_{-2}\cdots d_{-m})_r$$

with the value

$$.n = \sum_{i=1}^{m} d_{-i}r^{-i}$$

where the negative subscript for the digits indicates the appropriate negative power of r.

Internally, a digital computer performs arithmetic on integers without taking into account the position of the radix point. In other words, the programmer is responsible for keeping track of the radix point. This is called a *fixed point* representation.

In order to represent the complete set of integers, a notation for negative values is necessary. In ordinary arithmetic, a negative number is represented by prefixing the magnitude (or absolute value) of the number with a minus sign. For hand calculations, the use of a separate symbols to indicate positive (+) and negative (−) numbers is convenient. The computer circuits that manipulate positive and negative integers are also simplified if one of the digits in the positional notation of a number is used to indicate the sign of the integer. Two such possible representations of signed integers are the *sign magnitude* notation and the *complement* notation. In both notations, the most significant digit on the left in the positional form of the number indicates the sign.

The sign magnitude representation of a number in positional notation has the form

$$N_r = (d_{m-1}d_{m-2}\cdots d_1d_0)_r$$

where the sign of the number is indicated by the most significant (leftmost) digit

$$d_{m-1} = \begin{cases} 0 & \text{if } N_r \geq 0 \\ r-1 & \text{if } N_r < 0. \end{cases}$$

The magnitude of the number, written $|N_r|$, is

$$|N_r| = \sum_{i=0}^{m-2} d_i r^i \tag{4.3}$$

where only the first $m-1$ digits from the right are considered. According to the definitions and equation (4.3), the four-digit number $0101_2 = 5$ and $1101_2 = -5$. The number of digits, including the sign digit, in the representation must be specified or confusion could result. For example, 1101_2 in an eight-digit representation is assumed to be 00001101_2, which has the decimal value 13.

Most microprocessors have arithmetic instructions that operate on negative numbers represented in a complement number system. Complement representations have an advantage over sign magnitude notation because the sign digit does not have to be treated in a special way during addition and subtraction. This simplifies the arithmetic circuits. In these systems, positive numbers have the same representation as in sign magnitude notation, but the negative numbers are formed by computing the complement of the number according to the rules of the specific system being used. The two most common complement systems are the radix complement and the diminished radix complement systems.

In the general form, the radix complement of an m-digit number is computed mathematically as

$$N'_r = r^m - N_r \tag{4.4}$$

where N'_r is the radix complement of the base r number N_r. In a digital computer, only m-digit values can be represented. If any operation produces a result that requires more than m digits, the higher order digits are ignored. The two most commonly used radix complement systems are the two's complement and the ten's complement systems. If the number and its complement are added, the result is 0 to m digits, as expected. In a two's complement system, negative values always have a leading 1 and positive values have 0 as the leading digit.

The diminished radix complement is computed as

$$\overline{N_r} = r^m - N_r - 1 \tag{4.5}$$

which is one less than the radix complement value computed by equation (4.4).

4.2 The Floating Point Representation

The representation for numbers that we have considered above assume that the radix point is located in a fixed position, yielding an integer or a fraction as the interpretation of the internal machine representation. Of course, the radix point is not actually stored with the number, but its position must be remembered by the programmer. This method is called fixed point.

In practice, the machine value is limited to a finite range which is determined by the number of bits used in the representation. For a 32-bit word, the range of signed fixed point integers is about $+2^{31}$ or $+10^{11}$. Thus the limited range of fixed point notation is a drawback for certain applications.

To overcome many of the limitations of the fixed point notation, a representation that is the counterpart of the scientific notation is used for numbers in digital computers. The floating point notation represents a number as a fractional part times a selected base raised to a power. In the machine representation, only the fractional part and the value of the exponent are stored. The decimal equivalent is written as

$$N.n = f \times 10^e$$

where f is the fraction or *mantissa* and e is an integer called the *exponent*. The choice for the base is usually base 2, although base 16 is sometimes used.

The typical floating point format stores the fraction and the exponent together in an m-bit representation. The choice for a fixed length floating point format is commonly 32 or 64 bits, referred to as single precision and double precision, respectively. Extended formats with $m > 64$ are occasionally used when a greater range of precision is required.

Once the length of the floating point representation is chosen, a number of choices for both the length and the format of the fraction and exponent are possible. Many floating point formats employ a sign magnitude representation for the fraction. The fraction is generally normalized to yield as many significant digits as possible. Thus, in a base r system, the most significant digit is in the leftmost position in the fraction. For non-zero numbers in the binary system, the leftmost digit will be a 1. As the arithmetic unit or the program shifts the digits in the fraction during arithmetic operations, the exponent is adjusted accordingly. When normalized as a base 2 value, the magnitude of the fraction is $0.5 \leq |f| < 1$ unless the number is zero. The number of digits reserved for the fraction represents a compromise between the precision of the fraction and the range of the exponent. A typical single precision format (32 bits) might contain an 8-bit exponent and a 23-bit fraction and 1-bit for the sign.

An exponent could be represented in two's complement or any other notation that allows signed values. A different alternative, which permits the exponent to be represented internally as a positive number only, is to add an offset value. This value is often called an *excess*.

The overwhelming majority of the computer industry has adopted the IEEE standard for floating point [37]. Ultimately though any floating point format can only represent a tiny subset of the rational numbers. Since physical measurements have only finite accuracy, calculated values that depend on measured data are inherently inexact. Consequently, floating point arithmetic has a strong physical justification. However the vast majority of real numbers can only be represented approximately leading to the unavoidable notion of a *round off error*. Furthermore, these errors can accumulate leading to totally incorrect results.

Interval analysis [48] provides a way to handle the insidious nature of round-off errors. In interval analysis, a pair of bounding floating point numbers is maintained, which is guaranteed to contain the real number or interval in question. However, this interval can get unjustifiably large and thereby convey very little information.

Chapter 5

Exact Digital Representations

We have seen earlier that a certain set of equivalence classes on Cauchy sequences completely characterize the set of real numbers. It might be deduced from this that Cauchy sequences might provide a way to represent real numbers exactly in a digital computer. However, this is not true because in a digital computer only finite portions of a Cauchy sequence may be examined in finite time and a finite portion of a Cauchy sequence says absolutely nothing about the real number being represented. In fact, it is necessary to have bounding information about the portion of the Cauchy sequence that has not been examined. In this thesis, we will only consider representations of a real number, x say, by infinite sequences of intervals $\langle I_n \rangle_{n=0}^{\infty}$ such that $\bigcap_{n=0}^{\infty} I_n = \{x\}$. Without loss of generality, we can assume that the sequence of intervals is nested. We shall call such a sequence a *representative* of x. A sequence of intervals $\langle I_n \rangle_{n=0}^{\infty}$ is *valid* if $\bigcap_{n=0}^{\infty} I_n$ is a singleton set. In order to avoid a circular argument, the end-points of the intervals must be restricted to a dense strict subset $\mathbb{F}$ of $\mathbb{R}$. Typically, the *end-points set* $\mathbb{F}$ is taken to be the set of rational numbers $\mathbb{Q}$, but other contenders include the set of algebraic numbers $\mathbb{A}$, the b-adic numbers $\mathbb{B}(b) = \left\{\frac{n}{b^m} : n \in \mathbb{Z} \wedge m \in \mathbb{N}\right\}$ for any real number b greater than 1 and a quadratic field $\mathbb{Q}\left(\sqrt{d}\right) = \left\{p + q\sqrt{d} : p, q \in \mathbb{Q}\right\}$ for some square-free integer d greater than one. A finite sequence of intervals $\langle I_n \rangle_{n=0}^{m}$ will be interpreted as the infinite sequence $\langle I_n \rangle_{n=0}^{\infty}$ with $I_n = I_m$ for all $n > m$.

The axioms for the real numbers state that every real number has a reciprocal except 0. However, given a sequence of intervals, it cannot necessarily be decided in finite time whether it is a representative of 0. This means that we cannot necessarily avoid taking the reciprocal of 0. Therefore, it is convenient to include infinity, written ∞, in any exact representation of the real numbers. In other words, we shall be considering exact representations for the extended real numbers.

Of course, an infinite sequence of intervals $\langle I_n \rangle_{n=0}^{\infty}$ corresponds to the function $I : \mathbb{N} \to \mathbb{I}[\mathbb{F}]$. Recall that $\mathbb{I}[\mathbb{F}]$ is the set of all intervals (i.e. open, closed or otherwise) in $\mathbb{R}^\infty$ with end points restricted to the dense subset $\mathbb{F}$ of $\mathbb{R}^\infty$. Therefore, an extended real number may be represented simply by a function $\mathbb{N} \to \mathbb{I}[\mathbb{F}]$. In practice, a rate of convergence

$$|I_n| \leq \frac{1}{\alpha^n}$$

for some $\alpha > 1$ is imposed on the sequences of intervals $\langle I_n \rangle_{n=0}^{\infty}$. This approach can be used for developing non-incremental representations of the real numbers in a digital computer. In other words, I_n can be evaluated without necessarily evaluating I_i for $0 \leq i < n$. In particular, Ménissier-Morain [47] has implemented exact real arithmetic using this idea. However, the thrust of this thesis is to investigate whether an efficient incremental representation can be found. In other words, the evaluation of I_{n+1} must efficiently use the results of evaluating I_n.

Let us define the function $\eta : (\mathbb{N} \to \mathbb{I}[\mathbb{F}]) \to \mathbb{IR}^\infty$ by

$$\eta(I) = \bigcap_{n=0}^{\infty} I_n.$$

5.1 Domain of Real Intervals

For any closed interval $I \in \mathbb{I}^{\mathrm{C}}\mathbb{R}^\infty$ of the extended real line, the set of closed intervals $\mathbb{I}^{\mathrm{C}} I$ ordered by reverse inclusion is a continuous domain $\left(\mathbb{I}^{\mathrm{C}} I, \supseteq\right)$. Note that

$$\begin{aligned} I \sqsubseteq J &= I \supseteq J \\ I \sqcup J &= I \cap J \\ I \ll J &\Leftrightarrow \mathsf{interior}(I) \supseteq J \end{aligned}$$

where

$$\mathsf{interior}([a,b]) = (a,b).$$

Definition 9 (Continuous real domain) *Define the* continuous real domain $\mathbb{C}(I)$ *on* $I \in \mathbb{I}^{\mathrm{C}}\mathbb{R}^\infty$ *by*

$$\mathbb{C}(I) = \left(\mathbb{I}^{\mathrm{C}} I, \supseteq\right).$$

A sequence of nested closed intervals representing an extended real number can be modeled by a chain in the continuous domain $\mathbb{C}(\mathbb{R}^\infty)$ [63]. The extended real numbers are represented by singleton sets on the rim of the cone, while the other intervals are represented by points on the surface of the cone. The bottom element is $\mathbb{R}^\infty$. For any dense subsets $\mathbb{F}$ of I, the set of closed intervals $\mathbb{I}^{\mathrm{C}}[\mathbb{F}]$ in I with end-points taken from $\mathbb{F}$ is a basis for the continuous domain $\mathbb{C}(I)$. For any dense

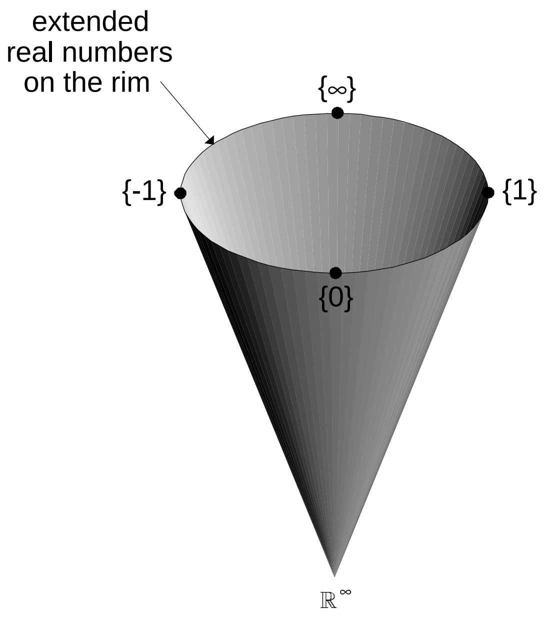


Figure 5.1: The continuous real domain $\left(\mathbb{I}^{C}\mathbb{R}^{\infty}, \supseteq\right)$.

subsets $\mathbb{F}$ of I, the ideal completion of the abstract basis $\left(\mathbb{I}^{C}\left[\mathbb{F}\right], \ll\right)$ is isomorphic to the continuous domain $\mathbb{C}\left(I\right)$. The maximal elements (or singleton sets) are identified with the elements of I. For a continuous function $f : \mathbb{C}\left(I\right) \to \mathbb{C}\left(I\right)$, if $\mathbb{I}^{C}\left[\mathbb{F}\right] \ni y \ll f(x)$ then there exists $\mathbb{I}^{C}\left[\mathbb{F}\right] \ni z \ll x$ such that $y \ll f(z)$.

For any closed interval $I \in \mathbb{I}^{C}\mathbb{R}^{\infty}$ of the extended real line and a dense subset $\mathbb{F}$ of I, the ideal completion $\left(\overline{(\mathbb{I}^{C}\left[\mathbb{F}\right], \supseteq)}, \subseteq\right)$ of $\mathbb{I}^{C}\left[\mathbb{F}\right]$ ordered by reverse inclusion is an algebraic domain.

Definition 10 (Algebraic real domain) *Define the* algebraic real domain $\mathbb{A}\left(\mathbb{F}\right)$ *on* $I \in \mathbb{I}^{C}\mathbb{R}^{\infty}$ *with respect to dense subset* $\mathbb{F}$ *of* I *by*

$$\mathbb{A}\left(\mathbb{F}\right) = \left(\overline{(\mathbb{I}^{C}\left[\mathbb{F}\right], \supseteq)}, \subseteq\right).$$

A sequence of nested closed intervals representing an extended real number can be modeled by a chain in the algebraic domain $\mathbb{A}\left(\mathbb{Q}^{\infty}\right)$ [63]. The bottom element is $\{\mathbb{R}^{\infty}\}$. The algebraic domain $\mathbb{A}\left(\mathbb{F}\right)$ is a Scott-domain. The compact elements of $\mathbb{A}\left(\mathbb{F}\right)$ are given by

$$K\left(\mathbb{A}\left(\mathbb{F}\right)\right) = \left\{\downarrow x \left| x \in \mathbb{I}^{C}\left(\mathbb{F}\right)\right.\right\}.$$

In other words, the ideal completion $\left(\overline{(\mathbb{I}^{C}\left[\mathbb{F}\right], \supseteq)}, \subseteq\right)$ of $(\mathbb{I}^{C}\left[\mathbb{F}\right], \supseteq)$ is an algebraic domain with basis isomorphic to $\mathbb{I}^{C}\left[\mathbb{F}\right]$. For any $d, e \in K\left(\mathbb{A}\left(\mathbb{R}^{\infty}\right)\right)$, the step

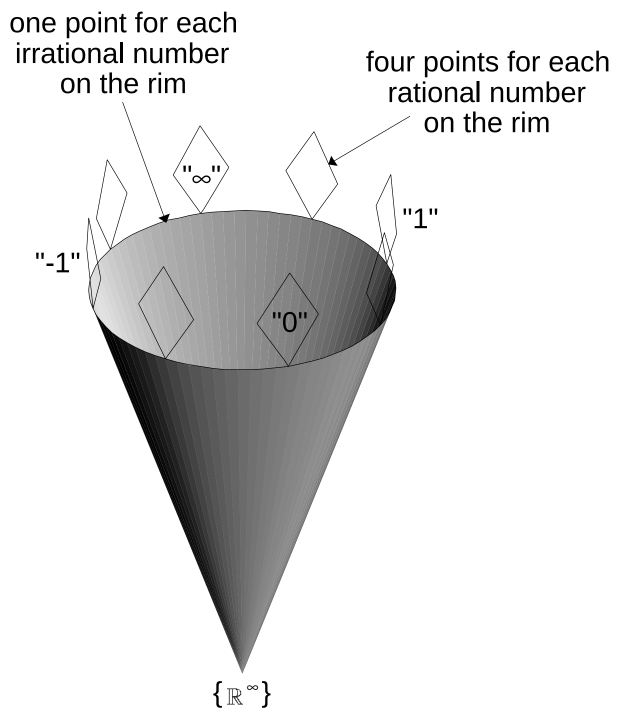

one point for each irrational number on the rim
four points for each rational number on the rim
"∞"
"1"
"-1"
"0"
$\{\mathbb{R}^\infty\}$

Figure 5.2: The algebraic real domain $\left(\overline{(\mathbb{I}^{\mathrm{C}}\left[\mathbb{Q}^\infty\right], \supseteq)}, \subseteq\right)$.

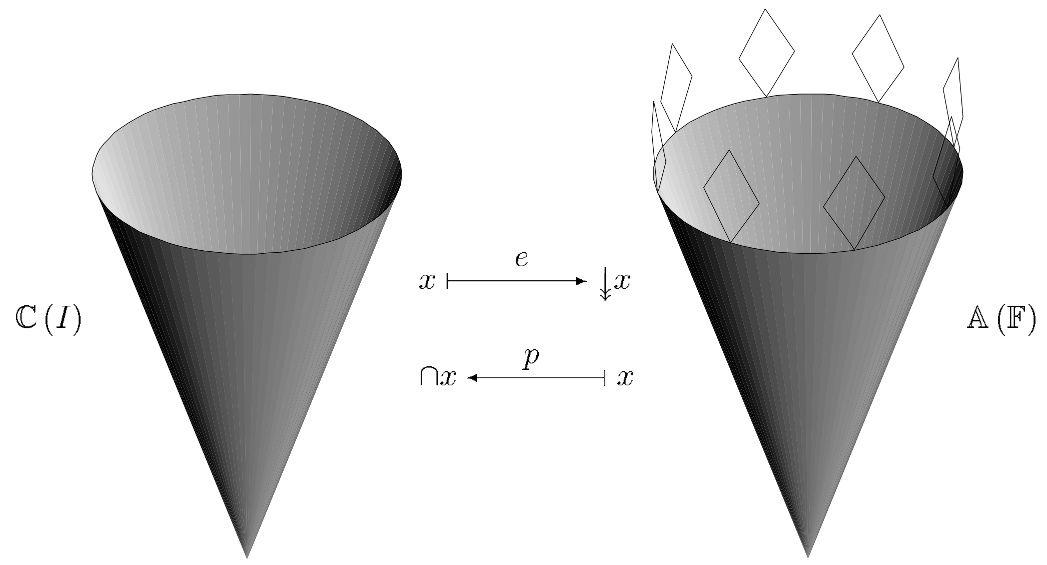
function $d \searrow e$ is compact and they are sufficient to generate a basis for the higher domain $[\mathbb{A}(\mathbb{R}^\infty) \to \mathbb{A}(\mathbb{R}^\infty)]$ [1]. The algebraic real domain has the advantage of allowing a distinction to be made between finitely and infinitely represented real numbers [11]. This is useful for efficiency reasons as we do not want to be forced to represent every number by an infinite product of matrices if it can be avoided. However, rational numbers cannot always be stored as a vector because ascertaining whether a real number is rational or irrational is undecidable. Given an interval $x \in \left(\mathbb{I}^{\mathrm{C}}[I], \supseteq\right)$, let us use the notation $\underline{x} = \mathsf{inf}(x)$, $\overline{x} = \mathsf{sup}(x)$ and

$$\begin{aligned}
\langle x\rangle &= \{y \in \mathbb{I}^{\mathrm{C}}[\mathbb{F}] \mid \underline{y} \leq \underline{x} \text{ and } \overline{x} \leq \overline{y}\} \\
\langle x\rangle\rangle &= \{y \in \mathbb{I}^{\mathrm{C}}[\mathbb{F}] \mid \underline{y} \leq \underline{x} \text{ and } \overline{x} < \overline{y}\} \\
\langle\langle x\rangle &= \{y \in \mathbb{I}^{\mathrm{C}}[\mathbb{F}] \mid \underline{y} < \underline{x} \text{ and } \overline{x} \leq \overline{y}\} \\
\langle\langle x\rangle\rangle &= \{y \in \mathbb{I}^{\mathrm{C}}[\mathbb{F}] \mid \underline{y} < \underline{x} \text{ and } \overline{x} < \overline{y}\}
\end{aligned}$$

where (perhaps counter intuitively) $\underline{I}<\underline{I}$ and $\overline{I} < \overline{I}$. Note that $\langle x\rangle \equiv{\downarrow}x$ and $\langle\langle x\rangle\rangle \equiv {\twoheaddownarrow}x$. In the algebraic domain, the elements $\langle x\rangle$, $\langle x\rangle\rangle$, $\langle\langle x\rangle$ and $\langle\langle x\rangle\rangle$ are distinct whenever $x \in \mathbb{I}^{\mathrm{C}}[\mathbb{F}]$, they provide only two distinct elements whenever $x \in \mathbb{I}^{\mathrm{C}} I$ with one end-point in $\mathbb{F}$ and they are indistinguishable whenever $x \in \mathbb{I}^{\mathrm{C}} I$ with neither end-point in $\mathbb{F}$.

$$\begin{tikzcd}
 & \langle x\rangle \arrow[dl, dash] \arrow[dr, dash] & \\
\langle x\rangle\rangle \arrow[dr, dash] & & \langle\langle x\rangle \arrow[dl, dash] \\
 & \langle\langle x\rangle\rangle &
\end{tikzcd}$$

Note that the functions $e : \mathbb{C}(I) \to \mathbb{A}(\mathbb{F})$ with $x \mapsto \downarrow x$ and $p : \mathbb{A}(\mathbb{F}) \to \mathbb{C}(I)$ with $x \mapsto \cap x$ form an *embedding/projection pair* with $p \circ e = \mathsf{Id}$ and $e \circ p \sqsubseteq \mathsf{Id}$.



For a continuous function $f : \mathbb{A}(\mathbb{F}) \to \mathbb{A}(\mathbb{F})$, if $y \sqsubseteq f(x)$ with y compact then $\exists z \sqsubseteq x$ such that $y \sqsubseteq f(z)$ with z compact.

5.2 Formal Digital Representations

An alternative indirect approach is to represent a real number by a sequence of digits that can be used to incrementally generate a sequence of intervals representing that real number. Ironically, although this approach is more elegant, it this has led to much less satisfactory implementations as testified by Boehm and Cartwright [5]. Nielsen and Kornerup [51] and Weihrauch [75] have examined the theoretical basis for such representations. A convenient definition for a formal digital representation is:

Definition 11 *The tuple* $(\mathbb{F}, \mathbb{D}, \theta)$ *is called a* ***digital representation*** *if*

- *the* ***end-points set*** $\mathbb{F}$ *is a subset of* $\mathbb{R}$*,*
- *the* ***digit set*** $\mathbb{D}$ *is a non-empty countable set of symbols and*
- *the* ***digit sequence map*** θ *is a partial functional*

$$(\mathbb{N} \rightsquigarrow \mathbb{D}) \rightsquigarrow (\mathbb{N} \rightsquigarrow \mathbb{I}(\mathbb{F}))\,.$$

So, a digital representation specifies a function for converting a sequence of digits into a sequence of intervals.

Definition 12 *The tuple* $(\mathbb{F}, \mathbb{D}, \theta, \Phi)$ *is called an* ***adequate digital representation*** *if*

- *the tuple* $(\mathbb{F}, \mathbb{D}, \theta)$ *is a* ***digital representation*** *and*
- *the* ***interval sequence map*** Φ *is a partial functional*

$$(\mathbb{N} \rightsquigarrow \mathbb{I}(\mathbb{F})) \rightsquigarrow (\mathbb{N} \rightsquigarrow \mathbb{D})$$

  satisfying

$$\bigcap_{n=0}^{\infty} I(n) = \{x\} \text{ for some } x \in \mathbb{R} \text{ implies } \bigcap_{n=0}^{\infty} \theta(\Phi(I))(n) = \{x\}.$$

So, an adequate digital representation specifies two functions; one for converting a sequence of digits into a sequence of intervals and the other for converting a sequence of intervals into a sequence of digits. A trivial example of an adequate digital representation is

$$(\mathbb{F}, \mathbb{I}[\mathbb{F}], I \mapsto I, I \mapsto I).$$

5.3 Decimal Expansion

The usual way to visualize real numbers is in terms of their decimal expansions; that is, one tends to think of them as entities such as 12.5 and $3.14159\cdots$. Of course, these can be viewed as infinite sequences $d_0 \cdot d_1 d_2 d_3 \cdots$ of natural numbers, where the first natural number is arbitrary and the other natural numbers lie between 0 and 9. In order to ensure that each real number has a unique representation, we exclude those sequences that terminate with nines. We could also assume that there are always infinitely many digits after the point by completing a finite sequence with 0's. Traditionally, the decimal expansion $d_0 \cdot d_1 d_2 d_3 \cdots$ for a real number, x say, is interpreted as the Cauchy sequence $\left\langle \sum_{i=0}^{n} \frac{d_i}{10^i} \right\rangle_{n=0}^{\infty}$. However, each member of this Cauchy sequence is actually endowed with an implicit piece of additional information; namely that

$$x \in \left[\sum_{i=0}^{n} \frac{d_i}{10^i}, \sum_{i=0}^{n} \frac{d_i}{10^i} + \frac{1}{10^n} \right)$$

for all $n \in \mathbb{N}$. The reason for this is that only real numbers in this interval can be represented by a decimal expansion starting with the digits $d_0 \cdot d_1 d_2 d_3 \cdots d_n$.

So, in fact, the unique decimal expansion $d_0 \cdot d_1 d_2 d_3 \cdots$ for a real number actually represents the sequence of intervals

$$\begin{array}{ll} & [d_0, d_0+1) \\ \supseteq & \left[d_0 + \frac{d_1}{10}, d_0 + \frac{d_1+1}{10}\right) \\ \supseteq & \left[d_0 + \frac{d_1}{10} + \frac{d_2}{100}, d_0 + \frac{d_1}{10} + \frac{d_2+1}{100}\right) \\ \supseteq & \cdots \end{array}$$

such that their intersection is the singleton set containing x. It can be shown that a real number is rational if and only if its decimal expansion is periodic.

Proposition 13 *The **decimal representation** for the non-zero real numbers corresponds to the adequate digital representation* $([0,\infty) \cap \mathbb{B}(10), \mathbb{N}, \theta, \Phi)$ *with*

$$\begin{aligned} \theta(d)(n) &= \left[\sum_{i=0}^{n} \frac{d(i)}{10^i}, \sum_{i=0}^{n} \frac{d(i)}{10^i} + \frac{1}{10^n}\right) \\ \Phi(I) &= \bigcap \left\{ d : \bigcap_{i=0}^{\infty} \theta(d)(i) \subseteq \bigcap_{j=0}^{\infty} I(j) \right\}. \end{aligned}$$

Proof:

$$\begin{aligned} \bigcap_{n=0}^{\infty} I(n) &= \{x\} \Rightarrow \\ \bigcap_{n=0}^{\infty} \theta(\Phi(I))(n) &= \bigcap_{n=0}^{\infty} \theta\left(\bigcap\left\{d : \left\{\sum_{i=0}^{\infty} \frac{d(i)}{10^i}\right\} \subseteq \{x\}\right\}\right)(n) \\ &= \bigcap_{n=0}^{\infty} \theta(d)(n) \text{ such that } \sum_{i=0}^{\infty} \frac{d(i)}{10^i} = x \\ &= \{x\} \ \blacksquare \end{aligned}$$

5.4 Incremental Digit Representation

It is convenient to define the following two special digital representations, which are similar to a representation by Nielsen and Kornerup [51]:

Definition 14 *The tuple* $(\mathbb{F}, B, \Delta, \psi, \Omega, \phi)$ *is called an **unsigned incremental digit representation** if*

- *the* ***end-points set*** $\mathbb{F}$ *is a dense strict subset* $\mathbb{F}$ *of* $\mathbb{R}$,
- *the* ***base interval*** B *is a member of* $\mathbb{I}[\mathbb{F}]$,
- *the* ***digit set*** Δ *is a non-empty countable set of symbols,*
- *the* ***digit map*** ψ *is a function* $\Delta \to \mathbb{F}' \to \mathbb{F}'$ *and*
- *the* ***terminator set*** Ω *is a non-empty countable set of symbols and*
- *the* ***terminator map*** ϕ *is a function* $\Omega \to \mathbb{F}'$

where $\mathbb{F}' = \mathbb{F} \cap B$ *and* $\psi(d)$ *is a monotonic function for all* $d \in \Delta$.

An infinite sequence of digits $\langle d_n \rangle_{n=0}^{\infty}$ represents a real number x in B if it induces an infinite sequence of intervals $\langle I_n \rangle_{n=0}^{\infty}$ with

$$I_n = \psi(d_0)(\psi(d_1)(\ldots\psi(d_n)(B)\ldots))$$

such that the induced sequence $\langle I_n \rangle_{n=0}^{\infty}$ is a representative of x. Similarly, a finite sequence of digits $\langle d_n \rangle_{n=0}^{m-1}$ terminated by $\tau \in \Omega$ represents a real number x in B if it induces a finite sequence of intervals $\langle I_n \rangle_{n=0}^{m}$ with

$$I_n = \begin{cases} \psi(d_0)(\psi(d_1)(\ldots\psi(d_n)(B)\ldots)) & \text{if } n < m \\ \psi(d_0)(\psi(d_1)(\ldots\psi(d_{m-1})(\phi(\{\tau\}))\ldots)) & \text{if } n = m \end{cases}$$

such that the induced sequence $\langle I_n \rangle_{n=0}^{m}$ is a representative of x. This unsigned incremental digit representation is equivalent to the digital representation $(\mathbb{F}', \Delta \cup \Omega, \theta)$ with

$$\theta(d)(n) = \begin{cases} \psi(d_0)(\ldots\psi(d_n)(B)\ldots) & \text{if } d_0,\ldots,d_n \in \Delta \\ \psi(d_0)(\ldots\psi(d_{n-1})(\phi(\{d_n\}))\ldots) & \begin{array}{l}\text{if } d_0,\ldots,d_{n-1} \in \Delta \\ \text{and } d_n \in \Omega.\end{array} \end{cases}$$

This representation is incremental in the sense that

$$\psi(d_0) \circ \psi(d_1) \circ \cdots \circ \psi(d_{n+1})$$

can be derived from the composition of

$$\psi(d_0) \circ \psi(d_1) \circ \cdots \circ \psi(d_n)$$

with

$$\psi(d_{n+1}).$$

Definition 15 *The tuple* $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ *is called a* ***signed incremental digit representation*** *if*

- $(\mathbb{F}, B, \Delta, \psi, \Omega, \phi)$ *is an unsigned incremental digit representation,*
- *the* ***sign set*** Σ *is a non-empty countable set of symbols and*
- *the* ***sign map*** φ *is a function* $\Sigma \to \mathbb{F}' \to \mathbb{F}$

where $\mathbb{F}' = \mathbb{F} \cap B$ *and* $\varphi(\sigma)$ *is a monotonic function for all* $\sigma \in \Sigma$.

A real number x in the unsigned incremental digit representation prefixed by a sign $\sigma \in \Sigma$ represents the real number $\varphi(\sigma)(x)$. This signed incremental digit representation is equivalent to the digital representation $(\mathbb{F}, \Sigma \cup \Delta \cup \Omega, \theta)$ with

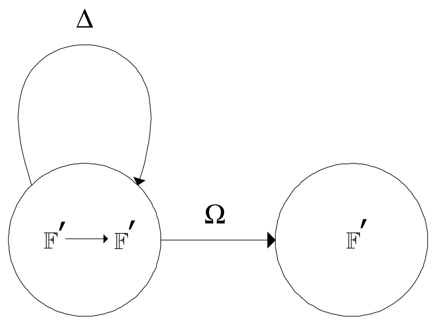
$$\theta(d)(n) = \begin{cases} \varphi(d_0)(\ldots\psi(d_n)(B)\ldots) & \begin{array}{l}\text{if } d_0 \in \Sigma \\ \text{and } d_1, \ldots, d_n \in \Delta\end{array} \\ \varphi(d_0)(\ldots\psi(d_{n-1})(\phi(\{d_n\}))\ldots) & \begin{array}{l}\text{if } d_0 \in \Sigma \\ \text{and } d_1, \ldots, d_{n-1} \in \Delta \\ \text{and } d_n \in \Omega.\end{array} \end{cases}$$

We will see concrete examples shortly.

5.5 The Automaton Connection

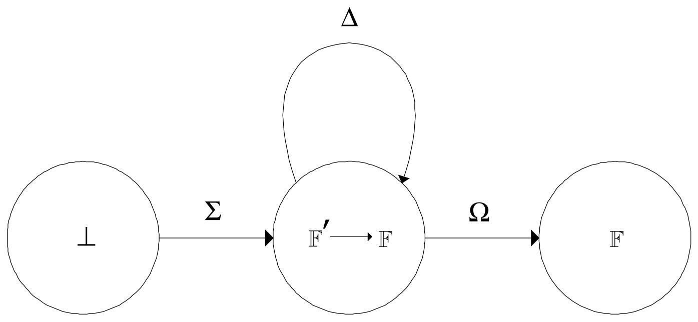
An incremental digit representation can be viewed as an automaton. This view is useful during implementation. For instance, an unsigned incremental digit representation $(\mathbb{F}, B, \Delta, \psi, \Omega, \phi)$ is equivalent to the automaton with states $S = (\mathbb{F}' \to \mathbb{F}') \cup \mathbb{F}'$, input alphabet $I = \Delta \cup \Omega$, output alphabet $O = \mathbb{I}[\mathbb{F}']$, state transition function $f_S : S \times I \to S$, output function $f_O : S \to O$ and starting state $x \mapsto x$ where $\mathbb{F}' = \mathbb{F} \cap B$ and

$$\begin{aligned} f_S(s, d \in \Delta) &= s \circ \psi(d) \\ f_S(s, \tau \in \Omega) &= s \circ \phi(\tau) \\ f_O(s) &= \begin{cases} s(B) & \text{if } s \in \mathbb{F}' \to \mathbb{F}' \\ \{s\} & \text{if } s \in \mathbb{F}'. \end{cases} \end{aligned}$$



Similarly, a signed incremental digit representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ is equivalent to the automaton with states $S = (\mathbb{F}' \to \mathbb{F}) \cup \mathbb{F} \cup \{\perp\}$, input alphabet $I = \Sigma \cup \Delta \cup \Omega$, output alphabet $O = \mathbb{I}[\mathbb{F}]$, state transition function $f_S : S \times I \to S$, output function $f_O : S \to O$ and starting state $\perp$ where $\mathbb{F}' = \mathbb{F} \cap B$ and

$$\begin{aligned}
f_S(s, \sigma \in \Sigma) &= \varphi(\sigma) \\
f_S(s, d \in \Delta) &= s \circ \psi(d) \\
f_S(s, \tau \in \Omega) &= s \circ \phi(\tau) \\
f_O(s) &= \begin{cases} s(B) & \text{if } s \in \mathbb{F}' \to \mathbb{F}' \\ \{s\} & \text{if } s \in \mathbb{F}'. \end{cases}
\end{aligned}$$



5.6 Linear Expansions

It is not hard to see that the decimal representation of the non-negative real numbers corresponds precisely to the signed incremental digit representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{aligned}
\mathbb{F} &= \mathbb{B}(10) \\
B &= [0, 1) \\
\Sigma &= \mathbb{N} \\
\varphi &= d \mapsto x \mapsto d + x
\end{aligned}$$

$$\begin{aligned}
\Delta &= \mathbb{N}(10) \\
\psi &= d \mapsto x \mapsto \frac{d+x}{10} \\
\Omega &= \mathbb{N}(10) - \{0\} \\
\phi &= \tau \mapsto \frac{\tau}{10}.
\end{aligned}$$

For a valid infinite sequence of digits $\langle d_n \rangle_{n=0}^{\infty}$, the induced valid infinite sequence of intervals is

$$\begin{aligned}
& d_0 + [0,1) \\
\supseteq\ & d_0 + \left(\frac{d_1 + [0,1)}{10}\right) \\
\supseteq\ & d_0 + \left(\frac{d_1 + \left(\frac{d_2 + [0,1)}{10}\right)}{10}\right) \\
\supseteq\ & \cdots
\end{aligned}$$

which is equal to

$$\begin{aligned}
& [d_0, d_0 + 1) \\
\supseteq\ & \left[d_0 + \frac{d_1}{10}, d_0 + \frac{d_1 + 1}{10}\right) \\
\supseteq\ & \left[d_0 + \frac{d_1}{10} + \frac{d_2}{100}, d_0 + \frac{d_1}{10} + \frac{d_2 + 1}{100}\right) \\
\supseteq\ & \cdots
\end{aligned}$$

as required. This can be generalized to the *radix* b expansions of the real numbers for any natural *base* b greater than 1 by $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{aligned}
\mathbb{F} &= \mathbb{B}(10) \\
B &= [0,1) \\
\Sigma &= \{+,-\} \times \mathbb{N} \\
\varphi &= (\sigma, d) \mapsto x \mapsto \sigma(d+x) \\
\Delta &= \mathbb{N}(b) \\
\psi &= d \mapsto x \mapsto \frac{d+x}{b} \\
\Omega &= \mathbb{N}(b) - \{0\} \\
\phi &= \tau \mapsto \frac{\tau}{b}.
\end{aligned}$$

and called a *positional representation.* A general linear representation, explored by Escardó [23], can be prescribed by the signed incremental digit representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{aligned}
\mathbb{F} &= \mathbb{Q} \\
B &= [0,1] \\
\Sigma &= \mathbb{Q}^2 \\
\varphi &= (p,q) \mapsto x \mapsto p + (q-p)\,x \\
\Delta &= (\mathbb{Q} \cap [0,1])^2 \\
\psi &= (p,q) \mapsto x \mapsto p + (q-p)\,x \\
\Omega &= \mathbb{Q} \cap [0,1] \\
\phi &= p \mapsto p.
\end{aligned}$$

5.7 Continued Fraction Expansions

A particularly important digital representation of the real numbers are continued fractions. The development

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots}}} \tag{5.1}$$

is called a *continued fraction* [7, 38] and it represents the sequence $\langle c_n \rangle_{n=0}^{\infty}$ where

$$c_n = a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots \, a_{n-1} + \cfrac{b_{n-1}}{a_n}}}}. \tag{5.2}$$

The continued fraction is described as convergent if this sequences converges. The quantity c_n is called the n^{th} *approximant* of the continued fraction. The n^{th} *continuation* is the continued fraction

$$a_n + \cfrac{b_n}{a_{n+1} + \cfrac{b_{n+1}}{a_{n+2} + \cfrac{b_{n+2}}{\ddots}}}$$

A *simple continued fraction* has $b_n = 1$ for all $n \in \mathbb{N}$.

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}} \tag{5.3}$$

For convenience, we will denote the continued fraction in equation (5.1) by

$$[a_0.b_0, a_1.b_1, a_2.b_2, \ldots] .$$

Similarly, we will denote the simple continued fraction in equation (5.3) by

$$[a_0, a_1, a_2, \ldots] .$$

Definition 16 *The* backward sequence $\langle e_n \rangle_{n=1}^{\infty}$ *of the continued fraction*

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots}}}$$

is given by

$$\begin{aligned} e_1 &= a_1 \\ e_n &= a_n + \cfrac{b_{n-1}}{a_{n-1} + \cfrac{b_{n-2}}{a_{n-2} + \cfrac{b_{n-3}}{\ddots \; a_2 + \cfrac{b_1}{a_1}}}} \end{aligned}$$

for $n \in \mathbb{N} - \{0\}$ where e_n is called the n^{th} term.

The following theorem demonstrates an invariance property that will prove extremely useful later in the thesis when we will wish to transform continued fractions while retaining their meaning.

Theorem 17 (Backward Theorem) *The convergent continued fraction*

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots}}}$$

represents the Cauchy sequence $\langle d_n\rangle_{n=0}^{\infty}$ *where*

$$\begin{aligned} d_0 &= a_0 + k_0 \\ d_n &= a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots \, a_{n-1} + \cfrac{b_{n-1}}{a_n + k_n}}}} \end{aligned}$$

for $n \in \mathbb{N} - \{0\}$ *and* $\langle k_n\rangle_{n=0}^{\infty}$ *is any sequence of extended real numbers such that the sequence*

$$\left\langle \frac{e_n}{k_n + e_n} \right\rangle_{n=1}^{\infty}$$

is eventually bounded where e_n *is the* n^{th} *term in the backward sequence of the continued fraction.*

Proof: The sequence $\langle A_n\rangle_{n=0}^{\infty}$ defined by the recurrence relation

$$\begin{aligned} A_0 &= 1 \\ A_1 &= a_0 \\ A_{n+2} &= a_{n+1}A_{n+1} + b_n A_n \end{aligned}$$

and the sequence $\langle B_n\rangle_{n=0}^{\infty}$ defined by the recurrence relation

$$\begin{aligned} B_0 &= 0 & (5.4) \\ B_1 &= 1 & (5.5) \\ B_{n+2} &= a_{n+1}B_{n+1} + b_n B_n. & (5.6) \end{aligned}$$

satisfy

$$\frac{A_{n+1}}{B_{n+1}} = c_n$$

with c_n as defined in equation (5.2) where

$$\lim_{n\to\infty} \frac{A_n}{B_n} = a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots}}}$$

by definition. This can be seen, if we let

$$\begin{aligned} f_n(x) &= a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots \; a_{n-1} + \cfrac{b_{n-1}}{a_n + x}}}} \\ &= \frac{A_{n+1} + A_n x}{B_{n+1} + B_n x} \end{aligned}$$

and observe that

$$\begin{aligned} f_n(0) &= \frac{A_{n+1}}{B_{n+1}} = c_n \\ f_n\left(\frac{b_n}{a_{n+1}}\right) &= \frac{a_{n+1}A_{n+1} + b_n A_n}{a_{n+1}B_{n+1} + b_n B_n} = \frac{A_{n+2}}{B_{n+2}} = c_{n+1}. \end{aligned}$$

Note that $B_n \neq 0$ for all $n \in \mathbb{N}$ because the continued fraction under consideration is convergent. It is now a straightforward matter to show that

$$d_n = \frac{A_n k_n + A_{n+1}}{B_n k_n + B_{n+1}}.$$

We know that given $\epsilon > 0$, there exists $N \in \mathbb{N}$ such that for all $n \geq N$

$$\left| \frac{A_{n+1}}{B_{n+1}} - \frac{A_n}{B_n} \right| < \epsilon.$$

But

$$\frac{A_{n+1}}{B_{n+1}} - \frac{A_n}{B_n} = \frac{A_{n+1}B_n - A_n B_{n+1}}{B_n B_{n+1}}$$

and

$$\frac{A_n k_n + A_{n+1}}{B_n k_n + B_{n+1}} - \frac{A_n}{B_n} = \frac{A_{n+1}B_n - A_n B_{n+1}}{B_n\left(B_n k_n + B_{n+1}\right)}.$$

Therefore

$$\left| \frac{A_n k_n + A_{n+1}}{B_n k_n + B_{n+1}} - \frac{A_n}{B_n} \right| < \left| \frac{B_{n+1}}{B_n k_n + B_{n+1}} \right| \epsilon .$$

But equations (5.4), (5.5) and (5.6) imply that

$$\begin{aligned} \frac{B_2}{B_1} &= a_1 \\ \frac{B_{n+1}}{B_n} &= a_n + \frac{b_{n-1}}{\left(\frac{B_n}{B_{n-1}}\right)} \end{aligned}$$

and so by induction we have

$$\frac{B_{n+1}}{B_n} = e_n. \blacksquare$$

In general, this theorem shows that all but the most contrived sequences $\langle k_n \rangle_{n=0}^{\infty}$ are allowable. For example, the continued fraction for $\sqrt{2}$ is

$$\begin{aligned}
\sqrt{2} &= 1 + \cfrac{1}{1+\sqrt{2}} \\
&= 1 + \cfrac{1}{(1+1) + \cfrac{1}{1+\sqrt{2}}} \\
&= 1 + \cfrac{1}{(1+1) + \cfrac{1}{(1+1) + \cfrac{1}{1+\sqrt{2}}}} \\
&= [1, \langle 2 \rangle_{n=0}^{\infty}]
\end{aligned}$$

with

$$\lim_{n\to\infty} e_n = 1 + \sqrt{2}.$$

Therefore, any sequence $\langle k_n \rangle_{n=0}^{\infty}$ that has no subsequence convergent to $-\left(1+\sqrt{2}\right)$ is allowable. This theorem also leads to the following important corollary.

Corollary 18 *Suppose the continued fraction*

$$x = a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots}}}$$

with positive coefficients converges to $x \in \mathbb{R}$ then the intervals

$$I_n = a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots \, a_{n-1} + \cfrac{b_{n-1}}{a_n + [0,\infty]}}}}$$

for $n \in \mathbb{N}$ are nested and

$$\{x\} = \bigcap_{n=0}^{\infty} I_n.$$

Proof: The sequence $\langle I_n \rangle_{n=0}^{\infty}$ of intervals is nested because

$$a_n + \frac{b_n}{[0,\infty]} \subseteq [0,\infty]$$

for all $n \in \mathbb{N}$. Using the definitions in theorem 17, $a_n, b_n > 0$ implies that $B_n > 0$. Therefore $e_n > 0$. Consider any sequence $\langle k_n \rangle_{n=0}^{\infty}$ with $k_n \in [0,\infty]$. Clearly

$$\left| \frac{e_n}{k_n + e_n} \right| \leq 1.$$

Therefore the sequence $\langle d_n \rangle_{n=0}^{\infty}$ converges to x. ■

In 1899, Pringsheim [65] showed that the divergence of the series

$$\sum_{n=2}^{\infty} \sqrt{\frac{b_{n-1} b_n}{a_n}} \tag{5.7}$$

is a necessary and sufficient condition for the convergence of the continued fraction $[a_0.b_0, a_1.b_1, a_2.b_2, \ldots]$ with positive coefficients.

A continued fraction can be viewed as a sequence of intervals provided some restrictions are applied to the coefficients. Firstly, let us consider $\mathbb{N}$-fractions, also known as *regular continued fractions*. The $\mathbb{N}$-fraction of a real number x is the simple continued fraction $[a_0, a_1, a_2, \ldots]$ with a_n recursively given by

$$\begin{aligned} (a_0, r_0) &= (\lfloor x \rfloor, x - \lfloor x \rfloor) \\ (a_{n+1}, r_{n+1}) &= \left(\left\lfloor \frac{1}{r_n} \right\rfloor, \frac{1}{r_n} - \left\lfloor \frac{1}{r_n} \right\rfloor \right). \end{aligned}$$

The procedure stops if $r_n = 0$. Observe that

$$0 \leq y - \lfloor y \rfloor < 1$$

for any real number y. Therefore

$$0 \leq r_n < 1 \Rightarrow r_n \in [0,1) \Rightarrow \frac{1}{r_n} \in (1,\infty].$$

It can be shown that the set of $\mathbb{N}$-fractions defined this way is isomorphic to the set of real numbers. Note that the above procedure excludes sequences that terminate with a one. It can be shown that a real number is rational if and only if its $\mathbb{N}$-fraction is finite and an irrational number is algebraic of degree 2 if and only if its $\mathbb{N}$-fraction is periodic. An $\mathbb{N}$-fraction denoted by $[\langle a_i \rangle_{i=0}^{\infty}]$ induces the following sequence of intervals

$$\begin{array}{ll} & a_0 + \cfrac{1}{(1,\infty]} \\ \supseteq & a_0 + \cfrac{1}{a_1 + \cfrac{1}{(1,\infty]}} \\ \supseteq & a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{(1,\infty]}}} \\ \supseteq & \cdots \end{array}$$

or equivalently

$$\begin{array}{ll} & [a_0, a_0 + 1) \\ \supseteq & \left(a_0 + \dfrac{1}{a_1 + 1}, a_0 + \dfrac{1}{a_1}\right] \\ \supseteq & \left[a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2}}, a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + 1}}\right) \\ \supseteq & \cdots . \end{array}$$

It follows that the signed incremental digit representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{array}{rcl} \mathbb{F} & = & \mathbb{Q} \\ B & = & (1,\infty] \\ \Sigma & = & \mathbb{Z} \\ \varphi & = & \sigma \mapsto x \mapsto \sigma + \dfrac{1}{x} \\ \Delta & = & \mathbb{N} - \{0\} \\ \psi & = & d \mapsto x \mapsto d + \dfrac{1}{x} \\ \Omega & = & \mathbb{N} - \{0, 1\} \\ \phi & = & \tau \mapsto \tau \end{array}$$

faithfully represents $\mathbb{N}$-fractions. For example, the $\mathbb{N}$-fraction for $\sqrt{2}$ is

$$\begin{array}{rcl} \sqrt{2} & = & [1, \langle 2 \rangle_{n=0}^{\infty}] \\ & = & [1,2) \supseteq \left(\frac{4}{3}, \frac{3}{2}\right] \supseteq \left[\frac{7}{5}, \frac{10}{7}\right) \supseteq \left(\frac{24}{17}, \frac{17}{12}\right] \supseteq \cdots \end{array}$$

and the $\mathbb{N}$-fraction for the golden ratio ϕ and the natural number e are

$$\begin{aligned} \phi &= \frac{\sqrt{5}+1}{2} = [\langle 1\rangle_{n=0}^{\infty}] \\ e &= [2, \langle 1, 2n+2, 1\rangle_{n=0}^{\infty}]\,. \end{aligned}$$

Here is another surprising $\mathbb{N}$-fraction

$$\sqrt{2}\frac{\exp\left(\sqrt{2}\right)+1}{\exp\left(\sqrt{2}\right)-1} = [\langle 2+8n, 3+4n\rangle_{n=0}^{\infty}]\,.$$

Definition 19 *The* continuation function $\xi_{\mathbb{S}} : \mathbb{Z} \to \mathcal{P}\left(\mathbb{R}^{\infty}\right)$ *for a particular type, $\mathbb{S}$ say, of continued fraction, $\mathbb{S}$-fractions say, is defined as*

$$\xi_{\mathbb{S}}\left(n\right) = \left\{[a_0, a_1, a_2, \ldots] \in \mathbb{S}\text{-}\mathit{fractions}\,|a_0 = n\right\}.$$

The continuation function is best understood as defining the set of real numbers that are expressible by continued function given the first digit. So, using this definition, we have

$$\begin{aligned} \xi_{\mathbb{N}}\left(n\right) &= \left\{[a_0, a_1, a_2, \ldots] \in \mathbb{N}\text{-fractions}\,|a_0 = n\right\} \\ &= [n] \cup \left\{[n, a_1, a_2, \ldots]\,|a_i \in \mathbb{N}-\{0\}\right\} \\ &= \{n\} \cup (n, n+1) \\ &= [n, n+1) \end{aligned}$$

as expected.

Secondly, let us consider $\mathbb{Z}$-fractions. The $\mathbb{Z}$-fractions of a real number x is the simple continued fraction $[a_0, a_1, a_2, \ldots]$ with a_n recursively given by

$$\begin{aligned} (a_0, r_0) &= \left(\lfloor x\rceil, x - \lfloor x\rceil\right) \\ (a_{n+1}, r_{n+1}) &= \left(\left\lfloor\frac{1}{r_n}\right\rceil, \frac{1}{r_n} - \left\lfloor\frac{1}{r_n}\right\rceil\right). \end{aligned}$$

The procedure stops if $r_n = 0$. Observe that

$$-\frac{1}{2} < y - \lfloor y\rceil \leq \frac{1}{2}$$

for any real number y. Therefore

$$-\frac{1}{2} < r_n \leq \frac{1}{2} \Rightarrow r_n \in \left(-\frac{1}{2}, \frac{1}{2}\right] \Rightarrow \frac{1}{r_n} \in [2, -2)\,.$$

It can be shown that the set of $\mathbb{Z}$-fractions defined this way is isomorphic to the set of real numbers. Note that the above procedure ensures that positive digits are not proceeded by minus two, negative digits are not preceded by two and excludes sequences that terminate with a minus two. A $\mathbb{Z}$-fraction denoted by $[\langle a_i; \rangle_{i=0}^{\infty}]$ induces the following sequence of intervals

$$\begin{aligned}
& a_0 + \frac{1}{[2,-2)} \\
\supseteq\ & a_0 + \cfrac{1}{a_1 + \cfrac{1}{[2,-2)}} \\
\supseteq\ & a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{[2,-2)}}} \\
\supseteq\ & \cdots
\end{aligned}$$

equal to

$$\begin{aligned}
& \left(a_0 - \tfrac{1}{2}, a_0 + \tfrac{1}{2}\right] \\
\supseteq\ & \left[a_0 + \frac{1}{a_1 + \frac{1}{2}}, a_0 + \frac{1}{a_1 - \frac{1}{2}}\right) \\
\supseteq\ & \left(a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 - \frac{1}{2}}}, a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \frac{1}{2}}}\right] \\
\supseteq\ & \cdots.
\end{aligned}$$

It follows that the signed incremental representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{aligned}
\mathbb{F} &= \mathbb{Q} \\
B &= [2,-2) \\
\Sigma &= \mathbb{Z} \\
\varphi &= \sigma \mapsto x \mapsto \sigma + \frac{1}{x} \\
\Delta &= \mathbb{Z} - \{-1, 0, 1\} \\
\psi &= d \mapsto x \mapsto d + \frac{1}{x} \\
\Omega &= \mathbb{Z} - \{-2, -1, 0, 1\} \\
\phi &= \tau \mapsto \tau
\end{aligned}$$

captures the notion of a $\mathbb{Z}$-fraction. For example, the $\mathbb{Z}$-fraction for the golden ratio ϕ and the natural number e are

$$\begin{aligned}
\phi &= 2+\cfrac{1}{-3+\cfrac{1}{1+\phi}} \\
&= 2+\cfrac{1}{-3+\cfrac{1}{(1+2)+\cfrac{1}{-3+\cfrac{1}{1+\phi}}}} \\
&= \left[2,\langle -3,3\rangle_{n=0}^{\infty}\right] \\
e &= \left[3,-4,\langle 2,4n+5,-2,-4n-7\rangle_{n=0}^{\infty}\right].
\end{aligned}$$

Coincidently, the $\mathbb{Z}$-fraction for $\sqrt{2}$ is the same as its $\mathbb{N}$-fraction, but its interpretation is very different.

$$\begin{aligned}
\sqrt{2} &= \left[1,\langle 2\rangle_{n=0}^{\infty}\right] \\
&= \left(\tfrac{1}{2},\tfrac{3}{2}\right] \supseteq \left[\tfrac{7}{5},\tfrac{5}{3}\right) \supseteq \left(\tfrac{11}{8},\tfrac{17}{12}\right] \supseteq \left[\tfrac{41}{29},\tfrac{27}{19}\right) \supseteq \cdots
\end{aligned}$$

Also, note that the continuation function for $\mathbb{Z}$-fraction is given by

$$\begin{aligned}
\xi_{\mathbb{Z}}(n) &= \{[a_0,a_1,a_2,\ldots]\in \mathbb{Z}\text{-fractions}\,|a_0=n\} \\
&= [n]\cup[n,2]\cup\{[n,a_1,a_2,\ldots]\,|a_i\in\mathbb{Z}-\{-1,0,1\}\} \\
&= \{n\}\cup\left\{n+\tfrac{1}{2}\right\}\cup\left(n-\tfrac{1}{2},n\right)\cup\left(n,n+\tfrac{1}{2}\right) \\
&= \left(n-\tfrac{1}{2},n+\tfrac{1}{2}\right]
\end{aligned}$$

as expected.

Many continued fractions for elementary functions have been derived by using various techniques [53, 72] from their Taylor series, most notably by Euler [24, 25].

Chapter 6

Rational Function Approximations

In this chapter, we examine the theoretical basis for representing real functions by sequences of rational functions by consolidating a variety of background material. We present certain known algorithms that can be easily implemented in a mathematical programming language such as Mathematica. The author found these algorithms useful in the search for elegant algorithms for the elementary functions.

6.1 Padé Approximants

A Padé approximant [4] to a function is a particular type of rational function approximation.

Definition 20 *The* L, M Padé Approximant *to* $f\left(x\right)$ *is the rational function, denoted by*

$$[L/M] = \frac{P(x)}{Q(x)},$$

where $P(x)$ *is a polynomial of degree at most* L *and* $Q(x)$ *is a polynomial of degree at most* M*. The coefficients of* $P(x)$ *and* $Q(x)$ *are determined from the Taylor series of* f *at* 0

$$f\left(x\right) = \sum_{n=0}^{\infty} a_n x^n$$

together with the conditions

$$\begin{aligned} f\left(x\right) - \frac{P(x)}{Q(x)} &= O\left(x^{L+M+1}\right), \\ Q(0) &= 1 \end{aligned}$$

and $P(x)$ and $Q(x)$ have no common factors [4, page 5].

Theorem 21 (uniqueness) *When the $[L/M]$ Padé approximant to a Taylor series $f(x)$ exists, then it is unique [4, page 8].*

In 1892, Padé [52] emphasized the importance of displaying the Padé approximants in tabular form in order to study their structure. The array

$$\begin{array}{cccccc}
[0/0] & [0/1] & [0/2] & [0/3] & [0/4] & \cdots \\
[1/0] & [1/1] & [1/2] & [1/3] & [1/4] & \cdots \\
[2/0] & [2/1] & [2/2] & [2/3] & [2/4] & \cdots \\
[3/0] & [3/1] & [3/2] & [3/3] & [3/4] & \cdots \\
[4/0] & [4/1] & [4/2] & [4/3] & [4/4] & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

is known as a Padé table. The partial sums of the Taylor series of f at 0 occupy the first column of the table. However, in general taking the sequence of rational functions down the diagonal provides faster and more efficient convergence and it is this property that is the basis of our interest. In particular, it turns out that continued fractions occupy this part of the Padé table as we shall see below. An analytic formula for the Padé approximant $[L/M]$ can be derived and from this formula it is evident that $C(L/M) \neq 0$ is a sufficient condition for its existence where

$$C(r/s) = \mathsf{det}\begin{vmatrix}
a_{r-s+1} & a_{r-s+2} & \cdots & a_r \\
a_{r-s+2} & a_{r-s+3} & \cdots & a_{r+1} \\
\vdots & \vdots & \ddots & \vdots \\
a_r & a_{r+1} & \cdots & a_{r+s-1}
\end{vmatrix}$$

as defined by Frobenius [26]. The array

$$\begin{array}{cccccc}
C(0/0) & C(0/1) & C(0/2) & C(0/3) & C(0/4) & \cdots \\
C(1/0) & C(1/1) & C(1/2) & C(1/3) & C(1/4) & \cdots \\
C(2/0) & C(2/1) & C(2/2) & C(2/3) & C(2/4) & \cdots \\
C(3/0) & C(3/1) & C(3/2) & C(3/3) & C(3/4) & \cdots \\
C(4/0) & C(4/1) & C(4/2) & C(4/3) & C(4/4) & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

is known as the C table. The C table can be efficiently computed [4] using the equations

$$\begin{aligned}
C(L/0) &= 1 \qquad (6.1)\\
C(L/1) &= a_L
\end{aligned}$$

$$\begin{aligned} C\,(0/M) &= (-1)^{\frac{1}{2}M(M-1)}\,a_0^M \\ C(L/M) &= \frac{C(L-1/M-1)C(L+1/M-1)-C(L/M-1)^2}{C(L/M-2)}. \end{aligned}$$

The relative position of the C table elements referenced in equation (6.1) are

	$C\,(L-1/M-1)$	
$C\,(L/M-2)$	$C\,(L/M-1)$	$C\,(L/M)$
	$C\,(L+1/M-1)$	.

6.2 Equivalence Transformation

Later in this thesis we will derive various algorithms for the transcendental functions from various continued fractions. In fact, we will be particularly interested in continued fractions with positive coefficients. In this light, it is useful to consider what transformations can be found in the literature. The so-called *equivalence transformation*

$$[\langle a_0.b_0\rangle_{n=0}^{\infty}] = [a_0.c_0b_0, \langle c_{n-1}a_n.c_{n-1}c_nb_n\rangle_{n=1}^{\infty}]$$

where c_n are any non-zero numbers, allows us to transform some continued fractions to ones with positive coefficients [4, page 44]. In particular,

$$[\langle a_0.b_0\rangle_{n=0}^{\infty}] = [a_0. - b_0, \langle -a_n.b_n\rangle_{n=1}^{\infty}]$$

when $c_n = -1$.

6.3 Stieltjes Type Continued Fraction

The *corresponding* [53] or *Stieltjes type* [72] continued fraction to the power series $\sum_{n=0}^{\infty} a_n x^n$ is

$$[a_0.a_1, \langle 1.b_n, 1.c_n\rangle_{n=0}^{\infty}]$$

where

$$b_n = -\frac{C(n+2/n+1)C(n/n)}{C(n+1/n)C(n+1/n+1)} \tag{6.2}$$

$$c_n = -\frac{C(n+2/n+2)C(n+1/n)}{C(n+1/n+1)C(n+2/n+1)} \tag{6.3}$$

[4, page 57]. The sequence of approximants for this continued fraction correspond to the *stair step sequence* of Padé approximants

$$\begin{array}{ccccc}
[0/0] & & & & \\
[1/0] & [1/1] & & & \\
 & [2/1] & [2/2] & & \\
 & & [3/2] & [3/3] & \\
 & & & [4/3] & [4/4] \\
 & & & & \ddots \quad \ddots
\end{array}$$

to $\sum_{n=0}^{\infty} a_n x^n$.

6.4 Jacobi Type Continued Fraction

The *associated* [53] or *Jacobi type* [72] continued fraction to the power series $\sum_{n=0}^{\infty} a_n x^n$ is

$$\left[a_0.a_1x, 1+b_0x. - b_0c_0x^2, \left\langle 1+(c_{n-1}+b_n)x. - b_nc_nx^2\right\rangle_{n=1}^{\infty}\right]$$

where b_n and c_n are as given in equations (6.2) and (6.3) [4, page 56]. The sequence of approximants for this continued fraction corresponds to the *diagonal sequence* of Padé approximants

$$\begin{array}{cccccc}
[0/0] & & & & & \\
 & [1/1] & & & & \\
 & & [2/2] & & & \\
 & & & [3/3] & & \\
 & & & & [4/4] & \\
 & & & & & \ddots
\end{array}$$

to $\sum_{n=0}^{\infty} a_n x^n$.

6.5 Euler type Continued Fraction

The *equivalent* or *Euler type* continued fraction to the power series $\sum_{n=0}^{\infty} a_n x^n$ is

$$\left[0.a_0, 1. - \frac{a_1}{a_0}x, \left\langle 1+\frac{a_n}{a_{n-1}}x. - \frac{a_{n+1}}{a_n}x\right\rangle_{n=1}^{\infty}\right].$$

The sequence of approximants for this continued fraction correspond to the *vertical sequence* of Padé approximants

$$\begin{array}{c}[0/0]\\ [1/0]\\ [2/0]\\ [3/0]\\ [4/0]\\ \vdots\end{array}$$

to $\sum_{n=0}^{\infty} a_n x^n$; namely the partial sums of the power series.

For example

$$\begin{aligned}
\sin(x) &= \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \\
&= \left[0.1, 1.\frac{x^2}{6}, \left\langle 1 - \frac{x^2}{2n(2n+1)} . \frac{x^2}{(2n+2)(2n+3)} \right\rangle_{n=1}^{\infty}\right], \\
\cos(x) &= \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \\
&= \left[0.1, 1.\frac{1}{2}x^2, \left\langle 1 - \frac{x^2}{2n(2n-1)} . \frac{x^2}{(2n+1)(2n+2)} \right\rangle_{n=1}^{\infty}\right], \\
\exp(-x) &= \sum_{n=0}^{\infty} \frac{(-x)^n}{n!} \\
&= \left[0.1, 1.x, \left\langle 1 - \frac{x}{n} . \frac{x}{n+1} \right\rangle_{n=1}^{\infty}\right].
\end{aligned}$$

6.6 The Hypergeometric Function

6.6.1 The Ordinary Hypergeometric Function

In terms of the *Pochhammer symbol*

$$\begin{aligned}
(a)_0 &= 1 \\
(a)_n &= a(a+1)(a+2)\cdots(a+n-1) = \frac{\Gamma(a+n)}{\Gamma(a)},
\end{aligned}$$

the *ordinary hypergeometric function* is defined by

$${}_2F_1(a, b; c; z) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!}$$

where a, b, c and z are complex numbers. This series converges for $|z| < 1$ so long as c is not a negative integer or zero. The hyperbolic differential equation

$$z\left(1-z\right)\frac{d^2y}{dz^2} + c - \left(1+a+b\right)z\frac{dy}{dz} - aby = 0$$

is satisfied by ${}_2F_1(a,b;c;z)$ together with twenty-three other variants [43], whose regions of convergence cover the entire complex plane. These expressions lead to an analytic continuation of the hypergeometric function to the remainder of the complex plane outside the unit circle. The ordinary hypergeometric function captures many interesting holomorphic functions.

$$\begin{aligned}
\log(1+x) &= x\,{}_2F_1\left(1,1;2;-x\right)\\
(1+x)^a &= {}_2F_1\left(-a,b;b;-x\right)\\
\arcsin\left(x\right) &= x\,{}_2F_1\left(\tfrac{1}{2},\tfrac{1}{2};\tfrac{3}{2};x^2\right)\\
\mathrm{arcsinh}\left(x\right) &= x\,{}_2F_1\left(\tfrac{1}{2},\tfrac{1}{2};\tfrac{3}{2};-x^2\right)\\
\arctan\left(x\right) &= x\,{}_2F_1\left(\tfrac{1}{2},1;\tfrac{3}{2};-x^2\right)\\
\mathrm{arctanh}\left(x\right) &= x\,{}_2F_1\left(\tfrac{1}{2},1;\tfrac{3}{2};x^2\right)\\
\sin\left(ax\right) &= {}_2F_1\left(\tfrac{1+a}{2},\tfrac{1-a}{2};\tfrac{3}{2};\sin\left(x\right)^2\right)\\
\cos\left(ax\right) &= {}_2F_1\left(\tfrac{a}{2},-\tfrac{a}{2};\tfrac{1}{2};\sin\left(x\right)^2\right)\\
\sinh\left(ax\right) &= {}_2F_1\left(\tfrac{1+a}{2},\tfrac{1-a}{2};\tfrac{3}{2};\sinh\left(x\right)^2\right)\\
\cosh\left(ax\right) &= {}_2F_1\left(\tfrac{a}{2},-\tfrac{a}{2};\tfrac{1}{2};\sinh\left(x\right)^2\right)
\end{aligned}$$

Gauss derived the following continued fraction

$$\begin{aligned}
\frac{{}_2F_1\left(a,b+1;c+1;x\right)}{{}_2F_1\left(a,b;c;x\right)} &= \left[0.1,\left\langle 1.-\frac{\left(n+a\right)\left(n+c-b\right)}{\left(2n+c\right)\left(2n+c+1\right)}x,\right.\right.\\
&\quad \left.\left. 1.-\frac{\left(n+b+1\right)\left(n+c-a+1\right)}{\left(2n+c+1\right)\left(2n+c+2\right)}x\right\rangle_{n=0}^{\infty}\right]. \qquad (6.4)
\end{aligned}$$

Since ${}_2F_1(a,0;c;x) = 1$, equation (6.4) becomes

$$\begin{aligned}
{}_2F_1\left(a,1;c;x\right) &= \left[0.1,\left\langle 1.-\frac{\left(n+a\right)\left(n+c-1\right)}{\left(2n+c-1\right)\left(2n+c\right)}x,\right.\right.\\
&\quad \left.\left. 1.-\frac{\left(n+1\right)\left(n+c-a\right)}{\left(2n+c\right)\left(2n+c+1\right)}x\right\rangle_{n=0}^{\infty}\right].
\end{aligned}$$

In particular,

$$\begin{aligned}
\log(1+x) &= x\,{}_2F_1\left(1,1;2;-x\right)\\
&= \left[0.x,\left\langle 1.\frac{n+1}{4n+2}x, 1.\frac{n+1}{4n+6}x\right\rangle_{n=0}^{\infty}\right],
\end{aligned}$$

$$\begin{aligned}\operatorname{arctanh}(x) &= x\,{}_2F_1\left(\tfrac{1}{2},1;\tfrac{3}{2};x^2\right)\\ &= \left[0.x,\left\langle 1.-\frac{(2n+1)^2}{(4n+1)(4n+3)}x^2,\right.\right.\\ &\qquad \left.\left. 1.-\frac{(2n+2)^2}{(4n+3)(4n+5)}x^2\right\rangle_{n=0}^{\infty}\right]\\ &= \left[0.x,\left\langle 1.-\frac{n^2}{4n^2-1}x^2\right\rangle_{n=1}^{\infty}\right],\end{aligned}$$

$$\begin{aligned}(1+x)^y &= {}_2F_1(-y,1;1;-x)\\ &= \left[0.1,\left\langle 1.\frac{n-y}{4n+2}x, 1.\frac{n+1+y}{4n+2}x\right\rangle_{n=0}^{\infty}\right],\end{aligned}$$

$$\begin{aligned}\int_0^x \frac{t^{p-1}}{(1+t^q)}dt &= \frac{x^p}{p}\,{}_2F_1\left(\frac{p}{q},1;\frac{p}{q}+1;-x^q\right)\\ &= \left[0.\frac{x^p}{p},\left\langle 1.\frac{(qn+p)^2}{(2qn+p)(2qn+p+q)}x^q,\right.\right.\\ &\qquad \left.\left. 1.\frac{(qn+q)^2}{(2qn+p+q)(2qn+p+2q)}x^q\right\rangle_{n=0}^{\infty}\right],\end{aligned}$$

$$\begin{aligned}\arctan(x) &= \int_0^x \frac{1}{(1+t^2)}dt\\ &= \left[0.x,\left\langle 1.\frac{(2n+1)^2}{(4n+1)(4n+3)}x^2,\right.\right.\\ &\qquad \left.\left. 1.\frac{(2n+2)^2}{(4n+3)(4n+5)}x^2\right\rangle_{n=0}^{\infty}\right]\\ &= \left[0.x,\left\langle 1.\frac{n^2}{4n^2-1}x^2\right\rangle_{n=1}^{\infty}\right],\end{aligned}$$

$$\frac{\arcsin(x)}{\sqrt{1-x^2}} = \frac{x\,{}_2F_1\left(\frac{1}{2},\frac{1}{2};\frac{3}{2};x^2\right)}{{}_2F_1\left(\frac{1}{2},-\frac{1}{2};\frac{1}{2};x^2\right)}$$

$$\begin{aligned}
&= \left[0.x, \left\langle 1. - \frac{(2n+1)(2n+2)}{(4n+1)(4n+3)}x^2, \right.\right.\\
&\qquad \left.\left. 1. - \frac{(2n+1)(2n+2)}{(4n+3)(4n+5)}x^2 \right\rangle_{n=0}^{\infty}\right],
\end{aligned}$$

$$\begin{aligned}
\frac{(1+x)^y-(1-x)^y}{(1+x)^y+(1-x)^y} &= xy\frac{{}_2F_1\left(\frac{1}{2}(1-y), \frac{1}{2}(2-y); \frac{3}{2}; x^2\right)}{{}_2F_1\left(\frac{1}{2}(1-y), -\frac{1}{2}y; \frac{1}{2}; x^2\right)}\\
&= \left[0.xy, \left\langle 1.\frac{y^2-(2n+1)^2}{(4n+1)(4n+3)}x^2, \right.\right.\\
&\qquad \left.\left. 1.\frac{y^2-(2n+2)^2}{(4n+3)(4n+5)}x^2 \right\rangle_{n=0}^{\infty}\right]\\
&= \left[0.xy, \left\langle 1.\frac{y^2-n^2}{4n^2-1}x^2 \right\rangle_{n=1}^{\infty}\right]
\end{aligned}$$

and

$$(1+x)^y = \left[1.xy, \left\langle 1.\frac{x(n-y)}{2(2n-1)}, 1.\frac{x(n+y)}{2(2n+1)} \right\rangle_{n=0}^{\infty}\right].$$

6.6.2 The Kummer Confluent Hypergeometric Function

The *Kummer confluent hypergeometric function* is defined by

$${}_1F_1(b; c; x) = \sum_{n=0}^{\infty} \frac{(b)_n}{(c)_n}\frac{x^n}{n!}.$$

The confluent hypergeometric function captures many interesting holomorphic functions including the *incomplete gamma function* $\gamma(a, x)$ and the *error function* $\operatorname{erf}(x)$

$$\begin{aligned}
\exp(x) &= {}_1F_1(b; b; x)\\
\operatorname{erf}(x) &= \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\mathrm{d}t = \frac{2x}{\sqrt{\pi}}\,{}_1F_1\left(\frac{1}{2}; \frac{3}{2}; -x^2\right)\\
\gamma(a, x) &= \int_0^x e^{-t}t^{a-1}\mathrm{d}t = \frac{x^a e^x}{a}\,{}_1F_1(1; a+1; x).
\end{aligned}$$

The confluent hypergeometric function can also be deduced as a special case of the ordinary hypergeometric function using

$${}_1F_1(b; c; x) = \lim_{a\to\infty} {}_2F_1\left(a, b; c; \frac{x}{a}\right).$$

Therefore, the continued fraction in equation (6.4) can be specialized to

$$\begin{aligned}\frac{{}_1F_1(b+1;c+1;x)}{{}_1F_1(b;c;x)} &= \left[0.1, \left\langle 1. - \frac{n+c-b}{(2n+c)(2n+c+1)}x,\right.\right.\\ &\quad \left.\left. 1.\frac{n+b+1}{(2n+c+1)(2n+c+2)}x\right\rangle_{n=0}^{\infty}\right]. \qquad (6.5)\end{aligned}$$

Alternatively, by symmetry, it can be specialized to

$$\begin{aligned}\frac{{}_1F_1(a;c+1;x)}{{}_1F_1(a;c;x)} &= \left[0.1, \left\langle 1.\frac{n+a}{(2n+c)(2n+c+1)}x,\right.\right.\\ &\quad \left.\left. 1. - \frac{n+c-a+1}{(2n+c+1)(2n+c+2)}x\right\rangle_{n=0}^{\infty}\right]. \qquad (6.6)\end{aligned}$$

Since ${}_1F_1(0;c;x) = 1$, equation (6.5) becomes

$$\begin{aligned}{}_1F_1(1;c;x) &= \left[0.1, \left\langle 1. - \frac{n+c-1}{(2n+c-1)(2n+c)}x,\right.\right.\\ &\quad \left.\left. 1.\frac{n+1}{(2n+c)(2n+c+1)}x\right\rangle_{n=0}^{\infty}\right].\end{aligned}$$

In particular,

$$\begin{aligned}\mathsf{exp}(x) &= {}_1F_1(1;1;x)\\ &= \left[0.1, \left\langle 1. - \frac{1}{4n+2}x, 1.\frac{1}{4n+2}x\right\rangle_{n=0}^{\infty}\right],\end{aligned}$$

the error function

$$\begin{aligned}\mathsf{erf}(x) &= \frac{2xe^{-x^2}}{\sqrt{\pi}}\frac{{}_1F_1\left(\frac{1}{2};\frac{3}{2};-x^2\right)}{{}_1F_1\left(\frac{1}{2};\frac{1}{2};-x^2\right)}\\ &= \frac{2xe^{-x^2}}{\sqrt{\pi}}\left[0.1, \left\langle 1. - \frac{4n+2}{(4n+1)(4n+3)}x^2,\right.\right.\\ &\quad \left.\left. 1.\frac{4n+4}{(4n+3)(4n+5)}x^2\right\rangle_{n=0}^{\infty}\right]\end{aligned}$$

and the incomplete gamma function

$$\begin{aligned}\gamma(a,x) &= \int_0^x e^{-t}t^{a-1}\mathrm{d}t = \frac{x^a e^x}{a}\,{}_1F_1(1;a+1;x)\\ &= \frac{x^a e^x}{a}\left[0.1, \left\langle 1. - \frac{n+a}{(2n+a)(2n+a+1)}x,\right.\right.\\ &\quad \left.\left. 1.\frac{n+1}{(2n+a+1)(2n+a+2)}x\right\rangle_{n=0}^{\infty}\right].\end{aligned}$$

6.6.3 The 0-1 Confluent Hypergeometric Function

The *0-1 confluent hypergeometric function* is defined by

$$ {}_0F_1\left(;c;x\right) = \sum_{n=0}^{\infty} \frac{1}{(c)_n} \frac{x^n}{n!}. $$

Again, this can also be deduced as a special case of the confluent hypergeometric function ${}_1F_1\left(b;c;x\right)$ using

$$ {}_0F_1\left(;c;x\right) = \lim_{b\to\infty} {}_1F_1\left(b;c;\frac{x}{b}\right). $$

Therefore, the continued fraction in equation (6.5) or (6.6) can be specialized to

$$ \begin{aligned} & \frac{{}_0F_1\left(;c+1;x\right)}{{}_0F_1\left(;c;x\right)} \\ = & \left[0.1, \left\langle 1.\frac{1}{(2n+c)(2n+c+1)}x, 1.\frac{1}{(2n+c+1)(2n+c+2)}x \right\rangle_{n=0}^{\infty}\right] \\ = & \left[0.1, \left\langle 1.\frac{1}{(n+c)(n+c+1)}x \right\rangle_{n=0}^{\infty}\right]. \end{aligned} $$

This confluent hypergeometric function captures many interesting holomorphic functions including the *Bessel function* $J_a(x)$ of the first kind.

$$ \begin{aligned} J_a(x) &= \left(\frac{1}{2}x\right)^a \frac{{}_0F_1\left(;a+1;-\frac{1}{4}x^2\right)}{\Gamma(a+1)} \\ \mathsf{sin}(x) &= x\,{}_0F_1\left(;\tfrac{3}{2};-\tfrac{1}{4}x^2\right) \\ \mathsf{sinh}(x) &= x\,{}_0F_1\left(;\tfrac{3}{2};\tfrac{1}{4}x^2\right) \\ \mathsf{cos}(x) &= {}_0F_1\left(;\tfrac{1}{2};-\tfrac{1}{4}x^2\right) \\ \mathsf{cosh}(x) &= {}_0F_1\left(;\tfrac{1}{2};\tfrac{1}{4}x^2\right) \end{aligned} $$

$$ \begin{aligned} \frac{J_a(x)}{J_{a-1}(x)} &= \frac{x}{2a}\frac{{}_0F_1\left(;a+1;-\frac{1}{4}x^2\right)}{{}_0F_1\left(;a;-\frac{1}{4}x^2\right)} \\ &= \left[0.\frac{x}{2a}, \left\langle 1. - \frac{1}{4(n+a)(n+a+1)}x^2 \right\rangle_{n=0}^{\infty}\right]. \end{aligned} $$

In particular, Lambert's continued fraction for $\mathsf{tan}(x)$ can be derived

$$ \begin{aligned} \mathsf{tan}(x) &= \frac{\mathsf{sin}(x)}{\mathsf{cos}(x)} = \frac{{}_0F_1\left(;\frac{3}{2};-\frac{1}{4}x^2\right)}{{}_0F_1\left(;\frac{1}{2};-\frac{1}{4}x^2\right)} \\ &= \left[0.1, \left\langle 1. - \frac{1}{(2n+1)(2n+3)}x^2 \right\rangle_{n=0}^{\infty}\right] \end{aligned} $$

and similarly for $\mathsf{tanh}(x)$

$$\begin{aligned}\mathsf{tanh}(x) &= \frac{\mathsf{sinh}(x)}{\mathsf{cosh}(x)} = \frac{{}_0F_1\left(;\frac{3}{2};\frac{1}{4}x^2\right)}{{}_0F_1\left(;\frac{1}{2};\frac{1}{4}x^2\right)} \\ &= \left[0.1, \left\langle 1.\frac{1}{(2n+1)(2n+3)}x^2\right\rangle_{n=0}^{\infty}\right].\end{aligned}$$

6.6.4 The 2-0 Confluent Hypergeometric Function

The *2-0 confluent hypergeometric function* is defined by

$${}_2F_0(a,b;;x) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{1} \frac{x^n}{n!}.$$

The confluent hypergeometric function can also be deduced as a special case of the ordinary hypergeometric function using

$${}_2F_0(a,b;;x) = \lim_{c\to\infty} {}_2F_1(a,b;c;cx).$$

Therefore, the continued fraction in equation (6.4) can be specialized to

$$\frac{{}_2F_0(a,b+1;;-x)}{{}_2F_0(a,b;;-x)} = \left[0.1, \langle 1.(n+a)x, 1.(n+b+1)x\rangle_{n=0}^{\infty}\right].$$

Since ${}_2F_0(a,0;;x) = 1$, it follows that

$${}_2F_0(a,1;;-x) = \left[0.1, \langle 1.(n+a)x, 1.(n+1)x\rangle_{n=0}^{\infty}\right].$$

In particular,

$$\int_0^{\infty} \frac{e^{-t}}{(1+tx)^a}\mathrm{d}t = {}_2F_0(a,1;;-x)$$

and the *complementary error function*

$$\begin{aligned}\mathsf{erfc}(x) &= 1 - \mathsf{erf}(x) = \frac{e^{-x^2}}{x\sqrt{\pi}} {}_2F_0\left(\tfrac{1}{2},1;;-x^{-2}\right) \\ &= \frac{e^{-x^2}}{x\sqrt{\pi}}\left[0.1, \left\langle 1.\tfrac{n+1}{2x^2}\right\rangle_{n=0}^{\infty}\right].\end{aligned}$$

Chapter 7

Effective Digital Representations

In this chapter, we consider the subset of exact digital representations that are effectively computable as understood in recursion theory.

7.1 Recursive Functions and Functionals

The study of recursive functions and functionals is intimately related to the study of the extended real numbers and functions respectively. Recall that an extended real number can be represented by a sequence of intervals $\mathbb{I}[\mathbb{F}]$ with end-points taken from a dense subset $\mathbb{F}$ of the extended real numbers $\mathbb{R}^{\infty}$. In other words, a function

$$f : \mathbb{N} \longrightarrow \mathbb{I}[\mathbb{F}]$$

from the natural numbers to the set of intervals $\mathbb{I}[\mathbb{F}]$. Also, recall that recursion theory only allows us to consider functions having domains and codomains with codings. In other words, only end-points sets $\mathbb{F}$ accompanied by a one-one effectively computable function to the natural numbers may be considered. This means that the end-points set must be countable at least. So, given an end-points set $\mathbb{F}$ and an associated coding function

$$\rho : \mathbb{I}[\mathbb{F}] \to \mathbb{N},$$

every real number can be represented by a function

$$\rho \circ f : \mathbb{N} \to \mathbb{N}$$

$$\mathbb{N} \xrightarrow{f} \mathbb{I}[\mathbb{F}] \xrightarrow{\rho} \mathbb{N}$$

from the natural numbers to the natural numbers. A real number is said to be effectively computable if it can be represented by a recursive function. The fact that the set of recursive functions is countable and the set of real numbers is uncountable implies that not all the extended real numbers can be effectively computable with respect to a given end-points set. The set of *computable real numbers* or *constructive real numbers* is classically considered to be the effectively computable real numbers with respect to the rational numbers. However, any real number can be considered to be effectively computable with respect to a suitable end-points set (e.g. x is effectively computable with respect to $x + \mathbb{Q}$).

In domain theory, a real number x is defined as computable with respect to $\mathbb{C}(\mathbb{Q}^\infty)$ if

$$\{\langle a, b\rangle \,|\, [a, b] \ll x\}$$

is recursively enumerable where $\langle , \rangle : \mathbb{Q}^\infty \times \mathbb{Q}^\infty \to \mathbb{N}$ is any suitable coding function [22]. This domain theoretic definition for a computable real number is equivalent to the classical one [22].

The set of closed intervals $\mathbb{I}^C[\mathbb{Q}^\infty]$ with end-points taken from the set of extended rational numbers $\mathbb{Q}^\infty$ is isomorphic to $(\mathbb{Q}^\infty \times \mathbb{Q}^\infty)_\perp$. This means that it is a particularly straightforward matter to construct a coding for $\mathbb{I}^C[\mathbb{Q}^\infty]$.

An n-ary real function $f : \mathbb{R}^n \to \mathbb{R}$ can be represented by a function from sequences of intervals to sequences of intervals; namely, a functional $F : (\mathbb{N} \to \mathbb{N})^n \to \mathbb{N} \to \mathbb{N}$ on the natural numbers [10, 13]. Classically, a real function is said to be effectively computable (with respect to a given end-points set $\mathbb{F}$) if it can be represented by a recursive functional. It follows that every effectively computable real function is continuous [77].

In domain theory, a real function $f : \mathbb{R}^\infty \to \mathbb{R}^\infty$ is defined as computable with respect to $\mathbb{C}(\mathbb{Q}^\infty)$ if there is a Scott continuous extension $g : \mathbb{C}(\mathbb{Q}^\infty) \to \mathbb{C}(\mathbb{Q}^\infty)$ such that

$$g(\{x\}) = \{f(x)\}$$

and

$$\{\langle a, b, c, d\rangle \,|\, [a, b] \ll g([c, d])\}$$

is recursively enumerable where $\langle , , , \rangle : \mathbb{Q}^\infty \times \mathbb{Q}^\infty \times \mathbb{Q}^\infty \times \mathbb{Q}^\infty \to \mathbb{N}$ is any coding function [22]. This domain theoretic definition for a computable real function is equivalent to the classical one [22].

A digital representation $(\mathbb{F}, \mathbb{D}, \theta)$ is effective if θ is an effectively computable functional and a coding function $\vartheta : \mathbb{D} \to \mathbb{N}$ exists for the digit set $\mathbb{D}$; an adequate digital representation $(\mathbb{F}, \mathbb{D}, \theta, \Phi)$ is effective if θ and Φ are effectively computable functionals and a coding function $\vartheta : \mathbb{D} \to \mathbb{N}$ exists for the digit set $\mathbb{D}$. It can be shown that the decimal representation is not an effectively adequate digital representation. For instance, consider the sequence of intervals $n \mapsto \left[1 - \frac{1}{n}, 1\right]$. No finite portion of this sequence contains sufficient information to decide even the first digit of its decimal expansion. It can be seen from the diagram

$$\begin{tikzcd}
\mathbb{N} \to \mathbb{I}[\mathbb{F}] \arrow[r, "F"] & \mathbb{N} \to \mathbb{I}[\mathbb{F}] \arrow[d, "\Phi"] \\
\mathbb{N} \to \mathbb{D} \arrow[u, "\Theta"] \arrow[r, "G"'] & \mathbb{N} \to \mathbb{D}.
\end{tikzcd}$$

that any effectively computable functional can be computed using an effectively adequate digital representation.

Definition 22 *A* redundant representation *is a digital representation in which every real number has more than one sequence of digits representing it.*

7.2 Redundant Positional Representations

The positional representations for the real numbers can be made redundant by extending the digit sets with negative counterparts. Redundancy ensures that the simplest mathematical operations are computable. For instance, multiplication by 3 is only computable with redundancy. For example, consider $3 \times \frac{1}{3}$. Note that, using the decimal representation, $\frac{1}{3}$ is represented by $0.333\cdots$ with the interpretation

$$[0,1] \supseteq [0.3, 0.4] \supseteq [0.33, 0.34] \supseteq \cdots.$$

Using the redundant decimal representation, $\frac{1}{3}$ can also be represented by $0.333\cdots$ but with the interpretation

$$[-1,1] \supseteq [0.2, 0.4] \supseteq [0.32, 0.34] \supseteq \cdots.$$

This difference ensures that for a given output precision, only a finite amount of input precision is required.

The redundant radix b positional representation is the signed incremental digit representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{aligned}
\mathbb{F} &= \mathbb{B}(b) \\
B &= [-1, 1] \\
\Sigma &= \mathbb{Z}
\end{aligned}$$

$$\begin{aligned}
\varphi &= \sigma \mapsto x \mapsto \sigma + x \\
\Delta &= \mathbb{Z}(b) \\
\psi &= d \mapsto x \mapsto \frac{d+x}{b} \\
\Omega &= \mathbb{Z}(b) \\
\phi &= \tau \mapsto \frac{\tau}{b}.
\end{aligned}$$

In particular, the *redundant binary representation* is given by the signed incremental digit representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{aligned}
\mathbb{F} &= \mathbb{B}(2) \\
B &= [-1, 1] \\
\Sigma &= \mathbb{Z} \\
\varphi &= \sigma \mapsto x \mapsto \sigma + x \\
\Delta &= \mathbb{Z}(2) \\
\psi &= d \mapsto x \mapsto \frac{d+x}{2} \\
\Omega &= \mathbb{Z}(2) \\
\phi &= \tau \mapsto \frac{\tau}{2}.
\end{aligned}$$

Example 23 *The finite sequence of symbols* 91011 *is a redundant binary representation for* $9\frac{11}{16}$. *The first symbol is called the sign, the last symbol is called the terminator and the other symbols are called digits and so let*

$$\begin{aligned}
\sigma &= 9 \in \Sigma \\
d_0 &= 1 \in \Delta \\
d_1 &= 0 \in \Delta \\
d_2 &= 1 \in \Delta \\
\tau &= 1 \in \Omega.
\end{aligned}$$

The finite sequence of intervals $I_0 I_1 I_2 I_3 I_4$ *induced by this representation is given by*

$$\begin{aligned}
I_0 &= \varphi(\sigma)(B) \\
&= \varphi(9)([-1, 1]) \\
&= [8, 10] \\
&= 9 + [-1, 1] \\
I_1 &= \varphi(\sigma)(\psi(d_0)(B))
\end{aligned}$$

$$\begin{aligned}
&= \varphi(9)(\psi(1)([-1,1])) \\
&= \varphi(9)([0,1]) \\
&= [9,10] \\
&= 9+\tfrac{1}{2}+\left[-\tfrac{1}{2},\tfrac{1}{2}\right] \\
I_2 &= \varphi(\sigma)(\psi(d_0)(\psi(d_1)(B))) \\
&= \varphi(9)(\psi(1)(\psi(0)(B))) \\
&= \varphi(9)\left(\psi(1)\left(\left[-\tfrac{1}{2},\tfrac{1}{2}\right]\right)\right) \\
&= \varphi(9)\left(\left[\tfrac{1}{4},\tfrac{3}{4}\right]\right) \\
&= \left[\tfrac{37}{4},\tfrac{39}{4}\right] \\
&= 9+\tfrac{1}{2}+\tfrac{0}{4}+\left[-\tfrac{1}{4},\tfrac{1}{4}\right] \\
I_3 &= \varphi(\sigma)(\psi(d_0)(\psi(d_1)(\psi(d_2)(B)))) \\
&= \varphi(9)(\psi(1)(\psi(0)(\psi(1)([-1,1])))) \\
&= \varphi(9)(\psi(1)(\psi(0)([0,1]))) \\
&= \varphi(9)\left(\psi(1)\left(\left[0,\tfrac{1}{2}\right]\right)\right) \\
&= \varphi(9)\left(\left[\tfrac{1}{2},\tfrac{3}{4}\right]\right) \\
&= \left[\tfrac{19}{2},\tfrac{39}{4}\right] \\
&= 9+\tfrac{1}{2}+\tfrac{0}{4}+\tfrac{1}{8}+\left[-\tfrac{1}{8},\tfrac{1}{8}\right] \\
I_4 &= \varphi(\sigma)(\psi(d_0)(\psi(d_1)(\psi(d_2)(\phi(\{\tau\}))))) \\
&= \varphi(9)(\psi(1)(\psi(0)(\psi(1)(\phi(\{1\}))))) \\
&= \varphi(9)\left(\psi(1)\left(\psi(0)\left(\psi(1)\left(\left\{\tfrac{1}{2}\right\}\right)\right)\right)\right) \\
&= \varphi(9)\left(\psi(1)\left(\psi(0)\left(\left\{\tfrac{3}{4}\right\}\right)\right)\right) \\
&= \varphi(9)\left(\psi(1)\left(\left\{\tfrac{3}{8}\right\}\right)\right) \\
&= \varphi(9)\left\{\tfrac{11}{16}\right\} \\
&= \left\{\tfrac{155}{16}\right\} \\
&= 9+\tfrac{1}{2}+\tfrac{0}{4}+\tfrac{1}{8}+\left\{\tfrac{1}{16}\right\}.
\end{aligned}$$

The redundant positional representation turns out to be an effectively adequate digital representation for the real numbers. Interestingly, the redundant binary representation was first introduced by Avizienis [2] in 1961 as a means to avoid carry propagation in the addition and subtraction of integers rather than as an effective means to represent exact real numbers.

Let t list denote the list type of t (i.e. a sequence of values with the same type t) and let : denote the infix constructor for building a list by attaching a new member to the front.

The algorithm for the scaled addition of two redundant positional expansions with radix $r \geq 3$ is well known [3]; namely

$$\begin{aligned}\mathsf{add} &: \mathbb{Z}(r)\ \mathsf{list} \times \mathbb{Z}(r)\ \mathsf{list} \to \mathbb{Z}(r)\ \mathsf{list} \\ \mathsf{add}(\alpha, \beta) &= \mathsf{add}'(\alpha, \beta, 0)\end{aligned}$$

where

$$\begin{aligned}\mathsf{add}' &: \mathbb{Z}(r)\ \mathsf{list} \times \mathbb{Z}(r)\ \mathsf{list} \times \mathbb{Z}(r) \to \mathbb{Z}(r)\ \mathsf{list} \\ \mathsf{add}'(a : \alpha, b : \beta, c) &= \begin{cases} c + 1 : \mathsf{add}'(\alpha, \beta, a + b - r) & \text{if } a + b \geq r - 1 \\ c - 1 : \mathsf{add}'(\alpha, \beta, a + b + r) & \text{if } a + b \leq 1 - r \\ c : \mathsf{add}'(\alpha, \beta, a + b) & \text{otherwise.} \end{cases}\end{aligned}$$

The interpretation is

$$\begin{aligned}[\![a : \alpha]\!] &= \frac{a + [\![\alpha]\!]}{r} \\ [\![\mathsf{add}(\alpha, \beta)]\!] &= \frac{[\![\alpha]\!] + [\![\beta]\!]}{r} \\ [\![\mathsf{add}'(\alpha, \beta, c)]\!] &= \frac{[\![\alpha]\!] + [\![\beta]\!] + c}{r}.\end{aligned}$$

The binary version of scaled addition (i.e. mean) is more tricky.

Proposition 24 *An algorithm for the mean of two redundant binary expansions is*

$$\begin{aligned}\mathsf{mean} &: \mathbb{Z}(2)\ \mathsf{list} \times \mathbb{Z}(2)\ \mathsf{list} \to \mathbb{Z}(2)\ \mathsf{list} \\ &\quad \mathsf{mean}(a : \alpha, b : \beta) \\ &= \begin{cases} \mathsf{div}(c, 2) : \mathsf{mean}(\alpha, \beta) & \textit{if } \mathsf{mod}(c, 2) = 0 \\ \mathsf{mean}'(\alpha, \beta, c) & \textit{otherwise} \end{cases}\end{aligned} \tag{7.1}$$

where

$$c = a + b$$

and

$$\begin{aligned}\mathsf{mean}' &: \mathbb{Z}(2)\ \mathsf{list} \times \mathbb{Z}(2)\ \mathsf{list} \times \mathbb{Z}(2) \to \mathbb{Z}(2)\ \mathsf{list} \\ &\quad \mathsf{mean}'(a : \alpha, b : \beta, c) \\ &= \begin{cases} c : \mathsf{mean}'(\alpha, \beta, -c) & \textit{if } a + b = c \\ h : \mathsf{div}(t, 2) : \mathsf{mean}(\alpha, \beta) & \textit{if } \mathsf{mod}(d, 2) = 0 \\ h : \mathsf{mean}'(\alpha, \beta, t) & \textit{otherwise} \end{cases}\end{aligned} \tag{7.2}$$

where

$$\begin{aligned} d &= a + b + 2c \\ h &= \mathsf{div}\,(d, 4) \\ t &= \mathsf{mod}\,(d, 4) \end{aligned}$$

where $\mathsf{div} : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ *and* $\mathsf{mod} : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ *are integer division and remainder respectively satisfying*

$$m \times \mathsf{div}\,(n, m) + \mathsf{mod}\,(n, m) = n.$$

The interpretation is

$$\begin{aligned} [\![a : \alpha]\!] &= \frac{a + [\![\alpha]\!]}{2} \\ [\![\mathsf{mean}\,(\alpha, \beta)]\!] &= \frac{[\![\alpha]\!] + [\![\beta]\!]}{2} \\ [\![\mathsf{mean}'\,(\alpha, \beta, c)]\!] &= \frac{[\![\alpha]\!] + [\![\beta]\!] + c}{4}. \end{aligned}$$

Proof : Consider equation (7.1).

$$\begin{aligned} [\![\mathsf{mean}\,(a : \alpha, b : \beta)]\!] &= \frac{\left(\dfrac{a + [\![\alpha]\!]}{2}\right) + \left(\dfrac{b + [\![\beta]\!]}{2}\right)}{2} \\ &= \frac{[\![\alpha]\!] + [\![\beta]\!] + (a + b)}{4} \end{aligned}$$

If we assume $\mathsf{mod}\,(c, 2) = 0$ then

$$\begin{aligned} [\![\mathsf{div}\,(c, 2) : \mathsf{mean}\,(\alpha, \beta)]\!] &= \frac{\left(\dfrac{a + b}{2}\right) + \left(\dfrac{[\![\alpha]\!] + [\![\beta]\!]}{2}\right)}{2} \\ &= \frac{[\![\alpha]\!] + [\![\beta]\!] + (a + b)}{4}. \end{aligned}$$

Otherwise we must assume $\mathsf{mod}\,(c, 2) \neq 0$ and so

$$[\![\mathsf{mean}'\,(\alpha, \beta, c)]\!] = \frac{[\![\alpha]\!] + [\![\beta]\!] + (a + b)}{4}.$$

Now consider equation (7.2).

$$\begin{aligned} [\![\mathsf{mean}'\,(a : \alpha, b : \beta, c)]\!] &= \frac{\left(\dfrac{a + [\![\alpha]\!]}{2}\right) + \left(\dfrac{b + [\![\beta]\!]}{2}\right) + c}{4} \\ &= \frac{[\![\alpha]\!] + [\![\beta]\!] + (a + b + 2c)}{8} \end{aligned}$$

If we assume $a+b=c$ then

$$\begin{aligned}
[\![c:\mathsf{mean}'(\alpha,\beta,-c)]\!] &= \frac{c+\left(\dfrac{[\![\alpha]\!]+[\![\beta]\!]-c}{4}\right)}{2} \\
&= \frac{[\![\alpha]\!]+[\![\beta]\!]+(a+b+2c)}{8}.
\end{aligned}$$

Alternatively, if we assume $a+b\neq c$ and $\mathsf{mod}(d,2)=0$ then

$$\begin{aligned}
[\![h:\mathsf{div}(t,2):\mathsf{mean}(\alpha,\beta)]\!] &= \frac{\mathsf{div}(d,4)+\left(\dfrac{\dfrac{\mathsf{mod}(d,4)}{2}+\left(\dfrac{[\![\alpha]\!]+[\![\beta]\!]}{2}\right)}{2}\right)}{2} \\
&= \frac{[\![\alpha]\!]+[\![\beta]\!]+4\times\mathsf{div}(d,4)+\mathsf{mod}(d,4)}{8} \\
&= \frac{[\![\alpha]\!]+[\![\beta]\!]+(a+b+2c)}{8}.
\end{aligned}$$

Otherwise, we must assume $a+b\neq c$ and $\mathsf{mod}(d,2)\neq 0$ and so

$$\begin{aligned}
[\![h:\mathsf{mean}'(\alpha,\beta,t)]\!] &= \frac{\mathsf{div}(d,4)+\left(\dfrac{[\![\alpha]\!]+[\![\beta]\!]+\mathsf{mod}(d,4)}{4}\right)}{2} \\
&= \frac{[\![\alpha]\!]+[\![\beta]\!]+4\times\mathsf{div}(d,4)+\mathsf{mod}(d,4)}{8} \\
&= \frac{[\![\alpha]\!]+[\![\beta]\!]+(a+b+2c)}{8}. \blacksquare
\end{aligned}$$

Proposition 25 *An algorithm for the comparison of two redundant binary expansions*

$$x \lesssim y = \begin{cases} \mathsf{true} & \textit{if } x<y \\ \mathsf{false} & \textit{if } x>y \end{cases}$$

is

$$\mathsf{less}:\mathbb{Z}(2)\ \mathsf{list}\times\mathbb{Z}(2)\ \mathsf{list}\times\mathbb{Z}(3)\to\mathsf{boolean}$$

$$\mathsf{less}(a:\alpha,b:\beta,c) = \begin{cases} \mathsf{true} & \textit{if } d<-2 \\ \mathsf{false} & \textit{if } d>2 \\ \mathsf{less}(\alpha,\beta,d) & \textit{otherwise} \end{cases} \tag{7.3}$$

where

$$d=2c+a-b.$$

The interpretation is

$$[\![a : \alpha]\!] = \frac{a + [\![\alpha]\!]}{2}$$

$$[\![\mathsf{less}\,(\alpha, \beta, c)]\!] = ([\![\alpha]\!] + c) \lesssim [\![\beta]\!].$$

Proof : Consider equation (7.3).

$$\begin{aligned}
[\![\mathsf{less}\,(a : \alpha, b : \beta, c)]\!] &= \left(\frac{a + [\![\alpha]\!]}{2} + c\right) \lesssim \frac{b + [\![\beta]\!]}{2} \\
&= (a + [\![\alpha]\!] + 2c) \lesssim (b + [\![\beta]\!]) \\
&= ([\![\alpha]\!] + d) \lesssim [\![\beta]\!] \\
&= \begin{cases} \mathsf{true} & \text{if } d < -2 \\ \mathsf{false} & \text{if } d > 2 \\ [\![\mathsf{less}\,(\alpha, \beta, d)]\!] & \text{otherwise.} \end{cases} \blacksquare
\end{aligned}$$

7.3 Redundant Continued Fractions

$\mathbb{N}$-fractions and $\mathbb{Z}$-fractions are not redundant because the floor and round operations are not effectively computable. However, the Euclidean part of a real number x, which is defined by Vuillemin [71] as any integer $\lfloor\!\lfloor x \rceil\!\rceil$ such that $d_{\mathrm{C}}\,(x, \lfloor\!\lfloor x \rceil\!\rceil) < 1$, is effectively computable. So, the $\mathbb{E}$-fraction of a real number x is the simple continued fraction $[a_0, a_1, a_2, \ldots]$ with a_n recursively given by

$$\begin{aligned}
(a_0, r_0) &= (\lfloor\!\lfloor x \rceil\!\rceil, x - \lfloor\!\lfloor x \rceil\!\rceil) \\
(a_{n+1}, r_{n+1}) &= \left(\lfloor\!\lfloor \frac{1}{r_n} \rceil\!\rceil, \frac{1}{r_n} - \lfloor\!\lfloor \frac{1}{r_n} \rceil\!\rceil\right).
\end{aligned}$$

It has been shown by Lester [45] that the continuation function $\xi_{\mathbb{E}}$ for $\mathbb{E}$-fractions (see definition 19) is

$$\xi_{\mathbb{E}}\,(n) = \left(\frac{-4n + (n^2+1)\sqrt{3}}{n^2 - 3}, \frac{-4n - (n^2+1)\sqrt{3}}{n^2 - 3}\right)$$

because

$$d_{\mathrm{C}}\,(x, \lfloor\!\lfloor x \rceil\!\rceil) < 1 \Leftrightarrow x \in \xi_{\mathbb{E}}\,(\lfloor\!\lfloor x \rceil\!\rceil).$$

On careful inspection, this expression can be simplified to a pair of Möbius transformations

$$\xi_{\mathbb{E}}\,(n) = \left(\frac{n\sqrt{3} - 1}{\sqrt{3} + n}, \frac{n\sqrt{3} + 1}{\sqrt{3} - n}\right).$$

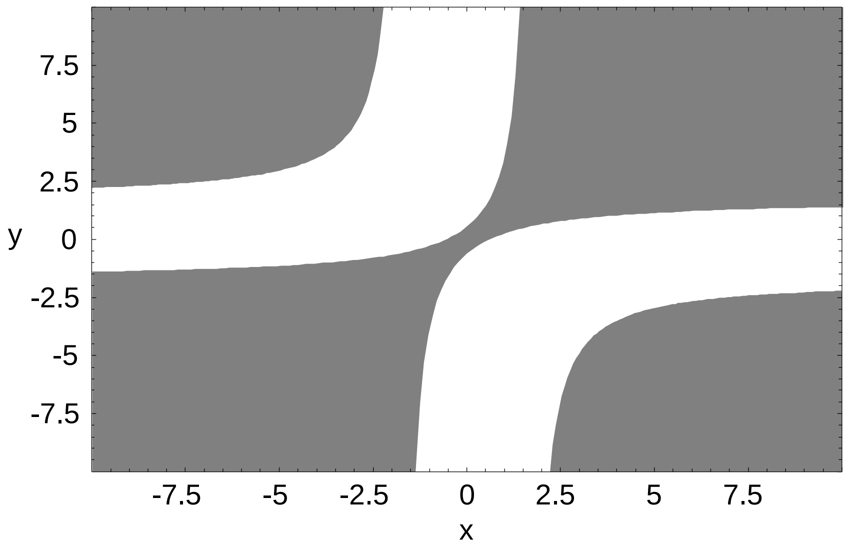


Figure 7.1: The shaded region is a plot of the points (x, y) such that $d_{\mathrm{C}}(x, y) < 1$.

The shaded region in Figure (7.1) shows a plot of the points (x, y) such that $d_{\mathrm{C}}(x, y) < 1$. So, a general $\mathbb{E}$-fraction $[a_0, a_1, a_2, \ldots]$ induces the sequence of intervals $\langle I_n \rangle_{n=0}^{\infty}$

$$\begin{aligned}
I_0 &= \xi_{\mathbb{E}}(a_0) \\
I_1 &= a_0 + \frac{1}{\xi_{\mathbb{E}}(a_1)} \\
I_2 &= a_0 + \frac{1}{a_1 + \frac{1}{\xi_{\mathbb{E}}(a_2)}} \\
I_3 &= a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\xi_{\mathbb{E}}(a_3)}}} \\
\vdots &= \vdots
\end{aligned}$$

Unfortunately, due to the inherent redundancy of this representation, there is no guarantee that this sequence is nested. This is perhaps, why Vuillemin goes on to consider the continuation function

$$\xi(n) = \begin{cases} (-2, 2) & \text{if } n = 0 \\ \left(|n| - \frac{1}{2}, \frac{1}{2} - |n|\right) & \text{if } n \neq 0. \end{cases}$$

So, what might a good continuation function be? Firstly, every extended real number needs to be covered

$$\bigcup_{n\in\mathbb{Z}} \xi(n) = \mathbb{R}^{\infty}. \tag{7.4}$$

otherwise we have real numbers that cannot be represented. Secondly, it would be convenient from a computational efficiency point of view if we could restrict ourselves to nested intervals

$$\bigcup_{m\in\mathbb{Z}} \left\{ n + \frac{1}{\xi(m)} \middle| n + \frac{1}{\xi(m)} \subset \xi(n) \right\} = \xi(n). \tag{7.5}$$

Unfortunately, the redefined continuation function does not satisfy these conditions. The problem can be illustrated with an example. Consider the $\mathbb{E}$-fraction for $\frac{5}{9}$.

$$\begin{array}{rcl} x & = & \frac{5}{9} \in \xi(1) = \left(\frac{1}{2}, -\frac{1}{2}\right) \\ (a_0, r_0) & = & \left(1, -\frac{4}{9}\right) \\ \xi(a_0) & = & \left(\frac{1}{2}, -\frac{1}{2}\right) \\ \frac{1}{r_0} & = & -\frac{9}{4} \in \xi(z) \text{ for } z \in \mathbb{Z}(3) \\ a_0 + \frac{1}{\xi(0)} & = & \left(\frac{3}{2}, \frac{1}{2}\right) \not\subset \xi(a_0) \\ a_0 + \frac{1}{\xi(\pm 1)} & = & (-1, 3) \not\subset \xi(a_0) \\ a_0 + \frac{1}{\xi(\pm 2)} & = & \left(\frac{1}{3}, \frac{5}{2}\right) \not\subset \xi(a_0) \end{array}$$

However, the simple continuation function

$$\xi_{\mathbb{P}}(n) = [n, \infty]$$

does satisfy equations (7.4)

$$\bigcup_{n\in\mathbb{Z}} \xi_{\mathbb{P}}(n) = \bigcup_{n\in\mathbb{Z}} [n, \infty] = \mathbb{R}^{\infty}$$

and (7.5)

$$\bigcup_{m\in\mathbb{Z}} \left\{ n + \frac{1}{\xi_{\mathbb{P}}(m)} \middle| n + \frac{1}{\xi_{\mathbb{P}}(m)} \subset \xi_{\mathbb{P}}(n) \right\} = \bigcup_{m\in\mathbb{N}} \left[n, n + \frac{1}{m} \right] = [n, \infty].$$

for all $n \in \mathbb{Z}$. We will call a continued fraction constructed using this continuation function, a $\mathbb{P}$-fraction. So, the $\mathbb{P}$-fraction of a real number x is the simple continued fraction $[a_0, a_1, a_2, \ldots]$ with a_n recursively given by

$$\begin{aligned}(a_0, r_0) &= (\lfloor\lfloor x \rfloor\rfloor, x - \lfloor\lfloor x \rfloor\rfloor) \\ (a_{n+1}, r_{n+1}) &= \left(\lfloor\lfloor \frac{1}{r_n} \rfloor\rfloor, \frac{1}{r_n} - \lfloor\lfloor \frac{1}{r_n} \rfloor\rfloor\right)\end{aligned}$$

where $\lfloor\lfloor x \rfloor\rfloor$ is defined as any integer less than or equal to x, which we will call the *basement* of x. Observe that

$$0 \leq y - \lfloor\lfloor y \rfloor\rfloor \leq \infty$$

for any extended real number y. Therefore

$$0 \leq r_n \leq \infty \Rightarrow r_n \in [0, \infty] \Rightarrow \frac{1}{r_n} \in [0, \infty] .$$

It follows that the signed incremental digit representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{aligned}\mathbb{F} &= \mathbb{Q} \\ B &= [0, \infty] \\ \Sigma &= \mathbb{Z} \\ \varphi &= \sigma \mapsto x \mapsto \sigma + \frac{1}{x} \\ \Delta &= \mathbb{N} \\ \psi &= d \mapsto x \mapsto d + \frac{1}{x} \\ \Omega &= \mathbb{N} \\ \phi &= \tau \mapsto \tau\end{aligned}$$

faithfully represents $\mathbb{P}$-fractions.

7.4 Incremental Floating Point

In 1983, Watanuki and Ercegovac [74] explored the possibilities of redundant radix b floating point. Let us extend their definitions from finite to infinite sequences of digits.

Definition 26 *The* ***incremental mantissa exact radix*** b ***floating point representation*** *of an extended real number, x say, is an infinite sequence of digits $\langle d_n \rangle_{n=0}^{\infty}$ with the interpretation*

$$x = (0.d_1 d_2 d_3 \ldots)_b \times b^{d_0}$$

where the first digit (the exponent) is an arbitrary integer and the other digits (the mantissa) are restricted to $\mathbb{Z}(b)$. *The representation is referred to as* $\gamma-$***normalized*** *if*

$$\frac{1}{b^{\gamma}} \leq (0.d_1 d_2 d_3 \ldots)_b \leq 1.$$

In particular, the representation is γ-normalized if $(d_1 d_2 \ldots d_\gamma)_b \in \mathbb{Z}(b^\gamma) - \mathbb{Z}(2)$. Therefore, the signed incremental digit representation $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ where

$$\begin{aligned}
\mathbb{F} &= \mathbb{B}(b) \\
B &= [-1, 1] \\
\Sigma &= \mathbb{Z}\times(\mathbb{Z}(b^\gamma) - \mathbb{Z}(2)) \\
\varphi &= (e, \sigma) \mapsto x \mapsto \frac{\sigma + x}{b^\gamma} \times b^e \\
\Delta &= \mathbb{Z}(b) \\
\psi &= d \mapsto x \mapsto \frac{d + x}{b} \\
\Omega &= \mathbb{Z}(b) - \{0\} \\
\phi &= \tau \mapsto \frac{\tau}{b}
\end{aligned}$$

corresponds to the incremental mantissa $\gamma-$normalized exact radix b floating point representation.

Example 27 *The signed incremental digit representation* $(\mathbb{F}, B, \Sigma, \varphi, \Delta, \psi, \Omega, \phi)$ *where*

$$\begin{aligned}
\mathbb{F} &= \mathbb{B}(2) \\
B &= [-1, 1] \\
\Sigma &= \mathbb{Z}\times\{-3, -2, 2, 3\} \\
\varphi &= (e, \sigma) \mapsto x \mapsto (\sigma + x) \times 2^{e-2} \\
\Delta &= \{-1, 0, 1\} \\
\psi &= d \mapsto x \mapsto \frac{d + x}{2} \\
\Omega &= \{-1, 1\} \\
\phi &= \tau \mapsto \frac{\tau}{2}
\end{aligned}$$

corresponds to the incremental mantissa $2-$*normalized exact radix* 2 *floating point representation. The finite sequence of symbols* $(5, 2)(-1)01$ *represents* 13. *The*

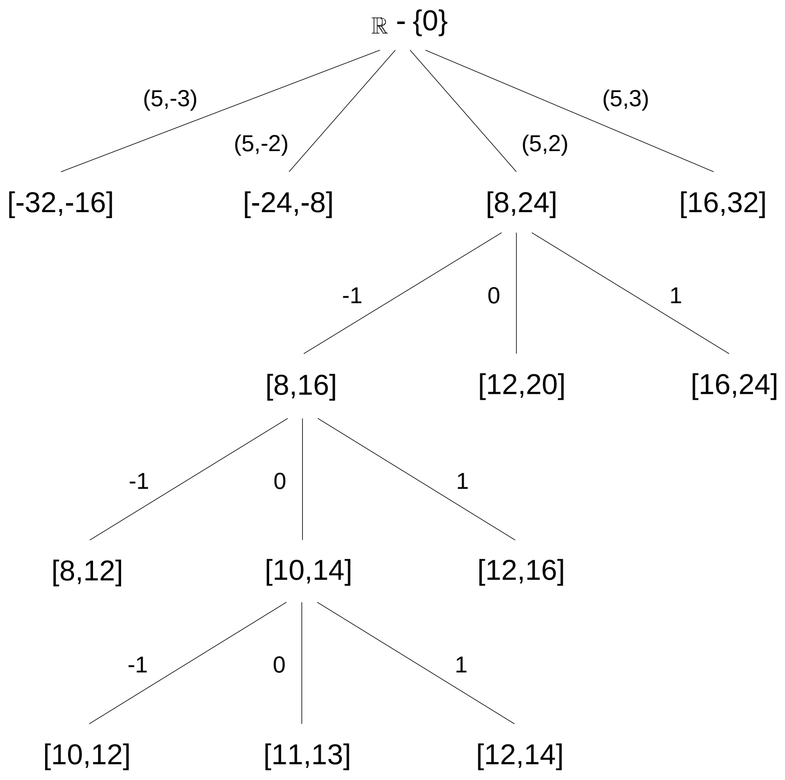


Figure 7.2: An illustration of the incremental mantissa 2−normalized exact radix 2 floating point representation.

first symbol is called the sign, the last symbol is called the terminator and the other symbols are called digits and so let

$$\begin{aligned}(e,\sigma) &= (5,2)\in\Sigma\\ d_0 &= -1\in\Delta\\ d_1 &= 0\in\Delta\\ \tau &= 1\in\Omega.\end{aligned}$$

The finite sequence of intervals $I_0I_1I_2I_3$ induced by this representation is given by

$$\begin{aligned}I_0 &= \varphi(e,\sigma)(B)\\ &= \varphi(5,2)([-1,1])\\ &= [8,24]\\ I_1 &= \varphi(e,\sigma)(\psi(d_0)(B))\\ &= \varphi(5,2)(\psi(-1)([-1,1]))\\ &= \varphi(5,2)([-1,0])\\ &= [8,16]\\ I_2 &= \varphi(e,\sigma)(\psi(d_0)(\psi(d_1)(B)))\\ &= \varphi(5,2)(\psi(-1)(\psi(0)([-1,1])))\\ &= \varphi(5,2)\left(\psi(-1)\left(\left[-\tfrac{1}{2},\tfrac{1}{2}\right]\right)\right)\\ &= \varphi(5,2)\left(\left[-\tfrac{3}{4},-\tfrac{1}{4}\right]\right)\\ &= [10,14]\\ I_4 &= \varphi(e,\sigma)(\psi(d_0)(\psi(d_1)(\phi(\tau))))\\ &= \varphi(5,2)(\psi(-1)(\psi(0)(\phi(\{1\}))))\\ &= \varphi(5,2)\left(\psi(-1)\left(\psi(0)\left(\left\{\tfrac{1}{2}\right\}\right)\right)\right)\\ &= \varphi(5,2)\left(\psi(-1)\left(\left\{\tfrac{1}{4}\right\}\right)\right)\\ &= \varphi(5,2)\left(\left\{-\tfrac{3}{8}\right\}\right)\\ &= \{13\}\end{aligned}$$

and illustrated in figure (7.2).

Note that $\gamma = 1$ for $b = 2$ is not allowed because the sign set would be empty. Also, note that this representation cannot handle 0 and ∞. This is essentially because the exponent is not incremental. Incrementality in the exponent can be achieved by replacing the integer e with a sequence of $(1+e)$ successor digits s if $e \geq 0$ and a sequence of $(1-e)$ predecessor digits p if $e \leq 0$ as pointed out by Nielsen and Kornerup [51] in 1995. This gives a fully incremental representation

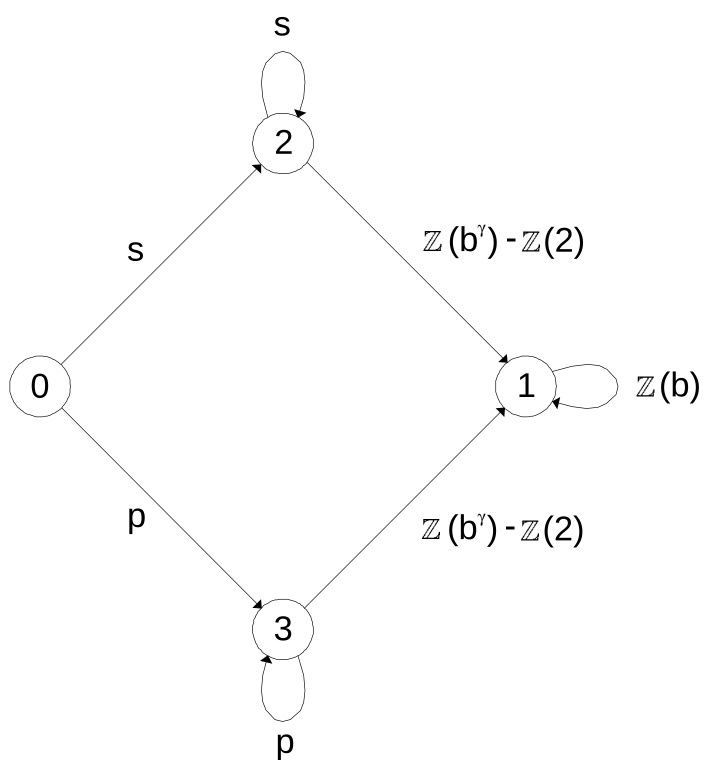


Figure 7.3: The automaton for the fully incremental γ-normalized exact radix b floating point representation.

corresponding to the digital representation $\left(\mathbb{B}\left(b\right), \{s,p\} \cup \mathbb{Z}\left(b^{\gamma}\right), \theta\right)$ where

$$\theta\left(i \mapsto d_i\right)\left(n\right) = \begin{cases} [1,-1] \times b^{n-1} & \text{if } d_0,\ldots,d_n = s \\ [-1,1] \times b^{1-n} & \text{if } d_0,\ldots,d_n = p \\ & \text{if } d_0,\ldots,d_{e-1} = s \\ m \times b^{e-\gamma} & \text{and } d_e \in \mathbb{Z}\left(b^{\gamma}\right) - \mathbb{Z}\left(2\right) \\ & \text{and } d_{e+1},\ldots,d_n \in \mathbb{Z}\left(b\right) \\ & \text{if } d_0,\ldots,d_{e-1} = p \\ m \times b^{-e} & \text{and } d_e \in \mathbb{Z}\left(b^{\gamma}\right) - \mathbb{Z}\left(2\right) \\ & \text{and } d_{e+1},\ldots,d_n \in \mathbb{Z}\left(b\right) \end{cases}$$

with

$$m = d_e + \left(0.d_{e+1}d_{e+2}\ldots d_n\right)_b + \frac{[-1,1]}{b^{n-e}}.$$

The corresponding automaton is illustrated in figure (7.3).

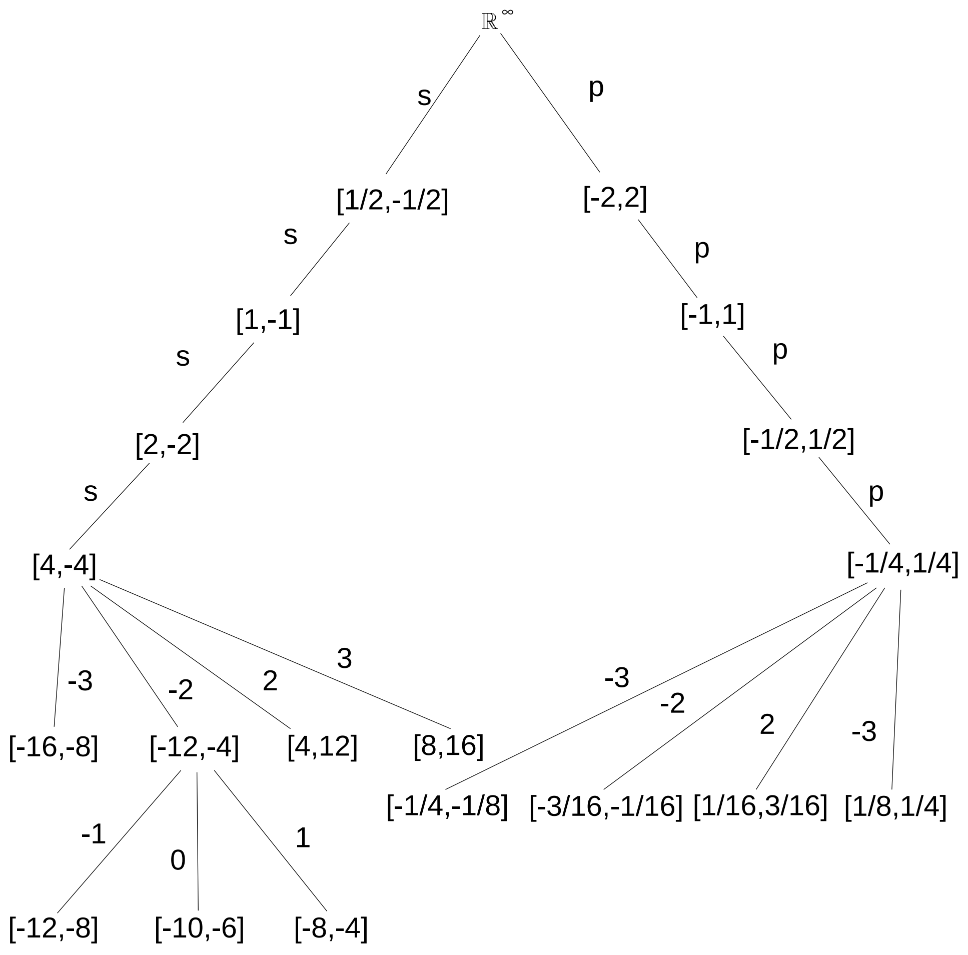


Figure 7.4: An illustration of the fully incremental 2-normalized exact radix 2 floating point representation.

7.5 Redundant If Operator

The basic arithmetic operations are computable only for *redundant* representations of the real numbers [49, 46, 8, 69]. In effect, non-determinism is being traded for computability. The basic predicates, such as equality and comparison, are not computable at all. However, the way-below relation $\ll$ on the continuous domain of intervals is recursively enumerable. Consequently, the if operator defined by

$$\begin{aligned} \mathsf{if} \quad &: \quad K \times \mathbb{F} \times (K \to t)^2 \to t \\ \mathsf{if}\, x \quad &< \quad q\,\mathsf{then}\, f\,\mathsf{else}\, g = \begin{cases} f(x) & \text{if } x < q \\ g(x) & \text{if } x \geq q \end{cases} \end{aligned} \tag{7.6}$$

where $K \in \mathbb{I}^{\mathrm{C}}\mathbb{R}$ and $\mathbb{F}$ is a dense subset of K, should be replaced by the *redundant if* operator

$$\begin{aligned} \mathsf{rif} \quad &: \quad \mathbb{I}^{\mathrm{C}}[K] \times \mathbb{I}^{\mathrm{C}}[\mathbb{F}]^2 \times \left(\mathbb{I}^{\mathrm{C}}[K] \to t\right)^2 \to t \\ \mathsf{rif}\, x \quad &< \quad (I, J)\ \mathsf{then}\, f\,\mathsf{else}\, g = \begin{cases} f(x) & \text{if } I \ll x \\ g(x) & \text{if } J \ll x \end{cases} \end{aligned} \tag{7.7}$$

where $K \in \mathbb{I}^{\mathrm{C}}\mathbb{R}^{\infty}$ and $\mathbb{F}$ is a dense subset of K. As it stands, the redundant if operator is a partial multivalued function. However, the side condition

$$\mathsf{interior}(I) \cup \mathsf{interior}(J) = \mathsf{interior}(K) \tag{7.8}$$

ensures that it is a total function and the side condition

$$f(\{x\}) = g(\{x\}) \text{ if } I \ll \{x\} \text{ and } J \ll \{x\} \tag{7.9}$$

ensures that it is single valued for the maximal elements. In practice, the redundant if operator as defined in equation (7.7) is still not useful because f and g are frequently only defined over I and J respectively; as is the case with analytic continuations over the real line. This usually means the insertion of some domain restricting functions between f and it's argument x and g and it's argument x.

The redundant if operator is similar to the *quasi-relational comparison* operator $<_\epsilon$ introduced by Boehm and Cartwright [6, 5]

$$x <_\epsilon y = \begin{cases} \mathsf{true} & \text{if } x < y - \epsilon \\ \mathsf{false} & \text{if } x > y + \epsilon \\ \text{either } \mathsf{true} \text{ or } \mathsf{false} & \text{otherwise.} \end{cases}$$

The redundant if operator will be used implicitly in statements like "if $f(d)$ for some $d \in D$". This notion will be used for the transcendental functions by utilizing the idea of analytic continuation in complex analysis as described in section 3.

The redundant if operator overcomes the problem of undecidability because one of the if conditions is satisfied in finite time. Therefore, the redundant if operator makes comparison feasible even when applied recursively. This contrasts with the parallel if operator [5, 10, 23]

$$\mathsf{pif}\, x < 0\, \mathsf{then}\, f(x)\ \mathsf{else}\, g(x) = \begin{cases} f(x) & \text{if } x < 0 \\ f(x) \sqcap g(x) & \text{if } x = 0 \\ g(x) & \text{if } x > 0. \end{cases}$$

In particular, recursive application of the parallel if operator may result in an exponential growth in resource usage.

Chapter 8

Linear Fractional Transformations

In this chapter, we explore the mathematical properties and representations of the extended real numbers together with various important classes of rational functions including Möbius transformations.

8.1 Vectors, Matrices and Tensors

For the sake of this thesis, a *vector* is a pair of integers, a *matrix* is a pair of vectors and a *tensor* is a pair of matrices. Therefore, the set of all vectors $\mathbb{V}$, the set of all matrices $\mathbb{M}$ and the set of all tensors $\mathbb{T}$ are given by

$$\begin{aligned} \mathbb{V} &= \mathbb{Z} \times \mathbb{Z} \\ \mathbb{M} &= \mathbb{V} \times \mathbb{V} \\ \mathbb{T} &= \mathbb{M} \times \mathbb{M} \end{aligned}$$

and for convenience

$$\begin{aligned} \mathbb{V}^* &= \mathbb{N} \times \mathbb{N} \\ \mathbb{M}^* &= \mathbb{V}^* \times \mathbb{V}^* \\ \mathbb{T}^* &= \mathbb{M}^* \times \mathbb{M}^*. \end{aligned}$$

The symbols V (and W), M (and N and O) and T (and U) will be used to denote vectors, matrices and tensors respectively. The vector (a, b) will be denoted by

$$\begin{pmatrix} a \\ b \end{pmatrix} \text{ and } \begin{array}{c} a \\ \Big| \\ b. \end{array}$$

The matrix $((a,b),(c,d))$ will be denoted by

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} \text{ and } \begin{tikzcd} a \arrow[r, dash] \arrow[d, dash] & c \arrow[d, dash] \\ b \arrow[r, dash] & d. \end{tikzcd}$$

The tensor $(((a,b),(c,d)),((e,f),(g,h)))$ will be denoted by

$$\begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix} \text{ and } \begin{tikzcd} a \arrow[rr, dash] \arrow[dr, dash] \arrow[dd, dash] & & e \arrow[dr, dash] \arrow[dd, dash] & \\ & c \arrow[rr, dash] \arrow[dd, dash] & & g \arrow[dd, dash] \\ b \arrow[rr, dash] \arrow[dr, dash] & & f \arrow[dr, dash] & \\ & d \arrow[rr, dash] & & h. \end{tikzcd}$$

A pair with the subscript 0 (1) denotes its first (second) projection respectively. In other words

$$\begin{aligned} (X,Y)_0 &= X \\ (X,Y)_1 &= Y. \end{aligned}$$

Let the *transpose* of a matrix and tensor be defined by

$$\begin{aligned} \left(M^{\mathsf{T}}\right)_{ij} &= M_{ji} \\ \left(T^{\mathsf{T}}\right)_{ijk} &= T_{jik}. \end{aligned}$$

Lemma 28

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} a & e & c & g \\ b & f & d & h \end{pmatrix}$$

Proof:

$$\left(\begin{pmatrix} T_{000} & T_{010} & T_{100} & T_{110} \\ T_{001} & T_{011} & T_{101} & T_{111} \end{pmatrix}^{\mathsf{T}}\right)_{ijk} = \begin{pmatrix} T_{000} & T_{010} & T_{100} & T_{110} \\ T_{001} & T_{011} & T_{101} & T_{111} \end{pmatrix}_{jik}$$

$$\begin{aligned}
&= (((T_{000}, T_{001}), (T_{010}, T_{011})), \\
&\quad ((T_{100}, T_{101}), (T_{110}, T_{111})))_{jik} \\
&= ((T_{j00}, T_{j01}), (T_{j10}, T_{j11}))_{ik} \\
&= (T_{ji0}, T_{ji1})_k \\
&= ((T_{0i0}, T_{0i1}), (T_{1i0}, T_{1i1}))_{jk} \\
&= (((T_{000}, T_{001}), (T_{100}, T_{101})), \\
&\quad ((T_{010}, T_{011}), (T_{110}, T_{111})))_{ijk} \\
&= \begin{pmatrix} T_{000} & T_{100} & T_{010} & T_{110} \\ T_{001} & T_{101} & T_{011} & T_{111} \end{pmatrix}_{ijk} \quad \blacksquare
\end{aligned}$$

The symbol $\bullet$ will be used to indicate the usual dot product between vectors, matrices and tensors.

Lemma 29

$$(M \bullet V)_i = \sum_j V_j M_{ji}$$

$$(M \bullet N)_{ij} = \sum_k N_{ik} M_{kj}$$

$$(M \bullet T)_{ijk} = \sum_l T_{ijl} M_{lk}$$

Proof:

$$\begin{aligned}
(M \bullet V)_i &= \left(\begin{pmatrix} M_{00} & M_{10} \\ M_{01} & M_{11} \end{pmatrix} \bullet \begin{pmatrix} V_0 \\ V_1 \end{pmatrix} \right)_i \\
&= \begin{pmatrix} M_{00}V_0 + M_{10}V_1 \\ M_{01}V_0 + M_{11}V_1 \end{pmatrix}_i \\
&= M_{0i}V_0 + M_{1i}V_1 \\
&= \sum_j V_j M_{ji}
\end{aligned}$$

$$\begin{aligned}
(M \bullet N)_{ij} &= \left(\begin{pmatrix} M_{00} & M_{10} \\ M_{01} & M_{11} \end{pmatrix} \bullet \begin{pmatrix} N_{00} & N_{10} \\ N_{01} & N_{11} \end{pmatrix} \right)_{ij} \\
&= \begin{pmatrix} M_{00}N_{00} + M_{10}N_{01} & M_{00}N_{10} + M_{10}N_{11} \\ M_{01}N_{00} + M_{11}N_{01} & M_{01}N_{10} + M_{11}N_{11} \end{pmatrix}_{ij} \\
&= \begin{pmatrix} M_{00}N_{i0} + M_{10}N_{i1} \\ M_{01}N_{i0} + M_{11}N_{i1} \end{pmatrix}_j \\
&= M_{0j}N_{i0} + M_{1j}N_{i1} \\
&= \sum_k N_{ik} M_{kj}
\end{aligned}$$

$$\begin{aligned}
(M \bullet T)_{ijk} &= \left(\begin{pmatrix} M_{00} & M_{10} \\ M_{01} & M_{11} \end{pmatrix} \bullet \begin{pmatrix} T_{000} & T_{010} & T_{100} & T_{110} \\ T_{001} & T_{011} & T_{101} & T_{111} \end{pmatrix} \right)_{ijk} \\
&= \left(\begin{matrix} M_{00}T_{000} + M_{10}T_{001} & M_{00}T_{010} + M_{10}T_{011} \\ M_{01}T_{000} + M_{11}T_{001} & M_{01}T_{010} + M_{11}T_{011} \end{matrix} \right. \\
&\qquad \left. \begin{matrix} M_{00}T_{100} + M_{10}T_{101} & M_{00}T_{110} + M_{10}T_{111} \\ M_{01}T_{100} + M_{11}T_{101} & M_{01}T_{110} + M_{11}T_{111} \end{matrix} \right)_{ijk} \\
&= \begin{pmatrix} M_{00}T_{i00} + M_{10}T_{i01} & M_{00}T_{i10} + M_{10}T_{i11} \\ M_{01}T_{i00} + M_{11}T_{i01} & M_{01}T_{i10} + M_{11}T_{i11} \end{pmatrix}_{jk} \\
&= \begin{pmatrix} M_{00}T_{ij0} + M_{10}T_{ij1} \\ M_{01}T_{ij0} + M_{11}T_{ij1} \end{pmatrix}_{k} \\
&= M_{0k}T_{ij0} + M_{1k}T_{ij1} \\
&= \sum_l T_{ijl}M_{lk} \ \blacksquare
\end{aligned}$$

The same symbol with a numerical subscript $\bullet_n$ will be used to indicate a more general dot product between vectors, matrices and tensors. In particular, $\bullet_1$ and $\bullet_2$ are given by

$$\begin{aligned}
(T \bullet_1 V)_{ij} &= \sum_k V_k T_{kij} \\
(T \bullet_1 M)_{ijk} &= \sum_l M_{il} T_{ljk} \\
(T \bullet_2 V)_{ij} &= \sum_k V_k T_{ikj} \\
(T \bullet_2 M)_{ijk} &= \sum_l M_{jl} T_{ilk}.
\end{aligned}$$

This is all we need in this thesis, but for the sake of completeness, the general dot products is given by

$$(X \bullet_n Y)_{i_1 i_2 \cdots i_r k_1 k_2 \cdots k_{n-1} k_{n+1} \cdots k_s} = \sum_{k_n} Y_{i_1 i_2 \cdots i_r k_n} X_{k_1 k_2 \cdots k_s}.$$

The symbols $\bullet_1$ and $\bullet_2$ will be referred to as the *left* product and *right* product respectively. Note that, in fact $\bullet = \bullet_1$.

Lemma 30

$$T \bullet_1 V = \left((T^{\mathsf{T}})_0 \bullet V, (T^{\mathsf{T}})_1 \bullet V \right)$$

$$T \bullet_1 M = \left((T^{\mathsf{T}})_0 \bullet M, (T^{\mathsf{T}})_1 \bullet M \right)^{\mathsf{T}}$$

$$T \bullet_2 V = (T_0 \bullet V, T_1 \bullet V)$$
$$T \bullet_2 M = (T_0 \bullet M, T_1 \bullet M)$$

Proof:

$$\begin{aligned}
(T \bullet_1 V)_{ij} &= \sum_k V_k T_{kij} \\
&= \sum_k V_k \left(T^\mathsf{T}\right)_{ikj} \\
&= \sum_k V_k \left(\left(T^\mathsf{T}\right)_i\right)_{kj} \\
&= \left(\left(T^\mathsf{T}\right)_i \bullet V\right)_j \\
&= \left(\left(T^\mathsf{T}\right)_0 \bullet V, \left(T^\mathsf{T}\right)_1 \bullet V\right)_{ij}
\end{aligned}$$

$$\begin{aligned}
(T \bullet_1 M)_{ijk} &= \sum_l M_{il} T_{ljk} \\
&= \sum_l M_{il} \left(T^\mathsf{T}\right)_{jlk} \\
&= \sum_l M_{il} \left(\left(T^\mathsf{T}\right)_j\right)_{lk} \\
&= \left(\left(T^\mathsf{T}\right)_j \bullet M\right)_{ik} \\
&= \left(\left(T^\mathsf{T}\right)_0 \bullet M, \left(T^\mathsf{T}\right)_1 \bullet M\right)_{jik} \\
&= \left(\left(\left(T^\mathsf{T}\right)_0 \bullet M, \left(T^\mathsf{T}\right)_1 \bullet M\right)^\mathsf{T}\right)_{ijk}
\end{aligned}$$

$$\begin{aligned}
(T \bullet_2 V)_{ij} &= \sum_k V_k T_{ikj} \\
&= \sum_k V_k (T_i)_{kj} \\
&= (T_i \bullet V)_j \\
&= (T_0 \bullet V, T_1 \bullet V)_{ij}
\end{aligned}$$

$$\begin{aligned}
(T \bullet_2 M)_{ijk} &= \sum_l M_{jl} T_{ilk} \\
&= \sum_l M_{jl} (T_i)_{lk} \\
&= (T_i \bullet M)_{jk} \\
&= (T_0 \bullet M, T_1 \bullet M)_{ijk} \quad \blacksquare
\end{aligned}$$

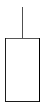
8.2 Vectors and Extended Rational Numbers

The set of vectors $\mathbb{V}$ with integer coefficients together with product $\times$ defined by

$$\begin{pmatrix} a \\ b \end{pmatrix} \times \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ bd \end{pmatrix}$$

is a monoid. Therefore, the kernel of the morphism $\Phi : \mathbb{V} \to \mathbb{Q}^{\infty}$ from the monoid of vectors $\mathbb{V}$ to the extended rational numbers given by

$$\Phi \begin{pmatrix} a \\ b \end{pmatrix} = \begin{cases} \perp & \text{if } a = 0 \text{ and } b = 0 \\ \infty & \text{if } a \neq 0 \text{ and } b = 0 \\ \frac{a}{b} & \text{if } b \neq 0 \end{cases}$$

is the monoid of non-zero integers $\mathbb{Z}^*$. Consequently, the monoid of extended rational numbers $\mathbb{Q}^{\infty}$ is isomorphic to the quotient monoid $\mathbb{V}/\mathbb{Z}^*$. In the light of this, we shall consider vectors to be equivalent, denoted by $\equiv$, up to scaling. For convenience, we will drop Φ whenever it is clear to do so and an extended real number will be pictured abstractly by . Let us define the set of *unsigned vectors* $\mathbb{V}^+$ by

$$\mathbb{V}^+ = \{V \in \mathbb{V} \,|\, \Phi(V) \in [0, \infty]\}.$$

and let $\left| \begin{pmatrix} a \\ b \end{pmatrix} \right| = \begin{pmatrix} |a| \\ |b| \end{pmatrix}$. Note that if $V \in \mathbb{V}^+$ then $V \equiv |V|$. Let the *sign* function $\sigma : \mathbb{V} \to \{-1, 0, 1\}$ be defined by

$$\sigma \begin{pmatrix} a \\ b \end{pmatrix} = \begin{cases} -1 & \text{if } a < 0 \text{ and } b \leq 0 \\ 0 & \text{if } a < 0 \text{ and } b > 0 \\ -1 & \text{if } a = 0 \text{ and } b < 0 \\ 0 & \text{if } a = 0 \text{ and } b = 0 \\ 1 & \text{if } a = 0 \text{ and } b > 0 \\ 0 & \text{if } a > 0 \text{ and } b < 0 \\ 1 & \text{if } a > 0 \text{ and } b \geq 0. \end{cases} \tag{8.1}$$

Note that

$$\mathbb{V}^+ = \{V \in \mathbb{V} \,|\, \sigma(V) \neq 0\}.$$

Therefore, the set of unsigned vectors can be identified by the set

$$\mathbb{V}^+ \cap \mathbb{V}^* = \mathbb{V}^* - \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}.$$

8.3 Matrices and Möbius Transformations

A *Möbius transformation* is considered to be a linear fractional transformation

$$x \mapsto \frac{ax+c}{bx+d}$$

on the extended complex plane $\mathbb{C}^\infty$ with $a, b, c, d \in \mathbb{C}$. A Möbius transformation with

$$\mathsf{det}\begin{pmatrix} a & c \\ b & d \end{pmatrix} = 0$$

will be referred to as a *singular Möbius transformation*. The set of non-singular Möbius transformations with coefficients taken from the restricted field $\mathbb{F} \subseteq \mathbb{C}$ together with function composition forms a group, which we shall denote by $\mathcal{M}(\mathbb{F})$. In particular, if we consider the group $\mathcal{M}(\mathbb{Q})$ and utilize the scaling invariance of Möbius transformation, we can restrict the coefficients to the set of integers $\mathbb{Z}$. Note that the singular Möbius transformations are not included in the group $\mathcal{M}(\mathbb{Q})$. However, the set of all Möbius transformations with integer coefficients does form a monoid.

Let $\Psi\begin{pmatrix} a & c \\ b & d \end{pmatrix}$ denote the Möbius transformation

$$x \mapsto \frac{ax+c}{bx+d} \tag{8.2}$$

on the extended real numbers $\mathbb{R}^\infty$ formed by the four integers a, b, c and d arranged conveniently in the matrix $M = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$. This definition includes singular matrices. So, this definition includes some strange mappings, such as

$$\Psi\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}(x) = \begin{cases} 1 & \text{if } x \neq 1 \\ \perp & \text{if } x = 1 \end{cases}$$

and

$$\Psi\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}(x) = 0 \times x = \begin{cases} 0 & \text{if } x \neq \infty \\ \perp & \text{if } x = \infty \end{cases}$$

and

$$\Psi\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}(x) = \perp.$$

The composition of Möbius transformations $\Psi(M)$ and $\Psi(N)$ is equivalent to the product of matrices M and N

$$\Psi(M)(\Psi(N)(x)) = \Psi(M \bullet N)(x).$$

Therefore, $\Psi : \mathbb{M} \to \mathbb{R}^\infty \to \mathbb{R}^\infty$ is a morphism from the monoid of matrices with integer coefficients $\mathbb{M}$ to Möbius transformations and the kernel of Ψ is the monoid of non-zero integers $\mathbb{Z}^*$. Consequently, the monoid of Möbius transformations is isomorphic to the quotient monoid $\mathbb{M}/\mathbb{Z}^*$. In the light of this, we shall consider matrices to be equivalent, denoted by $\equiv$, up to scaling. Let

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{8.3}$$

denote the identity matrix and define the *tame inverse* of a matrix by

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix}^\dagger = \begin{pmatrix} d & -c \\ -b & a \end{pmatrix} \tag{8.4}$$

and note that

$$M^\dagger \bullet M = \begin{pmatrix} \mathsf{det}(M) & 0 \\ 0 & \mathsf{det}(M) \end{pmatrix} \equiv I$$

provided that

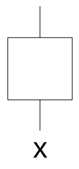
$$\mathsf{det}(M) \neq 0.$$

The function apply given by

$$\begin{array}{rcl} \mathsf{apply} : \Phi(\mathbb{V}) \times \Psi(\mathbb{M}) & \longrightarrow & \Phi(\mathbb{V}) \\ (x, f) & \longmapsto & f(x) \end{array}$$

is an action of the monoid of Möbius transformations on the extended real numbers because

$$\Psi(M)(\Phi(V)) = \Phi(M \bullet V).$$

For convenience, we will drop Ψ whenever it is clear to do so and a Möbius transformation will be pictured abstractly by 

. Define the set of *unsigned matrices* $\mathbb{M}^+$ by

$$\mathbb{M}^+ = \{M \in \mathbb{M} \,|\, \forall x \in [0, \infty] \cdot \Psi(M)(x) \in [0, \infty]\}$$

and let $\left|\begin{pmatrix} a & c \\ b & d \end{pmatrix}\right| = \begin{pmatrix} |a| & |c| \\ |b| & |d| \end{pmatrix}$. Note that

$$\begin{aligned} \mathbb{M}^+ &= \left\{M \in \mathbb{M} \,\middle|\, \forall V \in \mathbb{V}^+ \cdot \Phi(M \bullet V) \in [0, \infty]\right\} \\ &= \{M \in \mathbb{M} \,|\, \sigma(M_0) = \sigma(M_1) \neq 0\} \end{aligned}$$

and if $M \in \mathbb{M}^+$ then $M \equiv |M|$. Therefore, the set of unsigned matrices can be identified by the set

$$\mathbb{M}^+ \cap \mathbb{M}^* = \left\{ \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix} \right) \middle| a, b, c, d \in \mathbb{N} \wedge \left(\begin{smallmatrix} a \\ b \end{smallmatrix} \right), \left(\begin{smallmatrix} c \\ d \end{smallmatrix} \right) \neq \left(\begin{smallmatrix} 0 \\ 0 \end{smallmatrix} \right) \right\}.$$

8.4 The Theory of Möbius Transformations

In this section, we explore the relevant theoretical properties [20] of non-singular Möbius transformations with integer coefficients.

8.4.1 Special Base Interval

It is convenient to broaden the investigation to the group of Möbius transformations $\mathcal{M}(\mathbb{Q})$. One basic property of the group of Möbius transformations $\mathcal{M}(\mathbb{Q})$ is that it is 3-transitive. This means that for any pair of distinct triples (x_1, x_2, x_3) and (y_1, y_2, y_3) with $x_i, y_i \in \mathbb{Q}^\infty$ for all $i \in \{1, 2, 3\}$, there exists a unique Möbius transformation $M \in \mathcal{M}(\mathbb{Q})$ with $M(x_i) = y_i$ for all $i \in \{1, 2, 3\}$. An immediate consequence is the following property.

Proposition 31 *If $[p, q]$ and $[r, s]$ are two non-trivial closed intervals on the extended real line then there exists a Möbius transformation M with $M([p, q]) = [r, s]$.*

It follows that if we fix a base interval, then we can express, or encode, all other non-trivial closed intervals as the image of this base interval under a Möbius transformation. The most efficient base interval is $[0, \infty]$ as no computation is needed to determine the Möbius transformation in the proposition. The Möbius transformation is simply written down. Indeed $x \mapsto \dfrac{rx + s}{x + 1}$ and $x \mapsto \dfrac{sx + r}{x + 1}$ both map $[0, \infty]$ to $[r, s]$, the first reverses the orientation whilst the second preserves it. The closed interval $[0, \infty]$ will be called the *special base interval.*

A Möbius transformation M *refines* an interval $[p, q]$ if $M([p, q]) \subseteq [p, q]$. Clearly, the identity transformation refines any interval. We say a subset $S \subseteq \mathbb{M}$ refines $[p, q]$ if each elements of S refines $[p, q]$. It follows that there is a largest submonoid of $\mathbb{M}$, which refines a given interval. It is easy to see that $\mathbb{M}^+$ is the largest refining submonoid of the special base interval $[0, \infty]$ and more generally that $N\mathbb{M}^+N^\dagger$ is the largest refining submonoid of $N[0, \infty]$.

8.4.2 Classifications

The group of non-singular Möbius transformations $\mathcal{M}(\mathbb{R})$ can be classified according to their conjugacy classes and their geometric dynamics on the extended real line. Alternatively, we can look at the group $GL(2,\mathbb{R})$ because $\mathcal{M}(\mathbb{R}) \cong GL(2,\mathbb{R})/\mathbb{R}^*$. The problem of finding canonical representatives of conjugacy classes of $GL(2,\mathbb{R})$ is that of finding for a given matrix M a conjugate matrix $N^\dagger MN$ of pleasantly simple form. It is natural to start the search by looking for subgroups of $GL(2,\mathbb{R})$ fixed by M. This is the same as finding non-zero vectors V with real coefficients such that $M \bullet V = \lambda V$. It follows that

$$\mathsf{det}(M - \lambda I) = 0.$$

So, if

$$M = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

and

$$\mathsf{trace}(M) = a + d$$

then

$$\mathsf{det}(M - \lambda I) = \lambda^2 - \mathsf{trace}(M)\lambda + \mathsf{det}(M).$$

The polynomial

$$\chi(\lambda, M) = \lambda^2 - \mathsf{trace}(M)\lambda + \mathsf{det}(M) \tag{8.5}$$

is known as the *characteristic polynomial* of M, and its zeroes are the *eigenvalues* of M. An interesting property, which is a consequence of Cayley's Theorem [50], is that

$$\chi(M, M) = 0.$$

Let us define $\mathsf{invariance}(M) \in \mathbb{R}$ by

$$\mathsf{invariance}(M) = \frac{\mathsf{trace}(M)^2}{\mathsf{det}(M)}.$$

The roots of the characteristic polynomial $\chi(\lambda, M)$ are

$$\frac{1}{2}\mathsf{trace}(M) \pm \frac{1}{2}\sqrt{\left(\mathsf{trace}(M)^2 - 4\mathsf{det}(M)\right)}$$

with three distinct cases [50, chapter 12]:

1. **Two real roots :**

   $$\mathsf{invariance}(M) > 4$$

   The roots λ, μ are necessarily non-zero because $\mathsf{det}(M) \neq 0$.

   $$\chi(t, M) = (t - \lambda)(t - \mu)$$

   Without loss of generality, we can assume that $|\lambda| \geq |\mu|$. Therefore, a canonical representative is

   $$N = \begin{pmatrix} \lambda & 0 \\ 0 & \mu \end{pmatrix}$$

   because $\chi(\lambda, N) = 0$ and $\chi(\lambda, N) = 0$.

2. **One real root :**

   $$\mathsf{invariance}(M) = 4$$

   The root λ is necessarily non-zero because $\mathsf{det}(M) \neq 0$.

   $$\chi(t, M) = (t - \lambda)^2$$

   We know that $(M - \lambda)^2 = 0$. This reveals two possibilities.

   (a) $M - \lambda I = 0$. Therefore, a canonical representative is

   $$N = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

   because $N - \lambda I = 0$ and $\chi(\lambda, N) = 0$.

   (b) $M - \lambda I \neq 0$. Therefore, a canonical representative is

   $$N = \begin{pmatrix} \lambda & \lambda \\ 0 & \lambda \end{pmatrix}$$

   because $N - \lambda I \neq 0$ and $\chi(\lambda, N) = 0$.

3. **No real roots :**

   $$\mathsf{invariance}(M) < 4$$

   The roots are a complex conjugate pair $(\rho e^{i\alpha}, \rho e^{-i\alpha})$ where $\rho > 0$ and $\alpha \in (0, \pi)$.

   $$\chi(t, M) = \left(t - \rho e^{i\alpha}\right)\left(t - \rho e^{-i\alpha}\right) = t^2 - 2t\rho \mathsf{cos}(\alpha) + \rho^2$$

A canonical representative is

$$N = \rho R_\alpha$$
$$R_\alpha = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

because $\chi\left(\rho e^{\pm i\alpha}, N\right) = 0$.

In terms of classifications, it can be shown [50, exercise 17.5] that the canonical representatives of the conjugacy classes of the group of non-singular Möbius transformations with real coefficients are

- **Improper Hyperbolic :** $x \mapsto ax$ for $a < -1$.
- **Elliptic :** $x \mapsto \dfrac{x\cos(\alpha) + \sin(\alpha)}{-x\sin(\alpha) + \cos(\alpha)}$ for $\alpha \in (0, \pi]$.
- **Parabolic :** $x \mapsto x + 1$.
- **Identity :** $x \mapsto x$.
- **Proper Hyperbolic :** $x \mapsto ax$ for $a > 1$.

Note that hyperbolic corresponds to a *dilation* or *contraction*, elliptic corresponds to a *rotation* and parabolic corresponds to a *translation*. It can be shown [50, exercise 17.6] that $\mathsf{invariance}(M) = \mathsf{invariance}(N) \neq 4$ if and only if M and N are conjugate. In particular, it can be shown [50, exercise 17.7] that

- if $\mathsf{invariance}(M) < 0$ then M is improper hyperbolic,
- if $\mathsf{invariance}(M) \in [0, 4)$ then M is elliptic,
- if $\mathsf{invariance}(M) = 4$ then M is either parabolic or the identity,
- if $\mathsf{invariance}(M) > 4$ then M is proper hyperbolic.

Incidently, the term *loxodromic* is reserved for maps $M \in \mathcal{M}(\mathbb{C})$ in which the invariance $\mathsf{invariance}(M)$ is complex. So, which conjugacy class does negation belong? Well, note that

$$\mathsf{invariance}\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} = 0.$$

Therefore, negation is elliptic. But

$$\mathsf{invariance}(R_\alpha) = 4\cos^2(\alpha).$$

Therefore, negation belongs to the conjugacy class with the canonical representative

$$R_{\frac{\pi}{2}} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = x \mapsto -\frac{1}{x}.$$

8.4.3 Elliptic Maps

The elliptic maps are conjugate to rotations. Therefore, subsets of the elliptic maps provide ideal sign sets for incremental digit representations because they would provide an elegant way to cover the extended real line using the special base interval. Therefore, it would be interesting to consider the Möbius transformations that form cyclic groups.

Proposition 32 *For any finite order Möbius transformation $\Psi(M)$ in $\mathcal{M}(\mathbb{Q})$, the order is given by*

$$\mathsf{order}(M) = \begin{cases} 1 & \textit{if}\ \mathsf{invariance}(M) = 4 \\ 2 & \textit{if}\ \mathsf{invariance}(M) = 0 \\ 3 & \textit{if}\ \mathsf{invariance}(M) = 1 \\ 4 & \textit{if}\ \mathsf{invariance}(M) = 2 \\ 6 & \textit{if}\ \mathsf{invariance}(M) = 3. \end{cases}$$

Proof : The group of Möbius transformations $\mathcal{M}(\mathbb{Q})$ is isomorphic to $GL(2,\mathbb{Q})/\mathbb{Q}^*$. Let $\Psi(M)$ be an element of finite order in $\mathcal{M}(\mathbb{Q})$. In other words,

$$M^n \equiv I.$$

Therefore

$$M^n = \mu I$$

for some $\mu \in \mathbb{Q}^*$. The eigenvalues of M satisfy

$$\begin{aligned} \chi(\lambda, M) &= 0 \\ \lambda^n &= \mu. \end{aligned}$$

with $\chi(\lambda, M)$ as defined in equation (8.5). Let us change variable according to

$$\lambda = \omega \mathsf{exp}\left(\frac{\pi i r}{n}\right).$$

Therefore

$$\begin{aligned} (-1)^r \omega^n &= \mu \\ \mathsf{trace}(M) &= 2\omega \mathsf{cos}\left(\frac{\pi r}{n}\right) \\ \mathsf{det}(M) &= \omega^2 \\ \mathsf{invariance}(M) &= 4\mathsf{cos}^2\left(\frac{\pi r}{n}\right). \end{aligned}$$

But

$$\cos^2(x) = \frac{\cos(2x)+1}{2}$$

and

$$\mathsf{invariance}(M) \in \mathbb{Q}.$$

Therefore

$$\cos\left(\frac{2\pi r}{n}\right) \in \mathbb{Q}$$

and so

$$n \in \{1,2,3,4,6\}.$$

Also

$$\mathsf{order}(M) = n \text{ if } \mathsf{gcd}(r,n) = 1$$

and so the result follows immediately. ■

So, all finite order Möbius transformations are either elliptic or the identity. For example,

$$\begin{aligned}
\mathsf{order}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} &= 1 \\
\mathsf{order}\begin{pmatrix} 2 & -1 \\ 1 & -2 \end{pmatrix} &= 2 \\
\mathsf{order}\begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} &= 3 \\
\mathsf{order}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} &= 4 \\
\mathsf{order}\begin{pmatrix} 1 & -1 \\ 1 & 2 \end{pmatrix} &= 6.
\end{aligned}$$

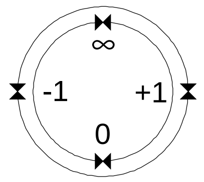
The example of order 4 is particularly interesting because it represents rotation by $\frac{\pi}{2}$

$$\begin{aligned}
S_\infty &= \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = x \mapsto \frac{1+x}{1-x} \\
S_- &= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = x \mapsto -\frac{1}{x} \equiv S_\infty \bullet S_\infty \\
S_0 &= \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = x \mapsto \frac{x-1}{x+1} \equiv S_\infty \bullet S_\infty \bullet S_\infty \\
S_+ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = x \mapsto x \equiv S_\infty \bullet S_\infty \bullet S_\infty \bullet S_\infty
\end{aligned}$$

and the Möbius transformations generated represent overlapping intervals that cover the extended real line

$$S_\infty([0,\infty]) = [1,-1]$$

$$S_-([0,\infty]) = [\infty, 0] \qquad S_+([0,\infty]) = [0,\infty]$$



$$S_0([0,\infty]) = [-1,1]\,.$$

The example of order 3 gives rise to

$$\begin{aligned} S_h &= \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} = x \mapsto \frac{1}{1-x} \\ S_l &= \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix} = x \mapsto x-1 \equiv S_h \bullet S_h \\ S_+ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = x \mapsto x \equiv S_h \bullet S_h \bullet S_h, \end{aligned}$$

which when applied to the special base interval $[0,\infty]$ represent 3 overlapping intervals that cover the extended real line

$$\begin{aligned} S_h([0,\infty]) &= [1,0] \\ S_l([0,\infty]) &= [\infty,1] \\ S_+([0,\infty]) &= [0,\infty]\,. \end{aligned}$$

The example of order 2 gives rise to

$$\begin{aligned} S_\times &= \begin{pmatrix} 2 & -1 \\ 1 & -2 \end{pmatrix} = x \mapsto \frac{2x-1}{x-2} \\ S_+ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = x \mapsto x \equiv S_- \bullet S_-, \end{aligned}$$

which when applied to the special base interval $[0,\infty]$ represent 2 overlapping intervals that cover the extended real line

$$\begin{aligned} S_\times([0,\infty]) &= \left[2,\tfrac{1}{2}\right] \\ S_+([0,\infty]) &= [0,\infty]\,. \end{aligned}$$

8.4.4 Hyperbolic Maps

The hyperbolic maps are conjugate to contracting linear (affine) maps. Therefore, they should form ideal candidates for the digit set in an incremental digit representation because they would provide a way to shrink intervals on the extended real line to a limit. Linear maps play an important role in the redundant positional representation as defined in section 7.2. For the redundant radix b positional representation the digit set is $\mathbb{Z}(b)$ and the digit map is

$$d \mapsto x \mapsto \frac{d+x}{b} = d \mapsto \begin{pmatrix} 1 & d \\ 0 & b \end{pmatrix}.$$

Note that

$$\mathsf{invariance}\begin{pmatrix} 1 & d \\ 0 & b \end{pmatrix} = 2 + b + \frac{1}{b}$$

and therefore this digit map is hyperbolic if $b \neq \pm 1$. In fact, it is proper hyperbolic if b is positive and improper hyperbolic if b is negative. The redundant positional representation operates with the base interval

$$[-1,1] = S_0([0,\infty]).$$

Therefore, the conjugate Möbius transformation given by

$${}^{b}D_d = \begin{pmatrix} b+1+d & b-1+d \\ b-1-d & b+1-d \end{pmatrix} \equiv S_0^{\dagger} \bullet \begin{pmatrix} 1 & d \\ 0 & b \end{pmatrix} \bullet S_0$$

forms the basis of the digit map $d \mapsto {}^{b}D_d$ operating on the special base interval $[0,\infty]$. This can be seen more clearly in the following commuting diagram [62, 20]

$$\begin{tikzcd}
{[-1,1]} \arrow[r, "\begin{pmatrix} 1 & d \\ 0 & b \end{pmatrix}"] & {[-1,1]} \\
{[0,\infty]} \arrow[u, "S_0"] \arrow[r, "{}^{b}D_d"'] & {[0,\infty]}. \arrow[u, "S_0"']
\end{tikzcd}$$

8.5 Tensors and Möbius Transformations

In order to compute the basic arithmetic operations on redundant continued fractions, Gosper [28] and Vuillemin [71] used *2-dimensional Möbius transformations.*

A *2-dimensional Möbius transformation* $\Upsilon(T)$ is a linear fractional transformation on the extended real numbers

$$(x,y) \mapsto \frac{axy + cx + ey + g}{bxy + dx + fy + h} \tag{8.6}$$

parameterized by the eight integers a, b, c, d, e, f, g and h arranged conveniently in the tensor $T = \begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}$. The basic arithmetic operations are given by

$$\begin{aligned}
T_{+}(x,y) &= \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}(x,y) = x + y & (8.7)\\
T_{-}(x,y) &= \begin{pmatrix} 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}(x,y) = x - y & (8.8)\\
T_{\times}(x,y) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}(x,y) = x \times y & (8.9)\\
T_{\div}(x,y) &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}(x,y) = x \div y. & (8.10)
\end{aligned}$$

However, the definition of a 2-dimensional Möbius transformation includes some strange functions such as

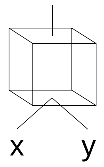
$$x \mapsto \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}(x,y) = \begin{cases} x \mapsto x & \text{if } y \neq \infty \\ x \mapsto \perp & \text{if } y = \infty \end{cases}$$

and

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}(x,y) = \perp.$$

It can be shown that

$$\begin{aligned}
\Upsilon(T)(M(x), y) &= \Upsilon(T \bullet_1 M)(x,y)\\
\Upsilon(T)(x, M(y)) &= \Upsilon(T \bullet_2 M)(x,y)\\
\Upsilon(T)(V, y) &= \Psi(T \bullet_1 V)(y)\\
\Upsilon(T)(x, V) &= \Psi(T \bullet_2 V)(x)\\
M(\Upsilon(T)(x,y)) &= \Upsilon(M \bullet T)(x,y).
\end{aligned}$$

We shall consider tensors to be equivalent, denoted by $\equiv$, up to scaling. For convenience, we will drop Υ whenever it is clear to do so and a 2-dimensional Möbius transformation will be pictured abstractly by 

. Define the set of

unsigned tensors $\mathbb{T}^+$ by

$$\mathbb{T}^+ = \{T \in \mathbb{T} \,| \forall x, y \in [0,\infty] \cdot \Upsilon(T)(x,y) \in [0,\infty]\}$$

and let $\left|\begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}\right| = \begin{pmatrix} |a| & |c| & |e| & |g| \\ |b| & |d| & |f| & |h| \end{pmatrix}$. Note that

$$\begin{aligned} \mathbb{T}^+ &= \{T \in \mathbb{T} \,| \forall V, W \in \mathbb{V}^+ \cdot \Phi(T \bullet_1 V \bullet W) \in [0,\infty]\} \\ &= \{T \in \mathbb{T} \,| \sigma(T_{00}) = \sigma(T_{01}) = \sigma(T_{10}) = \sigma(T_{11}) \neq 0\} \end{aligned}$$

and if $T \in \mathbb{T}^+$ then $T \equiv |T|$. Therefore, the set of unsigned tensors can be identified by the set

$$\mathbb{T}^+ \cap \mathbb{T}^* = \left\{ \begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix} \middle| \begin{array}{l} a,b,c,d,e,f,g,h \in \mathbb{N} \wedge \\ \begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix}, \begin{pmatrix} e \\ f \end{pmatrix}, \begin{pmatrix} g \\ h \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{array} \right\}.$$

8.6 Information

The extended rational numbers, Möbius transformations and 2-dimensional Möbius transformation will be collectively referred to as *linear fractional transformations.* So, let

$$\begin{aligned} \mathbb{L} &= \mathbb{V} \cup \mathbb{M} \cup \mathbb{T} \\ \mathbb{L}^* &= \mathbb{V}^* \cup \mathbb{M}^* \cup \mathbb{T}^* \\ \mathbb{L}^+ &= \mathbb{V}^+ \cup \mathbb{M}^+ \cup \mathbb{T}^+. \end{aligned}$$

Definition 33 *The* information *contained in a linear fractional transformation is*

$$\begin{aligned} \mathsf{info} &: \mathbb{L} \to \mathbb{I}^C[\mathbb{Q}^\infty] \\ \mathsf{info}(V) &= \begin{cases} \mathbb{R}^\infty & \textit{if } V = \bot \\ \{V\} & \textit{otherwise} \end{cases} \\ \mathsf{info}(M) &= \begin{cases} \mathbb{R}^\infty & \textit{if } \exists x \in [0,\infty] \cdot M(x) = \bot \\ M([0,\infty]) & \textit{otherwise} \end{cases} \\ \mathsf{info}(T) &= \begin{cases} \mathbb{R}^\infty & \textit{if } \exists x, y \in [0,\infty] \cdot T(x,y) = \bot \\ T([0,\infty],[0,\infty]) & \textit{otherwise.} \end{cases} \end{aligned}$$

Proposition 34

$$\begin{array}{lllllllll} V & \in & \mathbb{V}^+ & \textit{iff} & \forall M & \in & \mathbb{M} & \cdot & \mathsf{info}(M) \sqsubseteq \mathsf{info}(M \bullet V) \\ V & \in & \mathbb{V}^+ & \textit{iff} & \forall T & \in & \mathbb{T} & \cdot & \mathsf{info}(T) \sqsubseteq \mathsf{info}(T \bullet_1 V) \\ V & \in & \mathbb{V}^+ & \textit{iff} & \forall T & \in & \mathbb{T} & \cdot & \mathsf{info}(T) \sqsubseteq \mathsf{info}(T \bullet_2 V) \\ M & \in & \mathbb{M}^+ & \textit{iff} & \forall N & \in & \mathbb{M} & \cdot & \mathsf{info}(N) \sqsubseteq \mathsf{info}(N \bullet M) \\ M & \in & \mathbb{M}^+ & \textit{iff} & \forall T & \in & \mathbb{T} & \cdot & \mathsf{info}(T) \sqsubseteq \mathsf{info}(T \bullet_1 M) \\ M & \in & \mathbb{M}^+ & \textit{iff} & \forall T & \in & \mathbb{T} & \cdot & \mathsf{info}(T) \sqsubseteq \mathsf{info}(T \bullet_2 M) \\ T & \in & \mathbb{T}^+ & \textit{iff} & \forall M & \in & \mathbb{M} & \cdot & \mathsf{info}(M) \sqsubseteq \mathsf{info}(M \bullet T) \end{array}$$

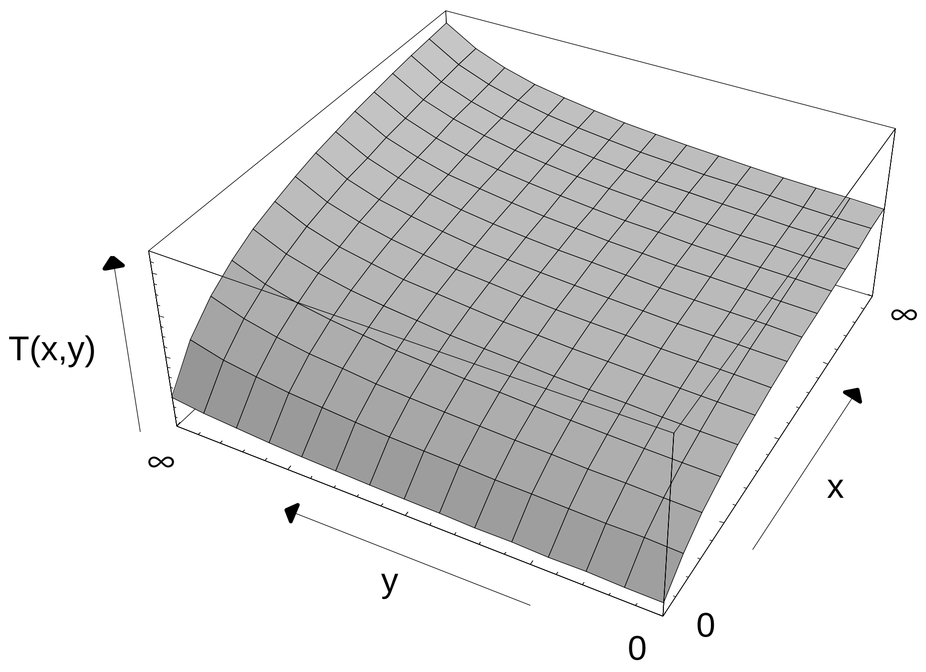


Figure 8.1: The surface plot of a typical tensor $T(x,y)$ over x and y.

Proof: We only really need to show that

$$V \in \mathbb{V}^{+} \text{ iff } \forall M \in \mathbb{M} \cdot \mathsf{info}(M) \sqsubseteq \mathsf{info}(M \bullet V).$$

"$\Rightarrow$". Assume $V \in \mathbb{V}^{+}$. By definition, $V \in [0,\infty]$. Consider $M \in \mathbb{M}$. Let $\mathsf{info}(M) = \mathbb{R}^{\infty}$. Then $\mathbb{R}^{\infty} \sqsubseteq \mathsf{info}(M \bullet V)$. Therefore $\mathsf{info}(M) \sqsubseteq \mathsf{info}(M \bullet V)$. Let $\mathsf{info}(M) = M([0,\infty])$. But $M([0,\infty]) \sqsubseteq M(\{V\})$. Therefore $\mathsf{info}(M) \sqsubseteq \mathsf{info}(M \bullet V)$. "$\Leftarrow$". Assume $\mathsf{info}(M) \sqsubseteq \mathsf{info}(M \bullet V)$ for all $M \in \mathbb{M}$. Choose $M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. But $\mathsf{info}(M) = [0,\infty]$ and $\mathsf{info}(M \bullet V) = \mathsf{info}(V)$. Therefore $[0,\infty] \sqsubseteq \mathsf{info}(V)$. So $\mathsf{info}(V) \neq \mathbb{R}^{\infty}$. Therefore $V \in [0,\infty]$. In other words, $V \in \mathbb{V}^{+}$. ■

As a result of this proposition, a linear fractional transformation is said to satisfy the *refinement property* if it is a member of $\mathbb{L}^{+}$. A 2-dimensional Möbius transformation is orientation preserving or reversing in each argument separately, with respect to the one-point compactification of the real numbers, as illustrated in figure 8.1. Therefore

$$\mathsf{info}\left(\begin{tikzcd}
a \arrow[rr, dash] \arrow[dd, dash] \arrow[dr, dash] & & e \arrow[dd, dash] \arrow[dr, dash] & \\
& c \arrow[rr, dash] \arrow[dd, dash] & & g \arrow[dd, dash] \\
b \arrow[rr, dash] \arrow[dr, dash] & & f \arrow[dr, dash] & \\
& d \arrow[rr, dash] & & h
\end{tikzcd}\right) =$$

$$\mathsf{info}\begin{pmatrix} a & c \\ b & d \end{pmatrix} \cup \mathsf{info}\begin{pmatrix} c & g \\ d & h \end{pmatrix} \cup \mathsf{info}\begin{pmatrix} g & e \\ h & f \end{pmatrix} \cup \mathsf{info}\begin{pmatrix} e & a \\ f & b \end{pmatrix}.$$

Deriving an elegant and efficient algorithm for the information function info is surprising difficult. Consider the relations $\asymp$ and $\prec$ on vectors $\mathbb{V}$ given by

$$\begin{array}{lll} V \asymp W & \text{if} & \exists r < 0 \cdot V = rW \\ V \preceq W & \text{if} & \mathsf{det}\,(V, W) \leq 0 \end{array}$$

and let

$$\begin{aligned}
\mathsf{bottom}\,(V_0, V_1, \ldots, V_{n-1}) &= \begin{cases} \mathsf{true} & \text{if } \exists i, j \in \mathbb{N}\,(n) \cdot V_i \asymp V_j \\ \mathsf{false} & \text{otherwise} \end{cases} \\
\mathsf{min}\,(V_0, V_1, \ldots, V_{n-1}) &= V\,(\mu i \cdot \forall j \in \mathbb{N}\,(n) \cdot V_i \preceq V_j) \\
\mathsf{max}\,(V_0, V_1, \ldots, V_{n-1}) &= V\,(\mu i \cdot \forall j \in \mathbb{N}\,(n) \cdot V_j \preceq V_i)
\end{aligned}$$

where μi indicates the minimum i and

$$\begin{aligned}
\mathsf{interval}\,(X) &= \begin{cases} \mathbb{R}^\infty & \text{if } \mathsf{bottom}\,(X) = \mathsf{true} \\ [\mathsf{min}\,(X), \mathsf{max}\,(X)] & \text{if } \mathsf{bottom}\,(X) = \mathsf{false} \end{cases} \\
\mathsf{list}\,(V) &= (V) \\
\mathsf{list}\,(M) &= (M_0, M_1) \\
\mathsf{list}\,(T) &= (T_{00}, T_{01}, T_{10}, T_{11})\,.
\end{aligned}$$

Proposition 35

$$\mathsf{info} = \mathsf{interval} \circ \mathsf{list}$$

Proof:

FOR VECTORS:

$$\begin{aligned}\mathsf{info}\,(V) &= \begin{cases}\mathbb{R}^\infty & \text{if } V = \perp \\ \{V\} & \text{otherwise}\end{cases} \\ &= \begin{cases}\mathbb{R}^\infty & \text{if } V \asymp V \\ [\mathsf{min}\,(\mathsf{list}\,(V))\,, \mathsf{max}\,(\mathsf{list}\,(V))] & \text{otherwise}\end{cases} \\ &= \mathsf{interval}\,(\mathsf{list}\,(V))\end{aligned}$$

For matrices:
Assume

$$\exists\, V \in \mathbb{V}^+ \cap \mathbb{V}^* \cdot M\,(V) = \perp$$

Therefore, there exists $\begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{V}^+ \cap \mathbb{V}^*$ (note that $a, b \geq 0$) such that

$$aM_0 + bM_1 = \perp$$

or equivalently

$$\begin{cases} M_1 = \perp & \text{if } a = 0 \text{ and } b > 0 \\ aM_0 + bM_1 = \perp & \text{if } a > 0 \text{ and } b > 0 \\ M_0 = \perp & \text{if } a > 0 \text{ and } b = 0. \end{cases}$$

Therefore

$$\exists i, j \in \mathbb{N}\,(2) \cdot M_i \asymp M_j.$$

Assume

$$\forall V \in \mathbb{V}^+ \cap \mathbb{V}^* \cdot M\,(V) \neq \perp$$

$$\begin{aligned}\left\{M\,(V)\,\middle|\,V \in \mathbb{V}^+\right\} &= \left\{aM_0 + bM_1 \middle| \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{V}^+\right\} \\ & [\mathsf{min}\,(\mathsf{list}\,(M))\,, \mathsf{max}\,(\mathsf{list}\,(M))]\end{aligned}$$

$$\begin{aligned}\mathsf{info}\,(M) &= \begin{cases}\mathbb{R}^\infty & \text{if } \exists V \in \mathbb{V}^+ \cdot M\,(V) = \perp \\ \{M\,(V)\,|V \in \mathbb{V}^+\} & \text{otherwise}\end{cases} \\ &= \begin{cases}\mathbb{R}^\infty & \text{if } \exists i, j \in \mathbb{N}\,(2) \cdot M_i \asymp M_j \\ [\mathsf{min}\,(\mathsf{list}\,(M))\,, \mathsf{max}\,(\mathsf{list}\,(M))] & \text{otherwise}\end{cases} \\ &= \mathsf{interval}\,(M_0, M_1)\end{aligned}$$

For tensors:
Assume

$$\exists V, W \in \mathbb{V}^+ \cap \mathbb{V}^* \cdot T\,(V, W) = \perp$$

Therefore, there exists $\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix} \in \mathbb{V}^+ \cap \mathbb{V}^*$ (note that $a, b, c, d \geq 0$) such that

$$acT_{00} + bcT_{01} + adT_{10} + bdT_{11} = \bot$$

or equivalently

$$\begin{cases} T_{11} = \bot & \text{if } a = 0 \wedge b > 0 \wedge c = 0 \wedge d > 0 \\ aT_{10} + bT_{11} = \bot & \text{if } a > 0 \wedge b > 0 \wedge c = 0 \wedge d > 0 \\ T_{10} = \bot & \text{if } a > 0 \wedge b = 0 \wedge c = 0 \wedge d > 0 \\ cT_{01} + dT_{11} = \bot & \text{if } a = 0 \wedge b > 0 \wedge c > 0 \wedge d > 0 \\ acT_{00} + bcT_{01} + adT_{10} + bdT_{11} = \bot & \text{if } a > 0 \wedge b > 0 \wedge c > 0 \wedge d > 0 \\ cT_{00} + dT_{10} = \bot & \text{if } a > 0 \wedge b = 0 \wedge c > 0 \wedge d > 0 \\ T_{01} = \bot & \text{if } a = 0 \wedge b > 0 \wedge c > 0 \wedge d = 0 \\ aT_{00} + bT_{01} = \bot & \text{if } a > 0 \wedge b > 0 \wedge c > 0 \wedge d = 0 \\ T_{00} = \bot & \text{if } a > 0 \wedge b = 0 \wedge c > 0 \wedge d = 0. \end{cases}$$

Therefore

$$\text{if } \exists i, j, k, l \in \mathbb{N}(2) \cdot T_{ij} \asymp T_{kl}.$$

Assume

$$\forall V, W \in \mathbb{V}^+ \cap \mathbb{V}^* \cdot T(V, W) \neq \bot$$

$$\begin{aligned} & \left\{ T(V, W) \,\middle|\, V, W \in \mathbb{V}^+ \right\} \\ = & \left\{ acT_{00} + bcT_{01} + adT_{10} + bdT_{11} \,\middle|\, \begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix} \in \mathbb{V}^+ \right\} \\ = & \left[\mathsf{min}(\mathsf{list}(T)), \mathsf{max}(\mathsf{list}(T)) \right] \end{aligned}$$

$$\begin{aligned} \mathsf{info}(T) &= \begin{cases} \mathbb{R}^\infty & \text{if } \exists\, V, W \in \mathbb{V}^+ \cdot T(V, W) = \bot \\ \{T(V, W) \,|\, V, W \in \mathbb{V}^+\} & \text{otherwise} \end{cases} \\ &= \begin{cases} \mathbb{R}^\infty & \text{if } \exists i, j, k, l \in \mathbb{N}(2) \cdot T_{ij} \asymp T_{kl} \\ [\mathsf{min}(\mathsf{list}(T)), \mathsf{max}(\mathsf{list}(T))] & \text{otherwise} \end{cases} \\ &= \mathsf{interval}(\mathsf{list}(T)) \ \blacksquare \end{aligned}$$

The information contained in a vector and a matrix can be written more explicitly as

$$\mathsf{info}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{cases} \mathbb{R}^\infty & \text{if } a = 0 \text{ and } b = 0 \\ \left\{ \frac{a}{b} \right\} & \text{otherwise} \end{cases}$$

and

$$\mathsf{info}\begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{cases} \nu(p,q,r,s) & \text{if } \exists p,q,r,s \in \mathbb{Z} \cdot \begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{pmatrix} rp & sp \\ rq & sq \end{pmatrix} \\ \left[\frac{a}{b}, \frac{c}{d}\right] & \text{if } \det\begin{pmatrix} a & c \\ b & d \end{pmatrix} < 0 \\ \left[\frac{c}{d}, \frac{a}{b}\right] & \text{if } \det\begin{pmatrix} a & c \\ b & d \end{pmatrix} > 0 \end{cases}$$

respectively where

$$\nu(p,q,r,s) = \begin{cases} \mathbb{R}^\infty & \text{if } rs \le 0 \\ \left\{\frac{p}{q}\right\} & \text{if } rs > 0. \end{cases}$$

The *head* of a tensor T that contains some information (i.e. $\mathsf{info}(T) \neq \mathbb{R}^\infty$) is given by

$$T^{\text{head}} = (\mathsf{max}(\mathsf{list}(T)), \mathsf{min}(\mathsf{list}(T))).$$

Note that

$$\begin{aligned} \mathsf{info}(T) \neq \mathbb{R}^\infty &\Rightarrow \mathsf{bottom}(\mathsf{list}(T)) = \mathsf{false} \\ &\Rightarrow \mathsf{bottom}(\mathsf{max}(\mathsf{list}(T)), \mathsf{min}(\mathsf{list}(T))) = \mathsf{false}. \end{aligned}$$

Therefore

$$\begin{aligned} \mathsf{info}\left(T^{\text{head}}\right) &= \mathsf{info}(H, L) \\ &= \mathsf{interval}(H, L) \\ &= [\mathsf{min}(H, L), \mathsf{max}(H, L)] \\ &= [L, H] \\ &= \mathsf{info}(T) \\ \text{where } L &= \mathsf{min}(\mathsf{list}(T)) \\ \text{and } H &= \mathsf{max}(\mathsf{list}(T)). \end{aligned}$$

In other words, the head of a tensor is a matrix that incorporates the same information as the tensor provided that the tensor contains some information. The *tail* of a tensor T that contains some information is given by

$$T^{\text{tail}} = \left(T^{\text{head}}\right)^\dagger \bullet T \in \mathbb{T}^+.$$

As mentioned earlier, Möbius transformations are usually considered over the entire extended complex plane rather than just the extended real line. Interestingly, Möbius transformations are conformal over the extended complex plane.

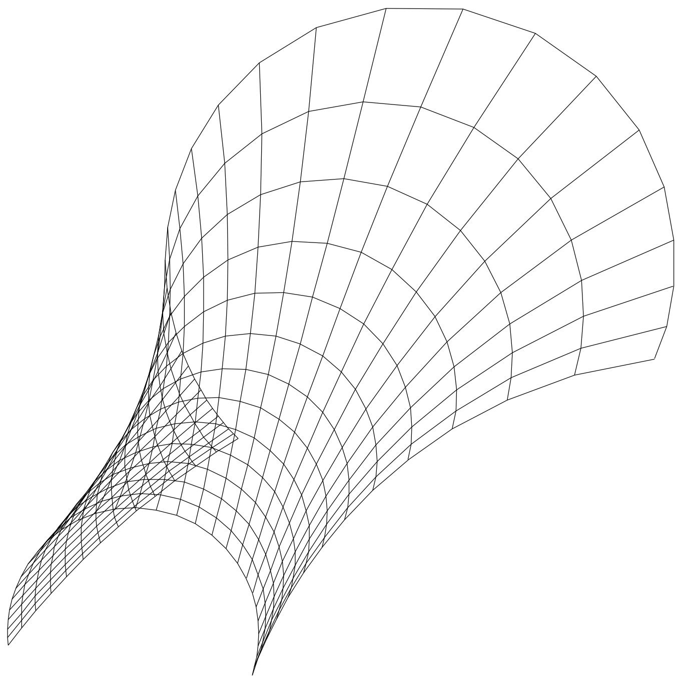


Figure 8.2: The information $T([0,\infty],[0,\infty]) \subseteq \mathbb{C}^{\infty}$ in a tensor T with Gaussian integer coefficients $\mathbb{Q}\left(\sqrt{-1}\right)$ is constructed using circular arcs on the extended complex plane $\mathbb{C}^{\infty}$.

In other words, circles are mapped to circles and lines are interpreted as circles of infinite radius. Complex numbers can be represented by considering vectors, matrices and tensors with Gaussian integer coefficients $\mathbb{Q}\left(\sqrt{-1}\right)$. In this way, the information $\{V\} \subseteq \mathbb{C}^{\infty}$ in a vector V is a point on the extended complex plane $\mathbb{C}^{\infty}$, the information $M([0,\infty]) \subseteq \mathbb{C}^{\infty}$ in a matrix M is a circular arc on the extended complex plane $\mathbb{C}^{\infty}$ and the information $T([0,\infty],[0,\infty]) \subseteq \mathbb{C}^{\infty}$ in a tensor T is an interval (i.e. connected subspace) of the extended complex plane $\mathbb{C}^{\infty}$ constructed using circular arcs as illustrated in figure 8.2. The main problem with this extension of Möbius transformations is in finding a good algorithm for a bounding box. However, should this difficulty be overcome, there is no reason why the contents of this thesis should not be extended to exact complex arithmetic in a non-trivial manner.

8.7 Quadratic Fractional Transformations

Gosper [28] and Vuillemin [71] examined quadratic fractional transformations of the form

$$x \mapsto \frac{ax^2+cx+e}{bx^2+dx+f},$$

which we will denote by

$$\begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix}.$$

Unfortunately, the composition of quadratic fractional transformations and linear fractional transformations is not particularly elegant. For instance

$$\begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} \circ \begin{pmatrix} g & i \\ h & j \end{pmatrix} =$$

$$\begin{pmatrix} ag^2+cgh+eh^2 & (2ai+cj)\,g+(ci+2ej)\,h & ai^2+cij+ej^2 \\ bg^2+dgh+fh^2 & (2bi+dj)\,g+(di+2fj)\,h & bi^2+dij+fj^2 \end{pmatrix}.$$

Another problem is that the formula for the information in a quadratic fractional transformation, defined by

$$\mathsf{info}\begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} = \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix}([0,\infty]),$$

involves square roots. However, note that

$$\begin{aligned} \mathsf{info}\begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} &\supseteq \mathsf{info}\begin{pmatrix} 2a & c & c & 2e \\ 2b & d & d & 2f \end{pmatrix} \\ &= \mathsf{interval}\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix}, \begin{pmatrix} e \\ f \end{pmatrix}\right) \end{aligned}$$

and indeed in practice

$$\mathsf{info}_{\approx}\begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} = \mathsf{interval}\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix}, \begin{pmatrix} e \\ f \end{pmatrix}\right)$$

provides a good approximation for the information in a quadratic fractional transformation. Practical implementations reveal that quadratic fractional transformations have considerable merit for computing x^2 and transcendental functions.

Chapter 9

General Normal Products

In section 8.4, it was concluded that the special base interval $[0, \infty]$ is extremely desirable because evaluating its application to Möbius transformations is computationally trivial. This fact together with proposition 34 leads naturally to the definition of the *general normal product* [59].

Definition 36 *The* ***unsigned general normal product representation*** *on* $[0, \infty]$ *is the unsigned incremental digit representation*

$$\left(\mathbb{Q}^\infty \cap [0, \infty], [0, \infty], \mathbb{M}^+, \Psi, \mathbb{V}^+, \Phi\right).$$

Definition 37 *The* ***signed general normal product representation*** *on* $[0, \infty]$ *is the signed incremental digit representation*

$$\left(\mathbb{Q}^\infty, [0, \infty], \mathbb{M}, \Psi, \mathbb{M}^+, \Psi, \mathbb{V}^+, \Phi\right).$$

A general normal product represents an extended real number x if the least upper bound of the induced chain is the singleton set $\{x\}$. Note that vectors and matrices with pairs of zeros are disallowed. This avoids any awkward complications. For instance,

$$\begin{aligned} \mathsf{info}\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} &= [0, 1] \\ \mathsf{info}\left(\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \bullet \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right) &= \mathbb{R}^\infty \end{aligned}$$

and so

$$\mathsf{info}\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \not\supseteq \mathsf{info}\left(\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \bullet \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right).$$

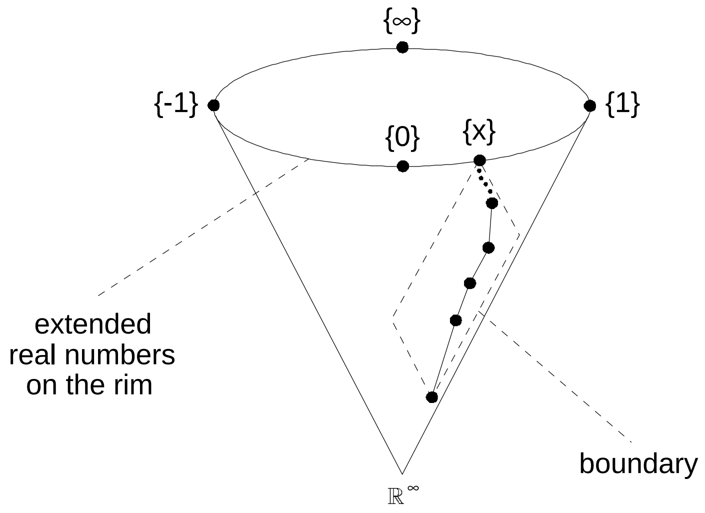


Figure 9.1: A chain representing the extended real number x on the continuous real domain $\mathbb{C}(\mathbb{R}^\infty)$ induced by a general normal product.

For example, the natural number $e = 2.71828\cdots$ can be expressed by the general normal product

$$e = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 4 & 3 \\ 3 & 2 \end{pmatrix}\begin{pmatrix} 6 & 5 \\ 5 & 4 \end{pmatrix}\begin{pmatrix} 8 & 7 \\ 7 & 6 \end{pmatrix}\cdots \tag{9.1}$$

The derivation for this general normal product can be found in section 10.2.3. This general normal product is interpreted by the following sequence of closed nested intervals:

$$\begin{aligned}
\mathsf{info}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} &= [2,\infty] \\
\mathsf{info}\left(\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\bullet\begin{pmatrix} 4 & 3 \\ 3 & 2 \end{pmatrix}\right) &= \left[\tfrac{8}{3},\tfrac{11}{4}\right] \\
\mathsf{info}\left(\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\bullet\begin{pmatrix} 4 & 3 \\ 3 & 2 \end{pmatrix}\bullet\begin{pmatrix} 6 & 5 \\ 5 & 4 \end{pmatrix}\right) &= \left[\tfrac{106}{39},\tfrac{87}{32}\right] \\
\mathsf{info}\left(\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\bullet\begin{pmatrix} 4 & 3 \\ 3 & 2 \end{pmatrix}\bullet\begin{pmatrix} 6 & 5 \\ 5 & 4 \end{pmatrix}\bullet\begin{pmatrix} 8 & 7 \\ 7 & 6 \end{pmatrix}\right) &= \left[\tfrac{1264}{465},\tfrac{1457}{536}\right]
\end{aligned}$$

A general normal product of the form

$$\prod_{n=0}^{\infty}\begin{pmatrix} a_n & b_n \\ 1 & 0 \end{pmatrix}$$

corresponds to the continued fraction $[\langle a_n . b_n \rangle_{n=0}^{\infty}]$. In fact, any general normal product consisting of non-singular matrices can be converted into a continued fraction of this form by using the identity

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{cases} \begin{pmatrix} \frac{a}{b} & \frac{c}{b} \\ 1 & 0 \end{pmatrix} & \text{if } d = 0 \\ \begin{pmatrix} \frac{c}{d} & a - \frac{bc}{d} \\ 1 & 0 \end{pmatrix} \bullet \begin{pmatrix} b & d \\ 1 & 0 \end{pmatrix} & \text{if } d \neq 0 \end{cases}$$

For example, the natural number given in equation (9.1) can be converted into the continued fraction

$$\begin{aligned} e &= \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{3}{2} & -\frac{1}{2} \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{5}{4} & -\frac{1}{4} \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 5 & 4 \\ 1 & 0 \end{pmatrix} \cdots \\ &= 2 + \cfrac{1}{\frac{3}{2} - \cfrac{\frac{1}{2}}{3 + \cfrac{2}{\frac{5}{4} - \cfrac{\frac{1}{4}}{5 + \cfrac{4}{\ddots}}}}} \end{aligned}$$

This means that Pringsheim's [65] test for convergence of continued fractions as expressed in equation (5.7) can be generalized to general normal products.

Proposition 38 *Given a sequence of non-singular matrices*

$$\left\langle \begin{pmatrix} a_n & c_n \\ b_n & d_n \end{pmatrix} \right\rangle_{n=0}^{\infty}$$

with $d_n \neq 0$ for all $n \in \mathbb{N}$ then the divergence of the series

$$\sum_{n=1}^{\infty} \left(\sqrt{\frac{d_{n-1}}{c_n}} + \frac{1}{\sqrt{b_n}} \right) \sqrt{\det \begin{pmatrix} a_n & c_n \\ b_n & d_n \end{pmatrix}}$$

is a necessary and sufficient condition for the convergence of the general normal product

$$\prod_{n=0}^{\infty} \begin{pmatrix} a_n & c_n \\ b_n & d_n \end{pmatrix}.$$

Proof:

$$\begin{aligned}
\prod_{n=0}^{\infty}\begin{pmatrix} a_n & c_n \\ b_n & d_n \end{pmatrix} &= \left[\left\langle \frac{c_n}{d_n}.a_n - \frac{b_n c_n}{d_n}, b_n.d_n \right\rangle_{n=0}^{\infty}\right] \\
&= [\langle \alpha_n.\beta_n \rangle_{n=0}^{\infty}] \\
\text{where } \alpha_{2n} &= \frac{c_n}{d_n}, \beta_{2n} = a_n - \frac{b_n c_n}{d_n}, \alpha_{2n+1} = b_n, \beta_{2n+1} = d_n
\end{aligned}$$

$$\begin{aligned}
\sum_{n=2}^{\infty}\sqrt{\frac{\beta_{n-1}\beta_n}{\alpha_n}} &= \sum_{n=1}^{\infty}\sqrt{\frac{\beta_{2n-1}\beta_{2n}}{\alpha_{2n}}} + \sqrt{\frac{\beta_{2n}\beta_{2n+1}}{\alpha_{2n+1}}} \\
&= \sum_{n=1}^{\infty}\left(\sqrt{\frac{d_{n-1}}{c_n}} + \frac{1}{\sqrt{b_n}}\right)\sqrt{\det\begin{pmatrix} a_n & c_n \\ b_n & d_n \end{pmatrix}} \blacksquare
\end{aligned}$$

Deriving a general normal product from a continued fraction is made difficult by the requirement for positive matrices $\mathbb{M}^+$. In particular, the product of matrices

$$\prod_{n=0}^{\infty}\begin{pmatrix} a_n & b_n \\ 1 & 0 \end{pmatrix}$$

corresponding to the continued fraction $[\langle a_n.b_n \rangle_{n=0}^{\infty}]$ is only a general normal product if

$$\begin{aligned}
a_0 &\in \mathbb{Z} \\
b_0 &\in \mathbb{Z} \\
\forall n \in \mathbb{N} \cdot a_{n+1} &\in \mathbb{N} \\
\forall n \in \mathbb{N} \cdot b_{n+1} &\in \mathbb{N} - \{0\}.
\end{aligned}$$

Otherwise, a sequence of matrices $M_n \in \mathbb{M}$ for all $n \in \mathbb{N}$ must be found satisfying

$$M_n^{-1} \bullet \begin{pmatrix} a_{n+1} & b_{n+1} \\ 1 & 0 \end{pmatrix} \bullet M_{n+1} \in \mathbb{M}^+$$

for all $n \in \mathbb{N}$. Therefore

$$[\langle a_n.b_n \rangle_{n=0}^{\infty}] = \left(\begin{pmatrix} a_0 & b_0 \\ 1 & 0 \end{pmatrix} \bullet M_0\right)\prod_{n=0}^{\infty}\left(M_n^{-1} \bullet \begin{pmatrix} a_{n+1} & b_{n+1} \\ 1 & 0 \end{pmatrix} \bullet M_{n+1}\right).$$

The definition of a general normal product precludes the use of the vector $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and any matrix of the form $\begin{pmatrix} a & 0 \\ b & 0 \end{pmatrix}$ or $\begin{pmatrix} 0 & c \\ 0 & d \end{pmatrix}$. In practice, this is awkward to enforce. So, we could alternatively allow these vectors and matrices and transfer the problem to the reduction rules rather than the representation. For instance, assimilation could be prescribed by the reduction rules

$$\begin{array}{rcl} M\left\{V^+\right\} & \rightarrow & \left(M \bullet V^+\right) \\ M\left\{N^+\left\{E\right\}\right\} & \rightarrow & \left(M \bullet N^+\right)\left\{E\right\} \end{array}$$

where $M \in \mathbb{M}$, $V^+ \in \mathbb{V}^+$ and $N^+ \in \mathbb{M}^+$. In other words, the vector $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and any matrix of the form $\begin{pmatrix} a & 0 \\ b & 0 \end{pmatrix}$ or $\begin{pmatrix} 0 & c \\ 0 & d \end{pmatrix}$ must not be absorbed directly. This is the approach taken in the theoretical languages of chapter 13 because the programmer cannot be syntactically prevented from using such vectors and matrices.

9.1 Unbiased Exact Floating Point

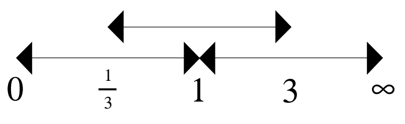
General normal products can be used to elegantly represent algorithms derived from the theory of continued fractions [59]. However, as a general tool for representing extended real numbers it is extremely difficult to control the size of the integers and the flow of information. Of course, the redundant positional representation does not suffer from these drawbacks. In section 8.4, it was suggested that the conjugate of the digit map of the redundant positional representation would be a suitable choice for an incremental digit representation with the special base interval $[0, \infty]$. In other words, the digit map $d \mapsto {}^bD_d$ where

$$ {}^bD_d = \begin{pmatrix} b+1+d & b-1+d \\ b-1-d & b+1-d \end{pmatrix}. \tag{9.2}$$

For radix 2, this gives rise to the three *digit matrices* [41, 51, 62, 20]

$$D_0 = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

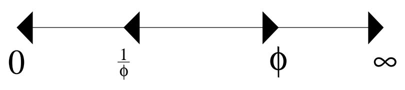
$$D_{-1} = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$$



$$D_1 = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}.$$

For convenience, D_d will be used to denote ${}^2D_d$. For radix $\phi = \frac{\sqrt{5}+1}{2}$ (i.e. the *golden ratio*), this gives rise to the two digit matrices

$${}^\phi D_{-\frac{1}{\phi}} = \begin{pmatrix} 1 & 0 \\ \frac{1}{\phi} & \phi \end{pmatrix}$$



$${}^\phi D_{\frac{1}{\phi}} = \begin{pmatrix} \phi & \frac{1}{\phi} \\ 0 & 1 \end{pmatrix}.$$

The golden ratio provides an interesting radix because the rules for addition and subtraction using the positional representation for the real numbers is extremely elegant as testified by Di Gianantonio [12]. If hardware assistance was provided at bit level for this radix then it would provide an extremely competitive alternative to radix 2. Alternatively, each coefficient could be stored as a pair of integers (a, b) representing $a\phi + b$ with

$$\begin{aligned}(a,b)+(c,d) &= (a+c, b+d)\\ (a,b)\times(c,d) &= (ac+ad+bc, ac+bd)\end{aligned}$$

and

$$\begin{aligned}{}^{\phi}D_{-\frac{1}{\phi}} &= \begin{pmatrix}(1,0) & (0,0)\\ (0,1) & (1,1)\end{pmatrix}\\ {}^{\phi}D_{\frac{1}{\phi}} &= \begin{pmatrix}(1,1) & (0,1)\\ (0,0) & (1,0)\end{pmatrix}.\end{aligned}$$

For example

$$\begin{aligned}(\phi+2)\times(3\phi+4) &\equiv (1,2)\times(3,4)\\ &= (1\times 3+1\times 4+2\times 3, 1\times 3+2\times 4)\\ &= (13,11)\\ &\equiv 13\phi+11\end{aligned}$$

and

$$\begin{pmatrix}(1,0) & (0,0)\\ (0,1) & (1,1)\end{pmatrix}\bullet\begin{pmatrix}(1,0) & (0,0)\\ (0,1) & (1,1)\end{pmatrix}=\begin{pmatrix}(1,1) & (0,0)\\ (2,1) & (3,2)\end{pmatrix}.$$

The base ϕ corresponds to the group of Möbius transformations $\mathcal{M}\left(\mathbb{Q}\left(\sqrt{5}\right)\right)$ with coefficients taken from the quadratic field $\mathbb{Q}\left(\sqrt{5}\right)$ for $\sqrt{5}$.

If we define

$$ {}^{b}\mathfrak{D}^{n}_{c} = \begin{pmatrix} b^n+c+1 & b^n+c-1\\ b^n-c-1 & b^n-c+1\end{pmatrix} \tag{9.3}$$

then it follows from the basic properties of the redundant positional representation that

$$ {}^{b}D_{d_1}{}^{b}D_{d_2}\ldots{}^{b}D_{d_n} = {}^{b}\mathfrak{D}^{n}_{c} \tag{9.4}$$

where

$$c=\sum_{i=1}^{n} d_i b^{n-i}.$$

For convenience, $\mathfrak{D}_c^n$ will be used to denote ${}^2\mathfrak{D}_c^n$. The implication of this identity is that any sequence of n radix 2 digit matrices can be compressed into $n+1$ bits of memory. For example, the two's complement representation of c. Note that

$$\begin{aligned} c &\in \mathbb{Z}(b^n) \\ {}^b\mathfrak{D}_c^n \bullet {}^bD_d &\equiv {}^b\mathfrak{D}_{bc+d}^{n+1}. \end{aligned}$$

Beware, however, that the original sequence of digits cannot usually be recovered.

Definition 39 *The* ***unsigned radix*** $b \in \mathbb{N}$ ***exact floating point representation*** *on* $[0,\infty]$ *is the restricted incremental digit representation*

$$\left([0,\infty], {}^bD_{\mathbb{Z}(b)}, \Psi, \mathbb{V}^+, \Phi\right).$$

This representation is called floating point because a sequence of digits can be divided into two parts in a fashion reminiscent of an exponent and a mantissa. An infinite sequence of ${}^bD_{b-1}$ digits represents ∞. Any other infinite sequence of digits can be identified as a finite sequence of n ${}^bD_{b-1}$ digits (exponent) followed by an infinite sequence of digits not starting with a ${}^bD_{b-1}$ digit (mantissa). In particular, the mantissa part represents a real number in

$$\bigcup_{d\in\mathbb{Z}(b)-\{b-1\}} \mathsf{info}\left({}^bD_d\right) = [0, 2b-1],$$

while the exponent part maps this number into the interval

$${}^bD_{b-1}^n([0,2b-1]) = \left[b^n-1, 2b^{n+1}-1\right].$$

The following proposition is useful for deciding how many more digits are required when the exponent plus two digits of an exact floating point number is known and a certain absolute tolerance is required.

Proposition 40 *For any* $e \in \mathbb{N}$, $\alpha \in \{-1,0\}$, $\beta \in \{-1,0,1\}$ *and* $\gamma \in \mathbb{Z}$, *if*

$$n = 2e + 1 - \gamma + (1+\alpha)(1+\beta)$$

then

$$2^{\gamma-1} < \mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_\alpha \bullet D_\beta \bullet \mathfrak{D}_c^n\right)\right) < 2^{\gamma+1}$$

for all $c \in \mathbb{Z}(2^n)$.

Proof: Note that

$$\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet \mathfrak{D}_c^n\right)\right) = \frac{4\times 2^{e+n}}{(2^n-c)^2-1}.$$

So, by considering end points

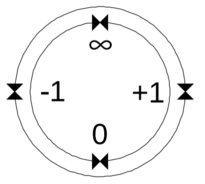
$$\begin{aligned}
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_{-1} \bullet \mathfrak{D}_c^{n'}\right)\right) &> 2^{e-n'-1}\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_{-1} \bullet \mathfrak{D}_c^{n'}\right)\right) &< 2^{e-n'+1}\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_0 \bullet D_{-1} \bullet \mathfrak{D}_c^{n'}\right)\right) &= \mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_{-1} \bullet D_1 \bullet \mathfrak{D}_c^{n'}\right)\right)\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_0 \bullet D_{-1} \bullet \mathfrak{D}_c^{n'}\right)\right) &> 2^{e-n'-2}\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_0 \bullet D_{-1} \bullet \mathfrak{D}_c^{n'}\right)\right) &< 2^{e-n'}\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_0 \bullet D_0 \bullet \mathfrak{D}_c^{n'}\right)\right) &> 2^{e-n'-1}\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_0 \bullet D_0 \bullet \mathfrak{D}_c^{n'}\right)\right) &< 2^{e-n'+1}\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_0 \bullet D_1 \bullet \mathfrak{D}_c^{n'}\right)\right) &= \mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_1 \bullet D_{-1} \bullet \mathfrak{D}_c^{n'}\right)\right)\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_0 \bullet D_1 \bullet \mathfrak{D}_c^{n'}\right)\right) &> 2^{e-n'}\\
\mathsf{width}\left(\mathsf{info}\left(D_1^e \bullet D_0 \bullet D_1 \bullet \mathfrak{D}_c^{n'}\right)\right) &< 2^{e-n'+2}
\end{aligned}$$

- Case $\alpha = -1$.
$$n = 2e + 1 - \gamma$$
$$n' = e - \gamma$$
- Case $\alpha = 0$ and $\beta = -1$.
$$n = 2e + 1 - \gamma$$
$$n' = e - \gamma - 1$$
- Case $\alpha = 0$ and $\beta = 0$.
$$n = 2e + 2 - \gamma$$
$$n' = e - \gamma$$
- Case $\alpha = 0$ and $\beta = 1$.
$$n = 2e + 3 - \gamma$$
$$n' = e + 1 - \gamma \blacksquare$$

There are no "natural" constants in physics, goes the maxim, except zero, one and infinity. Therefore, the most natural choice for redundantly dividing up the one-point compactification of the real numbers is the four intervals $[0, \infty]$, $[1, -1]$, $[\infty, 0]$ and $[-1, 1]$. These intervals can be represented by the four *sign matrices*

$$S_\infty = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

$$S_- = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$



$$S_+ = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$S_0 = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

which conveniently form a cyclic group of order 4 [62, 20, 50].

Definition 41 *The* ***signed radix*** $b \in \mathbb{N}$ ***exact floating point representation*** *on* $\mathbb{R}^\infty$ *be the unrestricted incremental digit representation*

$$\left([0,\infty]\,, S_{\{+,\infty,-,0\}}, \Psi, {}^{b}D_{\mathbb{Z}(b)}, \Psi, \mathbb{V}^+, \Phi\right).$$

Other sign matrices might be considered as discussed in section 8.4, but in this thesis we will concentrate on $S_{\{+,\infty,-,0\}}$.

9.2 Biased Exact Floating Point

For biased exact floating point, matrices are chosen such that addition and subtraction are particularly efficient. The simple automaton illustrated in figure 9.2 where $n \in \mathbb{Z}(b^\gamma) - \mathbb{Z}(2)$, $d \in \mathbb{Z}(b)$ and

$$\begin{aligned}
S &= \begin{pmatrix} 1 & 1 \\ -b & b \end{pmatrix} \\
P &= \begin{pmatrix} b & -b \\ 1 & 1 \end{pmatrix} \\
R &= \begin{pmatrix} b+1 & b-1 \\ b-1 & b+1 \end{pmatrix} \\
T_n &= \begin{pmatrix} n & n-2 \\ n+2 & n \end{pmatrix} \\
Q_n &= {}^{b^\gamma}D_n
\end{aligned}$$

corresponds to the fully incremental γ-normalized exact radix b floating point representation as defined in section 7.4 and illustrated in figure 7.3. Recall that γ is typically 1 except when $b = 2$ in which case it is typically 2. Observe that

$$\begin{aligned}
\mathsf{info}\,(SR^e) &= \mathsf{info}\begin{pmatrix} b^e & b^e \\ -b & b \end{pmatrix} \\
&= [1, -1] \times b^{e-1} \\
\mathsf{info}\,(SR^eT_n) &= \mathsf{info}\begin{pmatrix} (n+1)\,b^e & (n-1)\,b^e \\ b & b \end{pmatrix} \\
&= [n-1, n+1] \times b^{e-1} \\
\mathsf{info}\,(PR^e) &= \mathsf{info}\begin{pmatrix} b & -b \\ b^e & b^e \end{pmatrix} \\
&= [-1, 1] \times b^{-e+1} \\
\mathsf{info}\,(PR^eQ_n) &= \mathsf{info}\begin{pmatrix} n+1 & n-1 \\ b^{\gamma-1+e} & b^{\gamma-1+e} \end{pmatrix} \\
&= [n-1, n+1] \times b^{-e-\gamma+1}
\end{aligned}$$

for exponent $e \in \mathbb{N}$ as required. Recall figure 7.4 which illustrates the general properties of this representation for $\gamma = 2$ and $b = 2$.

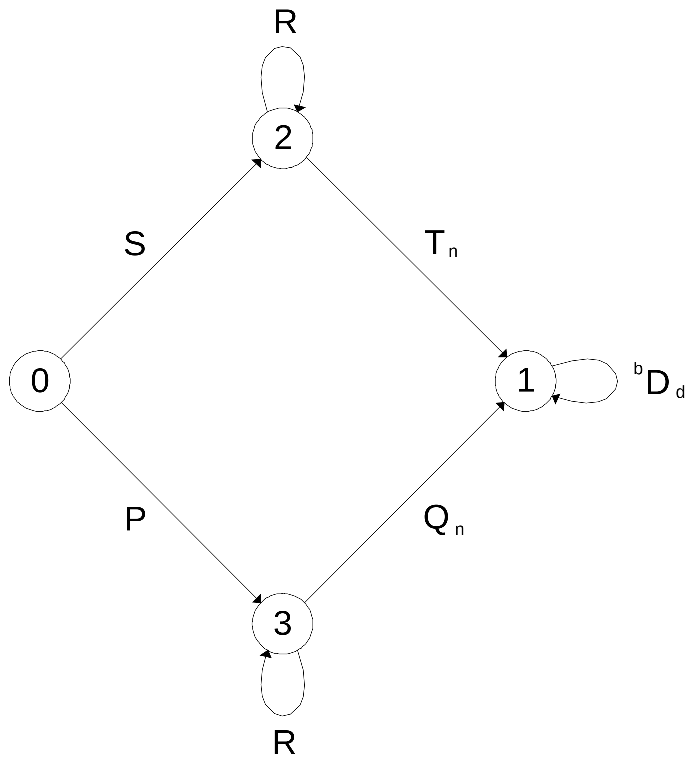


Figure 9.2: The automaton for biased radix b exact floating point.

Chapter 10

Expression Trees

In this chapter, we define the notion of an expression tree and use it to represent real numbers and real functions.

Definition 42 *An **unsigned expression tree** is a binary tree. Each node may be either*

- *a tensor $T \in \mathbb{T}^+$ with 2 children or*
- *a matrix $M \in \mathbb{M}^+$ with 1 child or*
- *a vector $V \in \mathbb{V}^+$ with no children.*

Definition 43 *A **signed expression tree** is a tree consisting of either*

- *a tensor $T \in \mathbb{T}$ connected to two unsigned expressions trees or*
- *a matrix $M \in \mathbb{M}$ connected to one unsigned expressions tree or*
- *a vector $V \in \mathbb{V}$ with no children.*

Let $\mathbb{E}$ and $\mathbb{E}^+$ denote the set of signed and unsigned expression trees respectively. An expression tree (e.g. as shown in figure 10.1) induces a directed set in the continuous real domain $\mathbb{C}(\mathbb{R}^\infty)$ (e.g. as illustrated in figure 10.2). Each element of the directed set corresponds to a particular *cut* through the expression tree. This element is then the interval derived by plugging each *exposed argument* with the special base interval $[0, \infty]$.

An expression tree represents an extended real number x if the least upper bound of the induced directed set is the singleton set $\{x\}$. The expression tree $E \in \mathbb{E}$ consisting of a root node $L \in \mathbb{L}$ connected to expression trees $E_{\{1,\ldots,N\}} \in \mathbb{E}^+$ with $N \in \{0, 1, 2\}$ will be denoted by $L\{E_1, \ldots, E_N\}$. Note that the general normal product representation and exact floating point representation are just special expression trees.

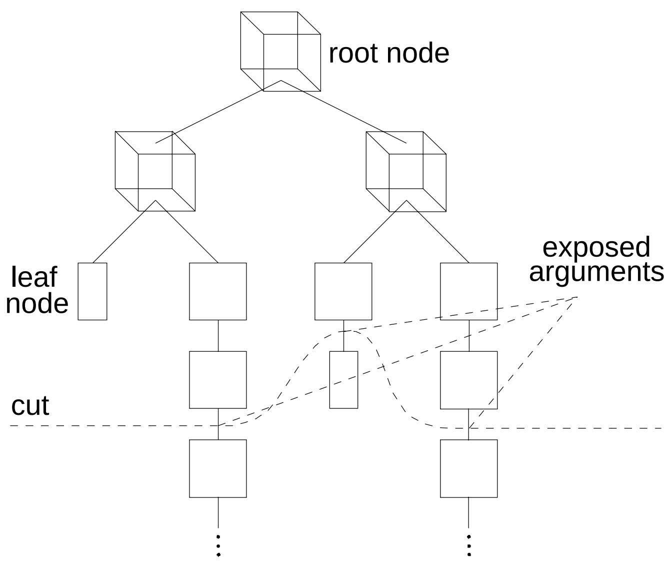


Figure 10.1: A typical expression tree

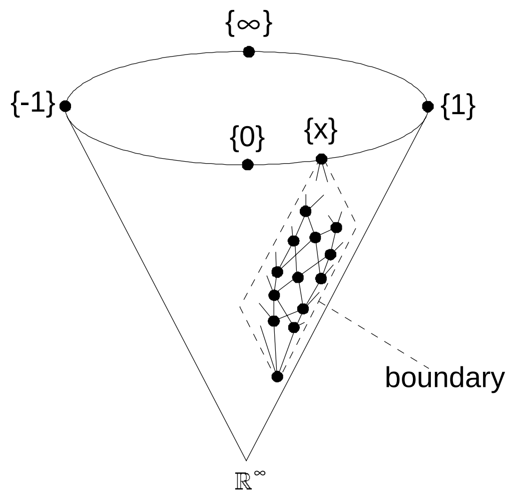


Figure 10.2: A directed set representing the extended real number x on the continuous real domain $\left(\mathbb{I}^{\mathrm{C}}\left(\mathbb{R}^{\infty}\right), \supseteq\right)$ induced by an expression tree.

10.1 Basic Arithmetic Operations

10.1.1 Matrix Application

The (square bracket) application of a matrix $M \in \mathbb{M}$ to a signed expression tree $L\{E_1, \dots, E_N\} \in \mathbb{E}$ can be reduced to the signed expression tree with the root node $M \bullet L$ connected to the same unsigned expression trees $E_{\{1,\dots,N\}}$

$$M[L\{E_1, \dots, E_N\}] = (M \bullet L)\{E_1, \dots, E_N\}. \tag{10.1}$$

10.1.2 Reciprocal and Negation

Reciprocation rec and negation neg are achieved by matrix square bracket application

$$\begin{aligned} \mathsf{rec}(E) &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}[E] \\ \mathsf{neg}(E) &= \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}[E]. \end{aligned}$$

A purely symbolic algorithm is possible if we restrict to signed exact floating point

$$\begin{aligned} \mathsf{rec}(S_+\{E\}) &= S_+\{\mathsf{rec}(E)\} \\ \mathsf{rec}(S_-\{E\}) &= S_-\{\mathsf{rec}(E)\} \\ \mathsf{rec}(S_0\{E\}) &= S_\infty\{\mathsf{rec}(E)\} \\ \mathsf{rec}(S_\infty\{E\}) &= S_0\{\mathsf{rec}(E)\} \\ \mathsf{rec}\left({}^bD_d\{E\}\right) &= {}^bD_{-d}\{\mathsf{rec}(E)\} \\ \mathsf{neg}(S_+\{E\}) &= S_-\{\mathsf{rec}(E)\} \\ \mathsf{neg}(S_-\{E\}) &= S_+\{\mathsf{rec}(E)\} \\ \mathsf{neg}(S_0\{E\}) &= S_0\{\mathsf{rec}(E)\} \\ \mathsf{neg}(S_\infty\{E\}) &= S_\infty\{\mathsf{rec}(E)\}. \end{aligned}$$

10.1.3 Tensor Application

The (square bracket) application of a tensor $T \in \mathbb{T}$ to two signed expression trees $K\{E_1, \dots, E_M\}$, $L\{F_1, \dots, F_N\} \in \mathbb{E}$ with $K, L \in \mathbb{V} \cup \mathbb{M}$ and $M, N \in \{0, 1\}$ can be reduced to the signed expression tree with the root node $T \bullet_1 K \bullet_2 L$ connected to the same unsigned expression trees $E_{\{1,\dots,M\}}$ and $F_{\{1,\dots,N\}}$

$$\begin{aligned} & T[K\{E_1, \dots, E_M\}, L\{F_1, \dots, F_N\}] \\ = \ & ((T \bullet_2 L) \bullet_1 K)\{E_1, \dots, E_M, F_1, \dots, F_N\}. \end{aligned} \tag{10.2}$$

Note that a signed expression tree with root node $T \in \mathbb{T}$ can be converted into a matrix connected to an unsigned expression tree or a vector using the reduction rules in the section 11.

10.1.4 Addition, Subtraction, Multiplication and Division

The basic arithmetic operations $+$, $-$,$\times$ and $\div$ are achieved by tensor square bracket application as pointed out by Gosper [28]. The actual tensors required are given in equations (8.7), (8.8), (8.9) and (8.10). For example, the signed expression tree corresponding to the subtraction of the two signed exact floating point numbers $S_\sigma \{D_{d_1} \{D_{d_2} \{\cdots\}\}\}$ and $S_\rho \{D_{e_1} \{D_{e_2} \{\cdots\}\}\}$ is

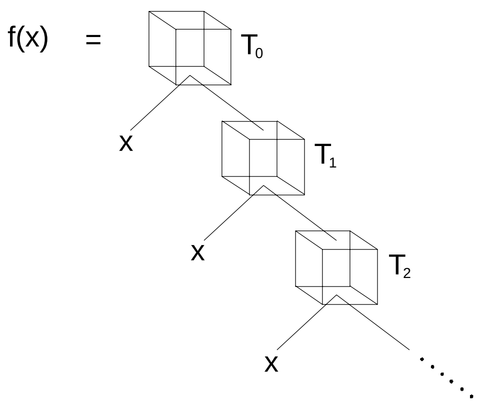
$$(T_- \bullet_1 S_\sigma \bullet_2 S_\rho) \{D_{d_1} \{D_{d_2} \{\cdots\}\}, D_{e_1} \{D_{e_2} \{\cdots\}\}\}.$$

10.2 Elementary Functions

The Taylor series of a function $f(x)$ can often be used to derive a number of continued fractions with the general form

$$f(x) = \alpha_0(x) + \cfrac{\beta_0(x)}{\alpha_1(x) + \cfrac{\beta_1(x)}{\alpha_2(x) + \cfrac{\beta_2(x)}{\ddots}}} \tag{10.3}$$

including the Stieltjes, Jacobi and Euler types [73, 4, 59]. Sometimes these continued fractions can be converted into an expression tree



(10.4)

for some $T_0 \in \mathbb{T}$ and $T_n \in \mathbb{T}^+$ for all $n \geq 1$ [59]. A general procedure for this is:

Part (i) Using the simple matrix identities

$$\begin{pmatrix} a & c\mu \\ b & 0 \end{pmatrix}\begin{pmatrix} d & f \\ e & 0 \end{pmatrix} = \begin{pmatrix} a & c \\ b & 0 \end{pmatrix}\begin{pmatrix} d & f \\ e\mu & 0 \end{pmatrix} \tag{10.5}$$

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix}\begin{pmatrix} e & g \\ f & h \end{pmatrix} = \begin{pmatrix} c & a \\ d & b \end{pmatrix}\begin{pmatrix} f & h \\ e & g \end{pmatrix} \tag{10.6}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\prod_{n=0}^{\infty}\begin{pmatrix} a_n & c_n \\ b_n & d_n \end{pmatrix} = \prod_{n=0}^{\infty}\begin{pmatrix} d_n & b_n \\ c_n & a_n \end{pmatrix} \tag{10.7}$$

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} \equiv \begin{pmatrix} a\lambda & c\lambda \\ b\lambda & d\lambda \end{pmatrix} \text{ for } \lambda \neq 0 \tag{10.8}$$

find

$$T_n = \begin{pmatrix} a_n & c_n & e_n & g_n \\ b_n & d_n & f_n & h_n \end{pmatrix} \in \mathbb{T}$$

such that

$$\prod_{n=0}^{\infty}\begin{pmatrix} \alpha_n(x) & \beta_n(x) \\ 1 & 0 \end{pmatrix} \equiv \prod_{n=0}^{\infty}\begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix}.$$

No particular method was used for doing this.

Part (ii) Find a sequence of matrices $M_n \in \mathbb{M}$ for $n \geq 0$ and a matrix N such that

$$M_{n-1}^{-1} \bullet T_n \bullet_1 N \bullet_2 M_n \in \mathbb{T}^+$$

for all $n \geq 1$. In which case

$$\begin{aligned} f(N(y)) &= (T_0 \bullet_1 N \bullet_2 M_0)\{y, E_1(y)\} \text{ where} \\ E_n(y) &= \left(M_{n-1}^{-1} \bullet T_n \bullet_1 N \bullet_2 M_n\right)\{y, E_{n+1}(y)\} \end{aligned}$$

for all $y \in [0, \infty]$. Note that the matrix N essentially limits the domain of the function to $\mathsf{info}(N)$.

The Backward Theorem (Theorem 17) must be used to justify these transformations as illustrated later. We will use the general procedure outlined above in the following subsections to derive algorithms for the basic set of transcendental functions; namely square root, natural logarithm, exponential, tangent and inverse tangent. The other transcendental functions can be derived from this basic set using the identities:

$$x^y = \mathsf{exp}(y \,\mathsf{log}(x))$$

$$\begin{aligned}
\mathsf{sin}(x) &= \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \left[\mathsf{tan}\left(\frac{x}{2}\right), \mathsf{tan}\left(\frac{x}{2}\right)\right] \\
\mathsf{cos}(x) &= \begin{pmatrix} -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \left[\mathsf{tan}\left(\frac{x}{2}\right), \mathsf{tan}\left(\frac{x}{2}\right)\right] \\
\mathsf{arcsin}(x) &= \mathsf{arctan}\left(\sqrt{\frac{x^2}{1-x^2}}\right) \\
\mathsf{arccos}(x) &= \mathsf{arctan}\left(\sqrt{\frac{1-x^2}{x^2}}\right) \\
\mathsf{sinh}(x) &= \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix} [\mathsf{exp}(x), \mathsf{exp}(x)] \\
\mathsf{cosh}(x) &= \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} [\mathsf{exp}(x), \mathsf{exp}(x)] \\
\mathsf{tanh}(x) &= \begin{pmatrix} 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \end{pmatrix} [\mathsf{exp}(x), \mathsf{exp}(x)] \\
\mathsf{arcsinh}(x) &= \mathsf{log}\left(x + \sqrt{x^2+1}\right) \\
\mathsf{arccosh}(x) &= \mathsf{log}\left(x + \sqrt{x^2-1}\right) \\
\mathsf{arctanh}(x) &= \frac{1}{2}\mathsf{log}\left(S_\infty[x]\right).
\end{aligned}$$

The complexity of the following algorithms have been favorably analyzed by Heckman [32, 34, 33].

10.2.1 Square Root

A simple algorithm for the square root $\sqrt{n}$ of a positive integer n is recursively given by

$$\sqrt{n} = \begin{pmatrix} \lfloor \sqrt{n} \rceil & n \\ 1 & \lfloor \sqrt{n} \rceil \end{pmatrix} \{\sqrt{n}\}.$$

This algorithm is related to Newton Raphson's iterative method for finding a root.

$$x_{n+1} \longmapsto x_n - \frac{f(x_n)}{f'(x_n)}$$

In particular, for $f(x) = x^2 - n$, we have

$$x_{n+1} \longmapsto \frac{x_n^2 + n}{2x_n}.$$

Compare this with

$$\begin{aligned}\begin{pmatrix} x_n & n \\ 1 & x_n \end{pmatrix} \bullet \begin{pmatrix} x_n & n \\ 1 & x_n \end{pmatrix} &= \begin{pmatrix} x_n^2 + n & 2x_n n \\ 2x_n & x_n^2 + n \end{pmatrix} \\ &\equiv \begin{pmatrix} \frac{x_n^2+n}{2x_n} & n \\ 1 & \frac{x_n^2+n}{2x_n} \end{pmatrix}.\end{aligned}$$

Incidently, this makes it clear why we choose $x_0 = \lfloor \sqrt{n} \rceil$. This formula is easily generalized to positive rational numbers by

$$\sqrt{\frac{p}{q}} = \begin{pmatrix} \lfloor \sqrt{pq} \rceil & p \\ q & \lfloor \sqrt{pq} \rceil \end{pmatrix} \left\{ \sqrt{\frac{p}{q}} \right\}.$$

The efficiency of the algorithm is related to the determinant of the matrix simply because

$$\mathsf{width}\left(\mathsf{info}\begin{pmatrix} a & c \\ b & d \end{pmatrix}\right) = \frac{\mathsf{det}\begin{pmatrix} a & c \\ b & d \end{pmatrix}}{bd}.$$

Clearly, the ideal determinant has absolute value 1 and in which case the formula will be directly related to a simple continued fraction. For example, consider $\sqrt{2}$

$$\begin{aligned}\sqrt{2} &= [1, \langle 2 \rangle_{n=0}^{\infty}] \\ &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \left\langle \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle_{n=0}^{\infty} \\ &= \left\langle \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix} \right\rangle_{n=0}^{\infty}\end{aligned}$$

because

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \bullet \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \bullet \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{-1} \equiv \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}.$$

Let us now consider the square root of any positive real number. Let $\mathsf{loop}\,(T, x)$ be the function represented by the expression tree in equation (10.4) with $T_n = T$ for all $n \geq 0$. It can be shown, for example, that

$$\sqrt{x} = \mathsf{loop}\,(T, x)$$

where

$$T = \begin{pmatrix} 1 & m & 1 & 0 \\ 0 & 1 & m & 1 \end{pmatrix}$$

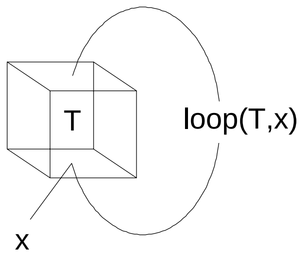
for any $m \in \mathbb{N} - \{0\}$. The contractivity of a matrix M is computed by Reinhold Heckmann [34] as

$$\begin{aligned} \Xi(M) &= \mathsf{sup}\left(\left|\left(S_0 \bullet M \bullet S_0^{\dagger}\right)'([-1,1])\right|\right) \\ &= \frac{|\mathsf{det}(M)|}{(\mathsf{min}(|a+b|, |c+d|))^2} \end{aligned}$$

and he showed that the efficiency of a matrix is inversely related to its contractivity and using this he found that

$$\sqrt{x} = \mathsf{loop}\left(\left(\begin{array}{cccc} 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{array}\right), x\right)$$

is the best general formula for square root. The usual algorithm for converting an expression tree into an exact floating point number can be improved dramatically for $\mathsf{loop}(T, x)$ by employing a feedback mechanism.



For $\mathsf{loop}(T, x)$ with $T \in \mathbb{T}^+$ and $x \in [0, \infty]$:

1. Ensure x is an unsigned exact floating point number.
2. While $D_d^{\dagger} \bullet T \bullet_2 D_d \in \mathbb{T}^+$ for some $d \in \{-1, 0, 1\}$, emit digit D_d and set T to $D_d^{\dagger} \bullet T \bullet_2 D_d$.
3. Absorb a digit D_d from x, set T to $T \bullet_1 D_d$ and goto 2.

This algorithm for $\mathsf{loop}(T, x)$ can be improved even further if x is a rational number. For $\mathsf{loop}(T, V)$ with $T \in \mathbb{T}^+$ and $V \in \mathbb{V}^+$:

1. Set M to $T \bullet_1 V$.
2. If $D_{-1}^{\dagger} \bullet M \bullet D_{-1} \in \mathbb{M}^+$ then set d to -1 else set d to 1.
3. Emit digit D_d, set M to $D_d^{\dagger} \bullet M \bullet D_d$ and goto 2.

Recall that the tame inverse $M^\dagger$ of a matrix M is defined in equation (8.4). In fact, Reinhold Heckmann [31] has improved this algorithm for the square root of a rational number even further

$$\begin{aligned}\sqrt{\frac{p}{q}} &= \mathsf{rollover}\,(p, q, p-q) \\ \mathsf{rollover}\,(a,b,c) &= \begin{cases} D_{-1}\,\{\mathsf{rollover}\,(4a, d, c)\} & \text{if } d \geq 0 \\ D_1\,\{\mathsf{rollover}\,(-d, 4b, c)\} & \text{otherwise}\end{cases}\end{aligned}$$

where

$$d = 2\,(b-a) + c.$$

10.2.2 Logarithm

The Stieltjes type continued fraction for logarithm [4] is

$$\mathsf{log}\,(1+x) = \left[0.x, 1.\frac{x}{2}, \left\langle 1.\frac{nx}{4n+2}, 1.\frac{(n+1)\,x}{4n+2}\right\rangle_{n\geq 1}\right]. \tag{10.9}$$

Note that

$$\begin{aligned} & \begin{pmatrix} 0 & x \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & \frac{x}{2} \\ 1 & 0 \end{pmatrix}\prod_{n=1}^{\infty}\begin{pmatrix} 1 & \frac{nx}{4n+2} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & \frac{(n+1)x}{4n+2} \\ 1 & 0 \end{pmatrix} \\ = & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & x \\ x & 0 \end{pmatrix}\prod_{n=1}^{\infty}\begin{pmatrix} 1 & \frac{n}{2} \\ \frac{n}{2} & 0 \end{pmatrix}\begin{pmatrix} 1 & \frac{x}{2n+1} \\ \frac{x}{2n+1} & 0 \end{pmatrix} \text{ by (10.5)} \\ \equiv & \prod_{n=0}^{\infty}\begin{pmatrix} 0 & x \\ x & 2n+1 \end{pmatrix}\begin{pmatrix} 0 & n+1 \\ n+1 & 2 \end{pmatrix} \text{ by (10.8) and (10.7).}\end{aligned}$$

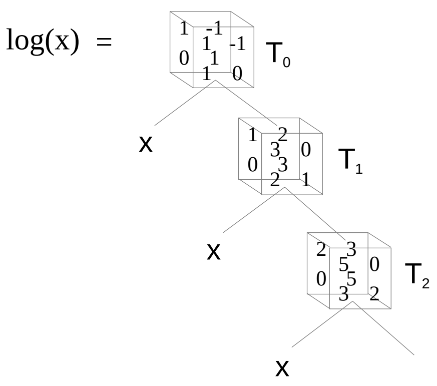
Therefore, according to part (i), the corresponding sequence of tensors is

$$T_n = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2n+1 \end{pmatrix} \bullet_2 \begin{pmatrix} 0 & n+1 \\ n+1 & 2 \end{pmatrix}.$$

Since $T_n \in \mathbb{T}^+$ for all $n \geq 0$, this immediately gives an expression tree for $\mathsf{log}\,(x)$ valid for all $x \in [1, \infty]$. However, we can do better. Note that for $M_n = \begin{pmatrix} 1 & -1 \\ 0 & n+1 \end{pmatrix}$ and $N = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$, $M_{n-1}^{-1} \bullet T_n \bullet_1 N \bullet_2 M_n \in \mathbb{T}^+$ for all $n \geq 1$. Therefore, according to part (ii), we have

$$\begin{aligned}\mathsf{log}\,(x) &= \begin{pmatrix} 1 & 1 & -1 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}\{x, E_1(x)\} \\ E_n(x) &= \begin{pmatrix} n & 2n+1 & n+1 & 0 \\ 0 & n+1 & 2n+1 & n \end{pmatrix}\{x, E_{n+1}(x)\}\end{aligned}$$

for all $x \in [0, \infty]$. Pictorially, we have



This expression tree is only efficient for x close to 1. However, the domain can easily be reduced to $\left[\frac{1}{2}, 2\right]$ by repeated application of the identity

$$\mathsf{log}\,(x) = \mathsf{log}\left(\frac{x}{2}\right) + \mathsf{log}\,(2)\,.$$

Using the identity

$$\mathsf{arctanh}\,(x) = \frac{1}{2}\mathsf{log}\left(\frac{1+x}{1-x}\right)$$

and the cyclic properties of the sign set $\{S_+, S_\infty, S_-, S_0\}$, inverse hyperbolic tangent can be given by

$$\begin{aligned}
\mathsf{arctanh}\,(S_+\{E\}) &= \frac{1}{2}\mathsf{log}\,(S_\infty\{E\}) \\
\mathsf{arctanh}\,(S_\infty\{E\}) &= \frac{1}{2}\mathsf{log}\,(S_-\{E\}) \\
\mathsf{arctanh}\,(S_-\{E\}) &= \frac{1}{2}\mathsf{log}\,(S_0\{E\}) \\
\mathsf{arctanh}\,(S_0\{E\}) &= \frac{1}{2}\mathsf{log}\,(S_+\{E\})\,.
\end{aligned}$$

As an aside, note that the expression tree can be collapsed into quadratic fractional transformations as defined in section 8.7

$$E_n(x) = \left(\begin{array}{c} (n^2+n)\,x^2 + (6n^2+12n+4)\,x + (n^2+3n+2) \\ (4n^2+8n+4)\,x + (4n^2+8n+2) \\[1ex] (4n^2+8n+2)\,x^2 + (4n^2+8n+4)\,x \\ (n^2+3n+2)\,x^2 + (6n^2+12n+4)\,x + (n^2+n) \end{array}\right) E_{n+2}(x)$$

$$= \left(\begin{pmatrix} n^2+n & 6n^2+12n+4 & n^2+3n+2 \\ 0 & 4n^2+8n+4 & 4n^2+8n+2 \end{pmatrix}, \begin{pmatrix} 4n^2+8n+2 & 4n^2+8n+4 & 0 \\ n^2+3n+2 & 6n^2+12n+4 & n^2+n \end{pmatrix} \right) E_{n+2}(x).$$

10.2.3 Exponential

A Jacobi type continued fraction for exponential [24, 25] is

$$\mathsf{exp}(x) = \left[1.x, 1-\frac{x}{2}.\frac{x^2}{12}, \left\langle 1.\frac{x^2}{16n^2-4} \right\rangle_{n\geq 2} \right]. \tag{10.10}$$

Note that

$$\begin{aligned} & \begin{pmatrix} 1 & x \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1-\frac{x}{2} & \frac{x^2}{12} \\ 1 & 0 \end{pmatrix} \prod_{n=2}^{\infty} \begin{pmatrix} 1 & \frac{x^2}{16n^2-4} \\ 1 & 0 \end{pmatrix} \\ = & \begin{pmatrix} 1 & x \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1-\frac{x}{2} & \frac{x}{2} \\ 1 & 0 \end{pmatrix} \prod_{n=2}^{\infty} \begin{pmatrix} 1 & \frac{x}{(4n-2)} \\ \frac{x}{(4n-2)} & 0 \end{pmatrix} \text{ by (10.5)} \\ \equiv & \begin{pmatrix} 2+x & x \\ 2-x & x \end{pmatrix} \prod_{n=1}^{\infty} \begin{pmatrix} 4n+2 & x \\ x & 0 \end{pmatrix} \text{ by (10.8).} \end{aligned}$$

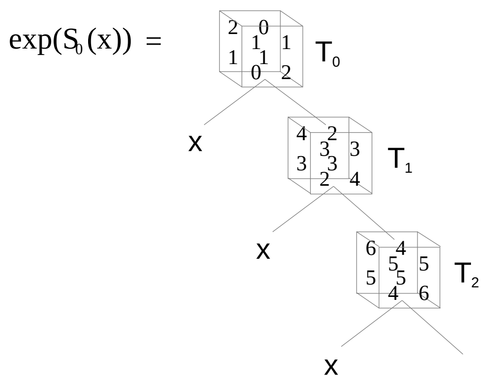
Therefore, according to part (i), the corresponding sequence of tensors is

$$T_n = \begin{cases} \begin{pmatrix} 1 & 1 & 2 & 0 \\ -1 & 1 & 2 & 0 \end{pmatrix} & \text{if } n=0 \\ \begin{pmatrix} 0 & 1 & 4n+2 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} & \text{if } n \geq 1. \end{cases}$$

Note that $S_\infty^{-1} \bullet T_n \bullet_1 S_0 \bullet_2 S_\infty \in \mathbb{T}^+$ for all $n \geq 1$. Therefore, according to part (ii), we have

$$\begin{aligned} \mathsf{exp}(S_0\{x\}) &= E_0(x) \\ E_n(x) &= \begin{pmatrix} 2n+2 & 2n+1 & 2n & 2n+1 \\ 2n+1 & 2n & 2n+1 & 2n+2 \end{pmatrix} \{x, E_{n+1}(x)\} \end{aligned}$$

for all $x \in [0, \infty]$. Pictorially, we have



This deals with exponential for $x \in [-1, 1]$. For x outside this range, repeatedly apply the identity

$$\exp(x) = \left(\exp\left(\frac{x}{2}\right)\right)^2$$

until $x \in [-1, 1]$. A general normal product for the natural number e can be derived

$$\begin{aligned} e &= \prod_{n=0}^{\infty} \begin{pmatrix} 2n+2 & 2n+1 & 2n & 2n+1 \\ 2n+1 & 2n & 2n+1 & 2n+2 \end{pmatrix} \bullet_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \prod_{n=0}^{\infty} \begin{pmatrix} 2n+2 & 2n+1 \\ 2n+1 & 2n \end{pmatrix} \end{aligned}$$

with a good convergence rate because the determinant of each matrix has an absolute value of 1.

10.2.4 Pi

A very fast formula for π can be derived using Ramanujan's formula [30, 59]

$$\frac{1}{\pi} = \sum_{n=0}^{\infty} (-1)^n \frac{12(6n)!}{(n!)^3(3n)!} \frac{545140134n + 13591409}{(640320^3)^{n+\frac{1}{2}}}. \tag{10.11}$$

Ramanujan's formula can be viewed as the instantiation of the Taylor expansion of a function $f(x) = \sum_{n=0}^{\infty} a_n x^n$

$$\begin{aligned} \frac{\sqrt{10005}}{\pi} &= f\left(\frac{1}{640320^3}\right) \\ a_n &= (-1)^n \frac{(6n)!}{(n!)^3(3n)!} \frac{545140134n + 13591409}{426880}. \end{aligned}$$

The general formula for the Euler type continued fraction [73] of a Taylor expansion $f(x) = \sum_{n=0}^{\infty} a_n x^n$ is

$$f(x) = \left[0.a_0, 1. - \frac{a_1}{a_0}x, \left\langle 1 + \frac{a_{n+1}}{a_n}x. - \frac{a_{n+2}}{a_{n+1}} \right\rangle_{n\geq 0} \right].$$

Note that

$$\begin{aligned}
& \begin{pmatrix} 0 & a_0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -\frac{a_1}{a_0}x \\ 1 & 0 \end{pmatrix} \prod_{n=0}^{\infty} \begin{pmatrix} 1 + \frac{a_{n+1}}{a_n}x & -\frac{a_{n+2}}{a_{n+1}}x \\ 1 & 0 \end{pmatrix} \\
= & \begin{pmatrix} 0 & a_0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \prod_{n=0}^{\infty} \begin{pmatrix} 1 + \frac{a_{n+1}}{a_n}x & 1 \\ -\frac{a_{n+1}}{a_n}x & 0 \end{pmatrix} \text{ by (10.5)} \\
\equiv & \begin{pmatrix} a_0 & 0 \\ 1 & 1 \end{pmatrix} \prod_{n=1}^{\infty} \begin{pmatrix} 1 + \frac{a_n}{a_{n-1}}x & 1 \\ -\frac{a_n}{a_{n-1}}x & 0 \end{pmatrix}
\end{aligned}$$

and

$$-\frac{a_n}{a_{n-1}}x = \frac{(2n-1)(6n-5)(6n-1)(545140134n + 13591409)}{10939058860032000n^3(545140134n - 531548725)}.$$

Therefore

$$\begin{aligned}
\frac{\sqrt{10005}}{\pi} &= \begin{pmatrix} 13591409 & 0 \\ 426880 & 426880 \end{pmatrix} \prod_{n=1}^{\infty} \begin{pmatrix} b_n - c_n & b_n \\ c_n & 0 \end{pmatrix} \text{ where} \\
b_n &= 10939058860032000n^3(545140134n - 531548725) \\
c_n &= (2n-1)(6n-5)(6n-1)(545140134n + 13591409).
\end{aligned}$$

We can do even better than this. For

$$M_n = \begin{pmatrix} 545140135n + 13591410 & 545140133n + 13591408 \\ -n-1 & n+1 \end{pmatrix},$$

$M_{n-1}^{-1} \bullet \begin{pmatrix} b_n - c_n & b_n \\ c_n & 0 \end{pmatrix} \bullet M_n \in \mathbb{T}^+$ for all $n \geq 1$. Therefore it can be shown, with some rescaling, that

$$\begin{aligned}
\frac{\sqrt{10005}}{\pi} &= \begin{pmatrix} 6795705 & 6795704 \\ 213440 & 213440 \end{pmatrix} \prod_{n=1}^{\infty} Q_n \text{ where} \\
Q_n &= \begin{pmatrix} e_n - d_n - c_n & e_n + d_n - c_n \\ e_n + d_n + c_n & e_n - d_n + c_n \end{pmatrix} \\
d_n &= (2n-1)(6n-5)(6n-1)(n+1) \\
e_n &= 10939058860032000n^4.
\end{aligned}$$

The convergence rate for this general normal product is extremely impressive. Each matrix corresponds to approximately 14 decimal places of information. Note that the entries of the matrix $Q_n \bullet Q_{n+1}$ are divisible by $2(n+1)$. Therefore the entries of the matrix $\prod_{n=1}^{N} Q_n$ are divisible by at least $N!$. However, we conjecture that the entries of the matrix $\prod_{n=1}^{N} Q_n$ are divisible by $\frac{5(N!)^4}{12^{N-1}}$.

10.2.5 Tangent

A continued fraction by Lambert [44] for tangent is

$$\mathsf{tan}\,(x) = \left[0, \left\langle \frac{4n+1}{x}, -\frac{4n+3}{x} \right\rangle_{n\geq 0}\right]. \tag{10.12}$$

Lemma 44 *The terms in the backward sequence (see definition 16) of the continued fraction*

$$\left[0, \left\langle \frac{4n+1}{x}, -\frac{4n+3}{x} \right\rangle_{n\geq 0}\right]$$

have absolute value greater than one for all $x \in [-1,1]$.

Proof: Let $e_n(x)$ be the n^{th} term in the backward sequence in question. In general, using the definitions in the Backward Theorem (theorem 17)

$$e_{n+2} = a_{n+2} + \frac{b_{n+1}}{e_{n+1}}$$

for all $n \in \mathbb{N}$ with

$$e_1 = a_1.$$

In this particular case

$$\begin{aligned} a_1 &\in [1,-1] \\ a_{n+2} &\in [3,-3] \\ b_n &= 1. \end{aligned}$$

Clearly

$$\begin{aligned} e_1 &\in [1,-1] \\ e_2 &\in [3,-3] + \frac{1}{[1,-1]} = [2,-2]. \end{aligned}$$

So, assuming that

$$e_{n+2} \in [2,-2],$$

it follows that

$$\begin{aligned} e_{n+3} &\in [3,-3] + \frac{1}{[2,-2]} \\ &\subseteq [2,-2] \\ \frac{e_{n+3}}{[-1,1]+e_{n+3}} &\subseteq \left[\tfrac{2}{3},2\right]. \blacksquare \end{aligned}$$

So, using the Backward Theorem, it follows that

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \prod_{n=0}^{\infty} \begin{pmatrix} \frac{4n+1}{x} & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -\frac{4n+3}{x} & 1 \\ 1 & 0 \end{pmatrix}$$

is a sound general normal product for tangent provided that we use the base interval $[-1,1]$ instead of the usual $[0,\infty]$. In other words, we consider the following sequence of nested intervals

$$\begin{aligned} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}([-1,1]) \\ \supseteq\ & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \frac{1}{x} & 1 \\ 1 & 0 \end{pmatrix}([-1,1]) \\ \supseteq\ & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \frac{1}{x} & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} -\frac{3}{x} & 1 \\ 1 & 0 \end{pmatrix}([-1,1]) \\ \supseteq\ & \cdots. \end{aligned}$$

But

$$\begin{aligned} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \prod_{n=0}^{\infty} \begin{pmatrix} \frac{4n+1}{x} & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -\frac{4n+3}{x} & 1 \\ 1 & 0 \end{pmatrix} \\ \equiv\ & \prod_{n=0}^{\infty} \begin{pmatrix} 0 & x \\ x & 4n+1 \end{pmatrix} \begin{pmatrix} 0 & x \\ x & -4n-3 \end{pmatrix} \quad \text{by (10.8) and (10.7)} \end{aligned}$$

and the corresponding sequence of tensors is

$$T_n = \begin{cases} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2n+1 \end{pmatrix} & \text{if } n \text{ even} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -2n-1 \end{pmatrix} & \text{if } n \text{ odd} \end{cases}$$

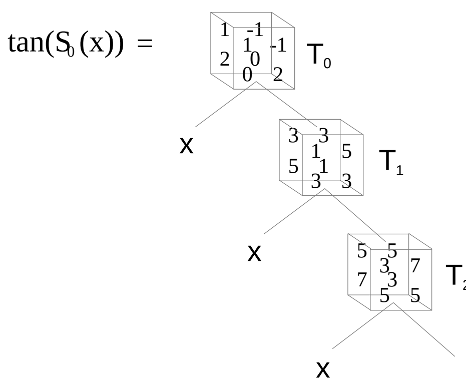
for all $n \geq 0$. Note that $S_0^{-1} \bullet T_n \bullet_1 S_0 \bullet_2 S_0 \in \mathbb{T}^+$ for all $n \geq 1$. This conveniently brings us back to the special base interval $[0,\infty]$ because

$$S_0([0,\infty]) = [-1,1].$$

Therefore, according to part (ii) and the application of equation (10.6), we have

$$\begin{aligned} \mathsf{tan}\left(S_0\left\{x\right\}\right) &= \begin{pmatrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{pmatrix} \left\{x, E_1\left(x\right)\right\} \qquad (10.13) \\ E_n\left(x\right) &= \begin{pmatrix} 2n+1 & 2n-1 & 2n+1 & 2n+3 \\ 2n+3 & 2n+1 & 2n-1 & 2n+1 \end{pmatrix} \left\{x, E_{n+1}\left(x\right)\right\} \end{aligned}$$

for all $x \in [0, \infty]$. Pictorially, we have



This deals with tangent for $x \in [-1, 1]$. For x outside this range, repeatedly apply the trigonometric identity

$$\mathsf{tan}\left(x\right) = \frac{2\,\mathsf{tan}\left(\frac{x}{2}\right)}{1 - \mathsf{tan}\left(\frac{x}{2}\right)^2} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \left[\mathsf{tan}\left(\frac{x}{2}\right), \mathsf{tan}\left(\frac{x}{2}\right)\right]$$

until $x \in [-1, 1]$. Beware that $\begin{pmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \notin \mathbb{T}^+$. Hence, the subtle presence of square brackets as defined in equation (10.2) instead of round brackets.

10.2.6 Inverse Tangent

A continued fraction for inverse tangent [73] is

$$\mathsf{arctan}\left(x\right) = \left[0.x, \left\langle 2n - 1.n^2x^2\right\rangle_{n\geq 1}\right]. \qquad (10.14)$$

The formula stated by Vuillemin [71] can be derived using equation (10.5)

$$\begin{pmatrix} 0 & x \\ 1 & 0 \end{pmatrix} \prod_{n=1}^{\infty} \begin{pmatrix} 2n-1 & n^2x^2 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \prod_{n=1}^{\infty} \begin{pmatrix} 2n-1 & n^2x \\ x & 0 \end{pmatrix}$$

and simplified even further using equation (10.7) to

$$\prod_{n=1}^{\infty}\begin{pmatrix} 0 & x \\ n^2 x & 2n-1 \end{pmatrix}.$$

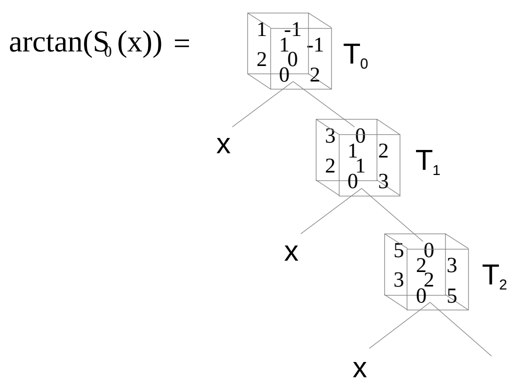
Therefore, the corresponding sequence of tensors according to part (i) is

$$T_n = \begin{pmatrix} 0 & 1 & 0 & 0 \\ (n+1)^2 & 0 & 0 & 2n+1 \end{pmatrix}.$$

Since, $T_n \in \mathbb{T}^+$ for all $n \geq 0$, this immediately gives an expression tree for $\mathsf{arctan}\,(x)$ valid for all $x \in [0, \infty]$. However, the efficiency of this expression tree decreases with increasing x. Note that for $M_n = \begin{pmatrix} 1 & -1 \\ n+1 & n+1 \end{pmatrix}$ and $N = S_0$, $M_{n-1}^{-1} \bullet T_n \bullet_1 N \bullet_2 M_n \in \mathbb{T}^+$ for all $n \geq 1$. Therefore, according to part (ii), we have

$$\begin{aligned} \mathsf{arctan}\,(S_0\,\{x\}) &= \begin{pmatrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{pmatrix} \{x, E_1\,(x)\} \qquad (10.15) \\ E_n\,(x) &= \begin{pmatrix} 2n+1 & n & 0 & n+1 \\ n+1 & 0 & n & 2n+1 \end{pmatrix} \{x, E_{n+1}\,(x)\}. \end{aligned}$$

Pictorially, we have



This equation for inverse tangent can be used to capture all the extended real numbers when used in conjunction with the following trigonometric identities

$$\begin{aligned} \mathsf{arctan}\,(S_+\,\{y\}) &= \mathsf{arctan}\,(S_0\,\{y\}) + \frac{\pi}{4} \\ \mathsf{arctan}\,(S_\infty\,\{y\}) &= \mathsf{arctan}\,(S_0\,\{y\}) + \frac{\pi}{2} \\ \mathsf{arctan}\,(S_-\,\{y\}) &= \mathsf{arctan}\,(S_0\,\{y\}) + \frac{3\pi}{4}. \end{aligned}$$

The identity

$$\mathsf{arctan}\,(a+b) = \mathsf{arctan}\,(a) + \mathsf{arctan}\left(\frac{b}{1+a\,(a+b)}\right)$$

can be used to derive an alternative algorithm for inverse tangent. Note that the inverse tangent of a rational number evaluates more quickly than the inverse tangent of an irrational number and the inverse tangent of a small number evaluates more quickly than the inverse tangent of a large number. So, consider a real number x and find a rational number $\frac{p}{q}$ close to x and let ϵ be the difference $x - \frac{p}{q}$. Therefore

$$\begin{aligned}\mathsf{arctan}\,(x) &= \mathsf{arctan}\left(\frac{p}{q}+\epsilon\right)\\ &\mathsf{arctan}\left(\frac{p}{q}\right) + \mathsf{arctan}\left(\left(\begin{array}{cc} q^2 & 0 \\ pq & p^2+q^2 \end{array}\right)[\epsilon]\right).\end{aligned}$$

10.2.7 Power Function

A Jacobi type continued fraction for the power function [70] is

$$(1+x)^y = \left[1.xy, \left\langle 1.\frac{x\,(n-y)}{2\,(2n-1)}, 1.\frac{x\,(n+y)}{2\,(2n+1)}\right\rangle_{n=0}^{\infty}\right].$$

This can be transformed to

$$(1+x)^y = \left(\begin{array}{cc} y & 1 \\ 0 & 1 \end{array}\right)\prod_{n=1}^{\infty}\left(\begin{array}{cc} 0 & x \\ x & 2n-1 \end{array}\right)\left(\begin{array}{cc} 0 & n-y \\ n+y & 2 \end{array}\right)$$

where $x \geq 0$ and $0 \leq y \leq 1$.

10.3 Miscellaneous functions

10.3.1 Real Modulus

A useful function is the real modulus function $\mathsf{rmod} : \mathbb{R}^\infty \times \mathbb{R}^\infty \to \mathbb{Z} \times \mathbb{R}^\infty$ given by

$$\mathsf{rmod}\,(x,y) = (n,z)$$

where

$$\begin{aligned} x &= ny+z \\ z &\in [-y,y]. \end{aligned}$$

This provides an alternative method to implement tangent; namely

$$\mathsf{tan}\,(x) = \begin{cases} S_+\,\{\mathsf{tan}\,(y)\} & \text{if } \mathsf{mod}\,(n,4) = 0 \\ S_\infty\,\{\mathsf{tan}\,(y)\} & \text{if } \mathsf{mod}\,(n,4) = 1 \\ S_-\,\{\mathsf{tan}\,(y)\} & \text{if } \mathsf{mod}\,(n,4) = 2 \\ S_0\,\{\mathsf{tan}\,(y)\} & \text{if } \mathsf{mod}\,(n,4) = 3 \end{cases}$$

where

$$(n, y) = \mathsf{rmod}\left(x, \frac{\pi}{4}\right).$$

Note that $y \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right] \subseteq \mathsf{info}\,(S_0)$. Therefore, equation (10.13) can be used to evaluate $\mathsf{tan}\,(y)$.

10.3.2 Complex Functions

The striking symmetry between the expression trees for logarithm and inverse tangent is tantalizing.

$$\mathsf{log} \qquad \begin{pmatrix} n & 2n+1 & n+1 & 0 \\ 0 & n+1 & 2n+1 & n \end{pmatrix}$$

$$\mathsf{arctan} \qquad \begin{pmatrix} 2n+1 & n & 0 & n+1 \\ n+1 & 0 & n & 2n+1 \end{pmatrix}$$

The following transformations provide a compelling link between these expression trees, but we do not provide any mathematical justification.

Suppose

$$\begin{aligned} f\,(S_0\,\{x\}) &= E_0\,(x) \\ E_n\,(x) &= T_n\,\{x, E_{n+1}\,(x)\} \\ T_n &= \begin{pmatrix} a_n & c_n & d_n & b_n \\ b_n & d_n & c_n & a_n \end{pmatrix} \end{aligned}$$

and suppose we want an expression tree for $f\,(ix)$ where $i = \sqrt{-1}$. Let

$$\begin{aligned} N &= \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix} \\ M &= \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \end{aligned}$$

and note that

$$iS_0 = S_0N$$

and

$$\begin{aligned} & M^{\dagger} \bullet \begin{pmatrix} a & c & d & b \\ b & d & c & a \end{pmatrix} \bullet_1 N \bullet_2 M \\ \equiv & \begin{pmatrix} a+d-b+c & b+c+a-d & a+d+b-c & b+c-a+d \\ b+c-a+d & a+d+b-c & b+c+a-d & a+d-b+c \end{pmatrix}. \end{aligned}$$

Therefore

$$\begin{aligned} f\left(iS_0\left\{x\right\}\right) &= M\left\{h_0\left(x\right)\right\} \\ h_n\left(x\right) &= U_n\left\{x, h_{n+1}\left(x\right)\right\} \\ U_n &= \left(\begin{matrix} a_n+d_n-b_n+c_n & b_n+c_n+a_n-d_n \\ b_n+c_n-a_n+d_n & a_n+d_n+b_n-c_n \end{matrix}\right. \\ & \qquad \left.\begin{matrix} a_n+d_n+b_n-c_n & b_n+c_n-a_n+d_n \\ b_n+c_n+a_n-d_n & a_n+d_n-b_n+c_n \end{matrix}\right). \end{aligned}$$

Example 45 *Note that*

$$\operatorname{arctanh}\left(S_0\left\{x\right\}\right) = -i\arctan\left(iS_0\left\{x\right\}\right)$$

and

$$\begin{aligned} \arctan\left(S_0\left\{x\right\}\right) &= \begin{pmatrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{pmatrix}\left\{x, E_1\left(x\right)\right\} \\ E_n\left(x\right) &= \begin{pmatrix} 2n+1 & n & 0 & n+1 \\ n+1 & 0 & n & 2n+1 \end{pmatrix}\left\{x, E_{n+1}\left(x\right)\right\}. \end{aligned}$$

But

$$\begin{aligned} & -i\begin{pmatrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{pmatrix} \bullet_1 N \bullet_2 M \\ \equiv & \begin{pmatrix} 1 & 1 & -1 & -1 \\ 0 & 2 & 2 & 0 \end{pmatrix} \\ & M^{\dagger} \bullet \begin{pmatrix} 2n+1 & n & 0 & n+1 \\ n+1 & 0 & n & 2n+1 \end{pmatrix} \bullet_1 N \bullet_2 M \\ \equiv & \begin{pmatrix} n & 2n+1 & n+1 & 0 \\ 0 & n+1 & 2n+1 & n \end{pmatrix}. \end{aligned}$$

So

$$\begin{aligned} \operatorname{arctanh}\left(S_0\left\{x\right\}\right) &= \begin{pmatrix} 1 & 1 & -1 & -1 \\ 0 & 2 & 2 & 0 \end{pmatrix}\left\{x, E_1\left(x\right)\right\} \\ E_n\left(x\right) &= \begin{pmatrix} n & 2n+1 & n+1 & 0 \\ 0 & n+1 & 2n+1 & n \end{pmatrix}\left\{x, E_{n+1}\left(x\right)\right\}. \end{aligned}$$

Example 46 *Note that*

$$\mathsf{tanh}\left(S_0\left\{x\right\}\right)=-i\mathsf{tan}\left(iS_0\left\{x\right\}\right)$$

and

$$\begin{aligned}\mathsf{tan}\left(S_0\left\{x\right\}\right) &= \begin{pmatrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{pmatrix}\left\{x,E_1\left(x\right)\right\} \\ E_n\left(x\right) &= \begin{pmatrix} 2n+1 & 2n-1 & 2n+1 & 2n+3 \\ 2n+3 & 2n+1 & 2n-1 & 2n+1 \end{pmatrix}\left\{x,E_{n+1}\left(x\right)\right\}.\end{aligned}$$

But

$$\begin{aligned}& -i\begin{pmatrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{pmatrix}\bullet_1 N\bullet_2 M \\ \equiv\ & \begin{pmatrix} 1 & 1 & -1 & -1 \\ 0 & 2 & 2 & 0 \end{pmatrix} \\ & M^\dagger\bullet\begin{pmatrix} 2n+1 & 2n-1 & 2n+1 & 2n+3 \\ 2n+3 & 2n+1 & 2n-1 & 2n+1 \end{pmatrix}\bullet_1 N\bullet_2 M \\ \equiv\ & \begin{pmatrix} 2n-1 & 2n+1 & 2n+3 & 2n+1 \\ 2n+1 & 2n+3 & 2n+1 & 2n-1 \end{pmatrix}.\end{aligned}$$

So

$$\begin{aligned}\mathsf{tanh}\left(S_0\left\{x\right\}\right) &= \begin{pmatrix} 1 & 1 & -1 & -1 \\ 0 & 2 & 2 & 0 \end{pmatrix}\left\{x,E_1\left(x\right)\right\} \\ E_n\left(x\right) &= \begin{pmatrix} 2n-1 & 2n+1 & 2n+3 & 2n+1 \\ 2n+1 & 2n+3 & 2n+1 & 2n-1 \end{pmatrix}\left\{x,E_{n+1}\left(x\right)\right\}.\end{aligned}$$

Example 47 *Note that*

$$\mathsf{exp}\left(S_0\left\{x\right\}\right)=E_0\left(x\right)$$

$$E_n\left(x\right)=\begin{pmatrix} 2n+2 & 2n+1 & 2n & 2n+1 \\ 2n+1 & 2n & 2n+1 & 2n+2 \end{pmatrix}\left\{x,E_{n+1}\left(x\right)\right\}.$$

Therefore

$$\begin{aligned}\mathsf{exp}\left(iS_0\left\{y\right\}\right) &= \begin{pmatrix} i & 1+2i & 2+i & 1 \\ 1 & 2+i & 1+2i & i \end{pmatrix}\left\{y,F_1\left(y\right)\right\} \\ F_n\left(y\right) &= \begin{pmatrix} 2n+1 & 2n+2 & 2n+1 & 2n \\ 2n & 2n+1 & 2n+2 & 2n+1 \end{pmatrix}\left\{y,F_{n+1}\left(y\right)\right\}.\end{aligned}$$

But

$$T_\times\bullet_2 M=\begin{pmatrix} 1 & -i & 0 & 0 \\ 0 & 0 & -i & 1 \end{pmatrix}.$$

Therefore

$$\mathsf{exp}\left(S_0\left\{x\right\}+iS_0\left\{y\right\}\right)=\begin{pmatrix}1 & -i & 0 & 0\\ 0 & 0 & -i & 1\end{pmatrix}\left\{E_0\left(x\right),F_0\left(y\right)\right\}$$

$$\begin{aligned}E_n\left(x\right) &= \begin{pmatrix}2n+2 & 2n+1 & 2n & 2n+1\\ 2n+1 & 2n & 2n+1 & 2n+2\end{pmatrix}\left\{x,E_{n+1}\left(x\right)\right\}\\ F_n\left(y\right) &= \begin{pmatrix}2n+1 & 2n+2 & 2n+1 & 2n\\ 2n & 2n+1 & 2n+2 & 2n+1\end{pmatrix}\left\{y,F_{n+1}\left(y\right)\right\}.\end{aligned}$$

Chapter 11

Normalization Algorithms

In chapter 13, we define theoretical languages and reduction rules for converting an expression tree into a general normal product. This is interesting at a theoretical level and helps verify the theoretical basis for many underlying ideas. However, at a practical level, we need to consider the process of converting an expression tree into an exact floating point number. We call these conversion processes normalization. The process of normalization can be seen as a tug of war between two subprocesses; namely *emission* and *absorption*.

Emission Extract information from the root node in order to construct an exact floating point number *if you can*.

Absorption Assimilate information from the depths of the expression tree into the root node *if you have to*.

11.1 Information Emission

In general, any non-singular matrix M can be *emitted* from an expression tree $E = L\{E_1, \ldots, E_N\}$ provided

$$\mathsf{info}(M) \supseteq \mathsf{info}(L)$$

or equivalently

$$M^\dagger \bullet L \in \mathbb{L}^+.$$

In which case, emission proceeds according to

$$E \to M\left\{M^\dagger[E]\right\}$$

or in other words

$$E \to M\left\{\left(M^\dagger \bullet L\right)\{E_1, \ldots, E_N\}\right\}$$

using the definition of matrix square bracket application as given in section 10.1.1.

For a signed expression tree E, only a sign matrix $S_{\{+,\infty,-,0\}}$ may be emitted leaving the unsigned expression tree $S_\sigma^\dagger[E]$

$$E \to S_\sigma \left\{S_\sigma^\dagger[E]\right\}. \tag{11.1}$$

For an unsigned expression tree E, only a digit matrix $D_{\mathbb{Z}(2)}$ may be emitted leaving the unsigned expression tree $D_d^\dagger[E]$

$$E \to D_d \left\{D_d^\dagger[E]\right\}. \tag{11.2}$$

The digit matrix D_0 should be avoided, if given a choice, because it involves slightly more computation.

11.2 Information Absorption

Absorption is the process of assimilation.

- Matrix assimilation $M \in \mathbb{M}$

$$M\left\{V^+\right\} \to \left(M \bullet V^+\right)$$

$$M\left\{N^+\{E\}\right\} \to \left(M \bullet N^+\right)\{E\}$$

$$M\left\{T^+\{E_1, E_2\}\right\} \to \left(M \bullet T^+\right)\{E_1, E_2\}$$

where $V^+ \in \mathbb{V}^+$, $M^+ \in \mathbb{M}^+$ and $T^+ \in \mathbb{T}^+$.

- Tensor assimilation $T \in \mathbb{T}$

$$T\left\{V^+, W^+\right\} \to \left(T \bullet_1 V^+ \bullet W^+\right)$$

$$T\left\{V^+, N^+\{E_2\}\right\} \to \left(T \bullet_2 V^+ \bullet N^+\right)\{E_2\}$$

$$T\left\{M^+\{E_1\}, W^+\right\} \to \left(T \bullet_1 M^+ \bullet_2 W^+\right)\{E_1\}$$

$$T\left\{M^+\{E_1\}, N^+\{E_2\}\right\} \to \left(T \bullet_1 M^+ \bullet_2 N^+\right)\{E_1, E_2\}$$

where $V^+, W^+ \in \mathbb{V}^+$ and $M^+, N^+ \in \mathbb{M}^+$.

11.3 Information Flow Analysis

One of the advantages of the exact floating point representation over the general normal product representation is that it gives a natural unit of information related to *metric* d_{P} defined in equation (3.5). This means that the complexity of algorithms can be more easily measured and controlled. In particular

$$d_{\mathrm{P}}(x,y) = |S_0(x) - S_0(y)|$$

for all $x, y \in [0,\infty]$. Note that

$$\begin{aligned} d_{\mathrm{P}}(D_d(x), D_d(y)) &= |(S_0 \bullet D_d)(x) - (S_0 \bullet D_d)(y)| \\ &= \left| \left(\begin{pmatrix} 1 & d \\ 0 & 2 \end{pmatrix} \bullet S_0 \right)(x) - \left(\begin{pmatrix} 1 & d \\ 0 & 2 \end{pmatrix} \bullet S_0 \right)(y) \right| \\ &= \frac{1}{2} d_{\mathrm{P}}(x,y). \end{aligned}$$

More generally, define the *distance function* $d : \mathbb{L}^+ \to [0,2]$ on unsigned linear fractional transformations by

$$d(L) = \mathsf{width}(\mathsf{info}(S_0 \bullet L))$$

because

$$d_{\mathrm{P}}(M(0), M(\infty)) = \mathsf{width}(\mathsf{info}(S_0 \bullet M))$$

for any $M \in \mathbb{M}^+$.

Example 48

$$\begin{aligned} M &= \begin{pmatrix} 5 & 5 \\ 10 & 11 \end{pmatrix} \\ S_0 \bullet M &= \begin{pmatrix} -5 & -6 \\ 15 & 16 \end{pmatrix} \\ \mathsf{info}(S_0 \bullet M) &= \left[-\frac{3}{8}, -\frac{1}{3} \right] \\ d(M) &= \frac{1}{24}. \end{aligned}$$

It can be seen that if $d(L) \leq 2^{-n}$ then at least n digit matrices can be emitted and if $d(L) > 2^{-n}$ then at most n digit matrices can be emitted. So, let

$$\begin{aligned} \mathsf{least}(L) &= \left\lfloor \log_2 \left(\frac{1}{d(L)} \right) \right\rfloor \\ \mathsf{most}(L) &= 1 + \mathsf{least}(L). \end{aligned}$$

In particular

$$\begin{aligned} d\left(\mathfrak{D}_c^n\right) &= 2^{1-n} \\ \mathsf{least}\left(\mathfrak{D}_c^n\right) &= n-1 \\ \mathsf{most}\left(\mathfrak{D}_c^n\right) &= n \end{aligned}$$

with $\mathfrak{D}_c^n$ as defined in equation (9.3). Of course, we know that exactly n digit matrices can be emitted from $\mathfrak{D}_c^n$ because it is defined as the product of n digit matrices.

Lemma 49 *If L is an unsigned linear fractional transformation then at least* $\mathsf{least}(L)$ *digit matrices can be emitted and at most* $\mathsf{most}(L)$ *digit matrices can be emitted.*

Note that

$$\mathsf{least}(V) = \mathsf{most}(V) = \infty$$

for $V \in \mathbb{V}^+$ and

$$\begin{aligned} \mathsf{most}(M) &= \left\lfloor \mathsf{log}_2\left(\frac{|a+b|\,|c+d|}{|\mathsf{det}(M)|}\right)\right\rfloor \\ \mathsf{least}(M) &= \mathsf{most}(M) - 1 \end{aligned}$$

for $M = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathbb{M}^+$. For example, the matrix

$$\begin{pmatrix} 5 & 5 \\ 10 & 11 \end{pmatrix} \equiv D_{-1} \bullet D_1 \bullet D_{-1} \bullet D_1 \bullet D_{-1} \bullet \begin{pmatrix} 10 & 0 \\ 5 & 16 \end{pmatrix}$$

can emit exactly 5 digit matrices and consistently

$$\begin{aligned} \mathsf{most}\begin{pmatrix} 5 & 5 \\ 10 & 11 \end{pmatrix} &= 5 \\ \mathsf{least}\begin{pmatrix} 5 & 5 \\ 10 & 11 \end{pmatrix} &= 4. \end{aligned}$$

Consider an arbitrary matrix $M \in \mathbb{M}^+$ applied to an unsigned general normal product. A reasonable question to ask is; can we decide in advance how many digit matrices we need to absorb into the matrix M in order that we can emit a specified number of digit matrices? Unfortunately, it turns out that we can only easily estimate how many digit matrices are required. The two estimates of particular interest to us are *underestimates* and *overestimates*. In particular, suppose that we want to emit $\epsilon \in \mathbb{N}$ digit matrices from a matrix $M \in \mathbb{M}^+$ applied to an unsigned general normal product and that $\delta \in \mathbb{N}$ is the estimated number of digit matrices required from the unsigned general normal product.

$$\begin{tikzcd} {} & M \arrow[l, "\epsilon"'] & {} \arrow[l, "\delta"'] \end{tikzcd}$$

Underestimate Find $\underline{\delta}(M, \epsilon)$ such that $d_{\mathrm{P}}(M(x), M(y)) > 2^{-\epsilon}$ whenever $d_{\mathrm{P}}(x, y) = 2^{1-\underline{\delta}(M,\epsilon)}$. In other words, if we absorb $\underline{\delta}$ digit matrices into M then at most ϵ digits can be emitted from M.

Overestimate Find $\overline{\delta}(M, \epsilon)$ such that $d_{\mathrm{P}}(M(x), M(y)) \leq 2^{-\epsilon}$ whenever $d_{\mathrm{P}}(x, y) = 2^{1-\overline{\delta}(M,\epsilon)}$. In other words, if we absorb $\overline{\delta}$ digit matrices into M then at least ϵ digits can be emitted from M.

Note that

$$\begin{aligned} d_{\mathrm{P}}(M(x), M(y)) &= |(S_0 \bullet M)(x) - (S_0 \bullet M)(y)| \\ &= \left|\left(S_0 \bullet M \bullet S_0^{\dagger}\right)(S_0(x)) - \left(S_0 \bullet M \bullet S_0^{\dagger}\right)(S_0(y))\right| \\ &= \left|\left(S_0 \bullet M \bullet S_0^{\dagger}\right)'(z)\right| d_{\mathrm{P}}(x, y) \text{ for some } z \in (-1, 1) \end{aligned}$$

by the mean value theorem provided that the function $\left(S_0 \bullet M \bullet S_0^{\dagger}\right)(x)$ is continuous over $[-1, 1]$ and differentiable over $(-1, 1)$. This condition is satisfied because $M \in \mathbb{M}^{+}$. Given that $M = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$, it can be shown that

$$\begin{aligned} \left(S_0 \bullet M \bullet S_0^{\dagger}\right)'(x) &= \frac{4\mathsf{det}(M)}{((a+b-c-d)x + (a+b+c+d))^2} \\ \left(S_0 \bullet M \bullet S_0^{\dagger}\right)'(-1) &= \frac{\mathsf{det}(M)}{(c+d)^2} \\ \left(S_0 \bullet M \bullet S_0^{\dagger}\right)'(1) &= \frac{\mathsf{det}(M)}{(a+b)^2}. \end{aligned}$$

Define the *expansivity* function $\Xi\colon \mathbb{M}^{+} \to \mathbb{Q}$ by

$$\begin{aligned} \Xi(M) &= \mathsf{sup}\{k \,|d_{\mathrm{P}}(M(x), M(y)) \geq k\, d_{\mathrm{P}}(x, y)\} \\ &= \mathsf{inf}\left(\left|\left(S_0 \bullet M \bullet S_0^{\dagger}\right)'([-1, 1])\right|\right) \\ &= \frac{|\mathsf{det}(M)|}{(\mathsf{max}(|a+b|, |c+d|))^2} \end{aligned} \qquad (11.3)$$

and the *contractivity* function $\overline{\Xi} : \mathbb{M}^{+} \to \mathbb{Q}$ by

$$\begin{aligned}
\overline{\Xi}(M) &= \inf\{k \,|d_{\mathrm{P}}(M(x), M(y)) \leq k\, d_{\mathrm{P}}(x,y)\} \\
&= \sup\left(\left|\left(S_0 \bullet M \bullet S_0^{\dagger}\right)'([-1,1])\right|\right) \\
&= \frac{|\det(M)|}{(\min(|a+b|, |c+d|))^2}. \qquad (11.4)
\end{aligned}$$

Theorem 50 *Given a matrix* $M \in \mathbb{M}^{+}$ *and* $\epsilon > 0$ *then*

$$\underline{\delta}(M,\epsilon) = \epsilon + \lceil \log_2(\underline{\Xi}(M)) \rceil$$

and

$$\overline{\delta}(M,\epsilon) = \epsilon + \left\lceil \log_2\left(\overline{\Xi}(M)\right) \right\rceil + 1$$

are underestimates and overestimates respectively.

Proof : Assume

$$d_{\mathrm{P}}(x,y) = 2^{1-\underline{\delta}(M,\epsilon)}.$$

But

$$\begin{aligned}
d_{\mathrm{P}}(M(x), M(y)) &\geq \underline{\Xi}(M)\, 2^{1-\underline{\delta}(M,\epsilon)} \\
&= \underline{\Xi}(M) \,\mathrm{pow}(2, 1-\epsilon - \lceil \log_2(\underline{\Xi}(M)) \rceil) \\
&= \underline{\Xi}(M)\, 2^{-\epsilon} \,\mathrm{pow}(2, 1 - \lceil \log_2(\underline{\Xi}(M)) \rceil) \\
&> \underline{\Xi}(M)\, 2^{-\epsilon} \,\mathrm{pow}(2, -\log_2(\underline{\Xi}(M))) \\
&= 2^{-\epsilon}
\end{aligned}$$

because $1 - \lceil x \rceil > -x$. Therefore

$$d_{\mathrm{P}}(M(x), M(y)) > 2^{-\epsilon}.$$

Assume

$$d_{\mathrm{P}}(x,y) = 2^{1-\overline{\delta}(M,\epsilon)}.$$

But

$$\begin{aligned}
d_{\mathrm{P}}(M(x), M(y)) &\leq \overline{\Xi}(M)\, 2^{1-\overline{\delta}(M,\epsilon)} \\
&= \overline{\Xi}(M) \,\mathrm{pow}\left(2, -\epsilon - \left\lceil \log_2\left(\overline{\Xi}(M)\right) \right\rceil\right) \\
&= \overline{\Xi}(M)\, 2^{-\epsilon} \,\mathrm{pow}\left(2, -\left\lceil \log_2\left(\overline{\Xi}(M)\right) \right\rceil\right) \\
&\leq \overline{\Xi}(M)\, 2^{-\epsilon} \,\mathrm{pow}\left(2, -\log_2\left(\overline{\Xi}(M)\right)\right) \\
&= 2^{-\epsilon}
\end{aligned}$$

because $-\lceil x \rceil \leq -x$. Therefore

$$d_{\mathrm{P}}(M(x), M(y)) \leq 2^{-\epsilon}. \blacksquare$$

Example 51 *Consider*

$$M = \begin{pmatrix} 5 & 5 \\ 10 & 11 \end{pmatrix}.$$

We have

$$\begin{aligned} \underline{\delta}(M,\epsilon) &= \epsilon + \left\lceil \log_2 \left(\frac{5}{16^2} \right) \right\rceil = \epsilon - 5 \\ \overline{\delta}(M,\epsilon) &= \epsilon + \left\lceil \log_2 \left(\frac{5}{15^2} \right) \right\rceil + 1 = \epsilon - 4. \end{aligned}$$

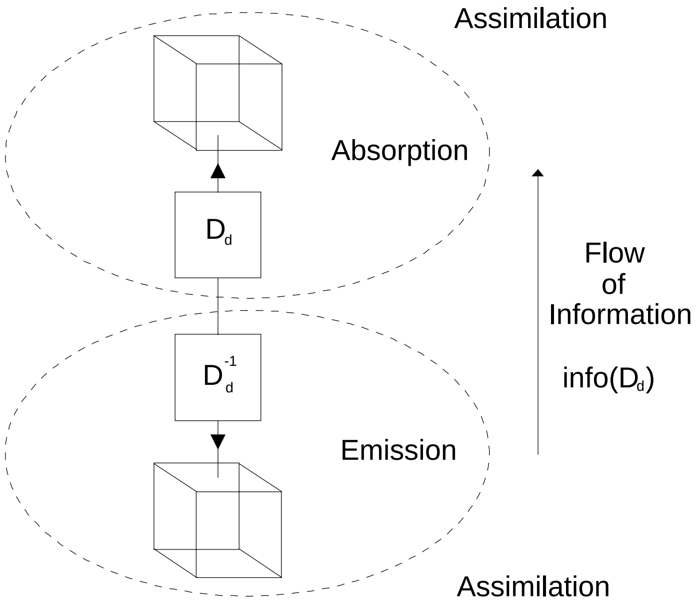
This means that if the matrix M is applied to an unsigned general normal product then

- *at most ϵ digit matrices can be emitted whenever $(\epsilon - 5)$ digit matrices are absorbed and*
- *at least ϵ digit matrices can be emitted whenever $(\epsilon - 4)$ digit matrices are absorbed.*

In summary, $\underline{\delta}(M,\epsilon)$ corresponds to a digit matrix absorption underestimate and $\overline{\delta}(M,\epsilon)$ corresponds to a digit matrix absorption overestimate whenever ϵ digit matrices are required from an unsigned matrix M applied to an unsigned general normal product. These formulae can and will be used for the optimization of evaluation.

11.4 Digit Exchange Policy

Tensors cannot be assimilated with tensors without introducing rank 4 tensors. The next best thing is an *information exchange* (i.e. a simultaneous emission and absorption).



Only the digit matrices $D_{\mathbb{Z}(2)}$ should be exchanged

$$\frac{E_2 \to D_d \left\{ D_d^\dagger \left[E_2 \right] \right\}}{T\left\{V^+, E_2\right\} \to \left(T \bullet_1 V^+ \bullet D_d\right) \left\{ D_d^\dagger \left[E_2 \right] \right\}}$$

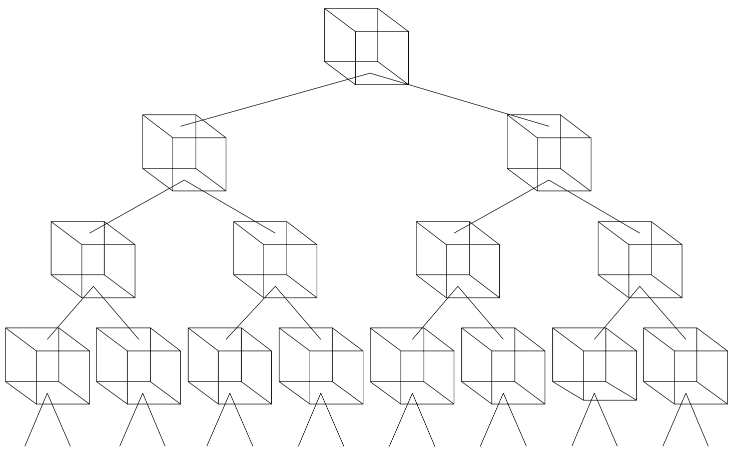
$$\frac{E_2 \to D_d \left\{ D_d^\dagger \left[E_2 \right] \right\}}{T\left\{M^+\left\{E_1\right\}, E_2\right\} \to \left(T \bullet_1 M^+ \bullet_2 D_d\right) \left\{ E_1, D_d^\dagger \left[E_2 \right] \right\}}$$

$$\frac{E_1 \to D_e \left\{ D_e^\dagger \left[E_1 \right] \right\}}{T\left\{E_1, W^+\right\} \to \left(T \bullet_1 D_e \bullet_2 W^+\right) \left\{ D_e^\dagger \left[E_1 \right] \right\}}$$

$$\frac{E_1 \to D_e \left\{ D_e^\dagger \left[E_1 \right] \right\}}{T\left\{E_1, N^+\left\{E_2\right\}\right\} \to \left(T \bullet_1 D_e \bullet N^+\right) \left\{ D_e^\dagger \left[E_1 \right], E_2 \right\}}$$

$$\frac{E_1 \to D_d \left\{ D_d^\dagger \left[E_1 \right] \right\} \qquad E_2 \to D_e \left\{ D_e^\dagger \left[E_2 \right] \right\}}{T\left\{E_1, E_2\right\} \to \left(T \bullet_1 D_d \bullet_2 D_e\right) \left\{ D_d^\dagger \left[E_1 \right], D_e^\dagger \left[E_2 \right] \right\}}.$$

This policy ensures a linear integer bit size growth for the coefficients in the two tensors. This conjecture has been proved by Reinhold Heckmann, which he calls the *law of big numbers* [32]. However, this policy introduces potential deadlock situations. For instance, consider an expression tree consisting of an infinite binary tree of tensors



and suppose that all the tensors contain insufficient information to emit a single digit matrix. Clearly, no digit exchange can occur and a deadlock results. In this case, a deadlock is inevitable with a digit exchange policy. However, in practice, we can restrict ourselves to expression trees that contain at most one infinite branch of tensors and without loss of generality we can insist that this branch is the right most branch as illustrated in equation (10.4). This still leaves other possible deadlock situations. Suppose that we have an expression tree as illustrated in equation (10.4) such that all the tensors contain insufficient information to emit a digit matrix and the argument x is given by an infinite unsigned general normal product. The root tensor T_0 cannot emit a digit and so, in the interest of fairness, it must absorb from the left and the right argument. The left argument is an infinite unsigned general normal product and so a matrix is absorbed. The right argument is an expression tree with root tensor T_1 and so a digit exchange is required. But the tensor T_1 cannot emit a digit and so we have a deadlock. This kind of deadlock can be avoided sometimes if we insist that a number of identity matrices are placed between every pair of connected tensors. These identity matrices act as a kind of delaying tactic. However, this still assumes that a digit emission is possible with sufficient absorption from the general normal product. Unfortunately, this may not be the case. For instance, consider the function $f(x)$ in equation (10.4) with

$$T_n = \begin{pmatrix} a_n & c_n & e_n & g_n \\ b_n & d_n & f_n & h_n \end{pmatrix} \in \mathbb{T}^+$$

for all $n \in \mathbb{N}$. Observe that

$$\begin{pmatrix} a_n & c_n & e_n & g_n \\ b_n & d_n & f_n & h_n \end{pmatrix} (x, y) = \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix} (y)$$

and so

$$f(x) = \prod_{n=0}^{\infty} \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix}.$$

This insight shows that the function $f(x)$ will deadlock if

$$\forall n \in \mathbb{N} \cdot \forall x \in [0, \infty] \cdot d \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix} > 1 \tag{11.5}$$

and x is given by an infinite unsigned general normal product. This is because whatever left absorption occurs it will never be enough to allow a digit matrix emission. Actually, we may allow

$$\forall n \in \mathbb{N} \cdot d \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix} = 1 \tag{11.6}$$

at isolated points. This is because whatever left absorption occurs it will never be as good as a singleton.

Example 52 *Consider*

$$f(x) = \mu y \cdot \begin{pmatrix} 0 & 1 & 0 & 2 \\ 2 & 1 & 3 & 1 \end{pmatrix} \{x, y\}$$

by Reinhold Heckmann. So

$$f(x) = \prod_{n=0}^{\infty} \begin{pmatrix} 0 & x+2 \\ 2x+3 & x+1 \end{pmatrix}.$$

But

$$d \begin{pmatrix} 0 & x+2 \\ 2x+3 & x+1 \end{pmatrix} = \frac{2x+4}{2x+3}.$$

Notice that $\frac{2x+4}{2x+3} > 1$ *for* $x \in [0, \infty)$ *and* $\frac{2x+4}{2x+3} = 1$ *at the isolated point* $x = \infty$*. Therefore the function* $f(x)$ *will deadlock whenever* x *is given by an infinite unsigned general normal product.*

Conversely, suppose that for each unsigned tensor T_n, a right absorption of $\delta \in \mathbb{N}$ digit matrices implies that $\delta + 1 \in \mathbb{N}$ digit matrices can be emitted if a sufficient number $\gamma(\delta) \in \mathbb{N}$ of digit matrices are absorbed from the left. This scenario can be compared to the notion an overestimate $\overline{\delta}(M, \epsilon)$ for a matrix $M \in \mathbb{M}^+$ applied to the unsigned general normal product $y \in \mathbb{E}^+$

$$\begin{aligned} M_n &= \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix} \\ \epsilon &\geq \delta + 1 \\ \overline{\delta}(M, \epsilon) &= \delta. \end{aligned}$$

Therefore, from theorem 50, the condition

$$\forall n \in \mathbb{N} \cdot \forall x \in [0, \infty] \cdot \Xi \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix} \leq \frac{1}{4}$$

must be satisfied. Actually, this condition must be strengthened slightly to

$$\forall n \in \mathbb{N} \cdot \forall x \in [0, \infty] \cdot \Xi \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix} < \frac{1}{4} \tag{11.7}$$

with

$$\forall n \in \mathbb{N} \cdot \Xi \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix} = \frac{1}{4} \tag{11.8}$$

at isolated points because whatever left absorption occurs it will never be as good as a singleton. Unfortunately, uncertainty remains for expression trees that do not satisfy the *deadlock conditions* (11.5) and (11.6) or the *deadlock-free conditions* (11.7) and (11.8).

Note that

$$\Xi \begin{pmatrix} ax + e & cx + g \\ bx + f & dx + h \end{pmatrix} = \frac{|(ax+e)(dx+h) - (bx+f)(cx+g)|}{(\mathsf{min}\,(ax+e+bx+f, cx+g+dx+h))^2}.$$

Therefore

$$\forall x \in [0, \infty] \cdot \Xi \begin{pmatrix} ax + e & cx + g \\ bx + f & dx + h \end{pmatrix} < \frac{1}{4}$$

if and only if

$$\forall x \in [0, \infty] \cdot \frac{|(ax+e)(dx+h) - (bx+f)(cx+g)|}{(ax+e+bx+f)^2} < \frac{1}{4}$$

and

$$\forall x \in [0, \infty] \cdot \frac{|(ax+e)(dx+h) - (bx+f)(cx+g)|}{(cx+g+dx+h)^2} < \frac{1}{4}.$$

Example 53 *Consider the formula for tangent in equation (10.13).*

$$\begin{aligned} &\Xi \begin{pmatrix} (2n+1)x + (2n+1) & (2n-1)x + (2n+3) \\ (2n+3)x + (2n-1) & (2n+1)x + (2n+1) \end{pmatrix} \\ &= \left(\frac{x-1}{2xn + 2n + 2\mathsf{min}\,(1, x)} \right)^2 \\ &\leq \frac{1}{4n^2}. \end{aligned}$$

Clearly $\Xi(M) < \frac{1}{4}$ *for all* $n \geq 1$ *except at* $x \in \{0, \infty\}$ *for* $n = 1$ *where it is equal to* $\frac{1}{4}$. *This means that the formula for tangent in equation (10.13) is deadlock-free.*

Example 54 *Consider, the formula for inverse tangent in equation (10.15). Note that*

$$\Xi\begin{pmatrix} 2nx+x & nx+n+1 \\ nx+x+n & 2n+1 \end{pmatrix} = \frac{n\left(x-1\right)^2\left(n+1\right)}{\left(\mathsf{min}\left(3nx+2x+n, 3n+2+nx\right)\right)^2}$$

does not satisfy equations (11.7) and (11.8). However, let us consider a left absorption of D_0, which corresponds to the variable substitution $x = D_0\left\{y\right\}$.

$$\begin{pmatrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{pmatrix} \bullet_1 D_0 = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 3 & 1 & 1 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 2n+1 & n & 0 & n+1 \\ n+1 & 0 & n & 2n+1 \end{pmatrix} \bullet_1 D_0 = \begin{pmatrix} 6n+3 & 4n+1 & 2n+1 & 4n+3 \\ 4n+3 & 2n+1 & 4n+1 & 6n+3 \end{pmatrix}.$$

In other words, let us consider the following formula for inverse tangent

$$\mathsf{arctan}\left(S_0\left\{D_0\left\{y\right\}\right\}\right) = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 3 & 1 & 1 & 3 \end{pmatrix}\left\{y, E_1\left(y\right)\right\}$$

$$E_n\left(y\right) = \begin{pmatrix} 6n+3 & 4n+1 & 2n+1 & 4n+3 \\ 4n+3 & 2n+1 & 4n+1 & 6n+3 \end{pmatrix}\left\{y, E_{n+1}\left(y\right)\right\}.$$

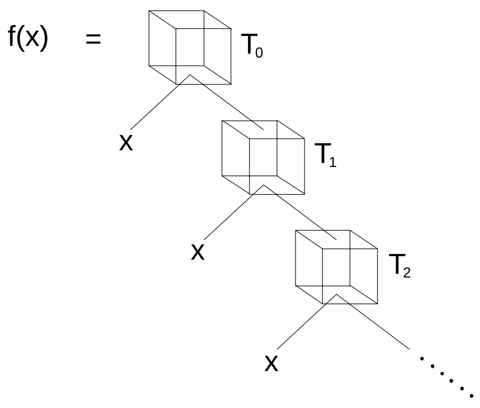
Observer that

$$\begin{aligned} &\Xi\begin{pmatrix} \left(6n+3\right)y+\left(2n+1\right) & \left(4n+1\right)y+\left(4n+3\right) \\ \left(4n+3\right)y+\left(4n+1\right) & \left(2n+1\right)y+\left(6n+3\right) \end{pmatrix} \\ &= \frac{n\left(y-1\right)^2\left(n+1\right)}{\left(\mathsf{min}\left(5yn+3y+3n+1, 3yn+y+5n+3\right)\right)^2} \\ &\leq \frac{n\left(n+1\right)}{\left(3n+1\right)^2} \\ &< \frac{1}{4}. \end{aligned}$$

for all $n \in \mathbb{N} - \left\{0\right\}$ and for all $y \in [0, \infty]$. Therefore, the revised formula for inverse tangent is deadlock-free. We do not know if the more elegant formula in equation (10.15) is deadlock-free or not.

In general, a formula based on a Taylor series can always be made deadlock-free if the domain of the function is sufficiently reduced. This is not a problem because various identities can be used to make up for the shortfall. In any case, a smaller domain means a more rapid convergence, leading to greater efficiency. The following proposition summarizes the above ideas.

Proposition 55 *The expression tree*



with $T_0 \in \mathbb{T}$ *and* $T_n = \begin{pmatrix} a_n & c_n & e_n & g_n \\ b_n & d_n & f_n & h_n \end{pmatrix} \in \mathbb{T}^+$ *for all* $n \geq 1$ *is deadlock-free if there exists* $m \in \mathbb{N}$ *such that*

$$\Xi \begin{pmatrix} a_n x + e_n & c_n x + g_n \\ b_n x + f_n & d_n x + h_n \end{pmatrix} < \frac{1}{4}$$

for all $x \in [0, \infty]$ *and for all* $n > m$, *where*

$$\Xi \begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} = \frac{\left| \mathsf{det} \begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} \right|}{\left(\mathsf{min}\left(|\alpha + \beta|, |\gamma + \delta|\right)\right)^2}.$$

11.5 Tensor Absorption Strategy

In the sections above, we assumed that absorption into a tensor occurred simultaneously from both arguments. In practice, it is usually best to absorb from one argument at a time and not necessarily alternately. The word "strategy" refers to the process of deciding whether to absorb from the left argument or to absorb from the right argument of a tensor. Consider a tensor with two unsigned general normal product arguments. Suppose that both arguments are infinite sequences of matrices. In this scenario, a strategy consists of an infinite sequence of states generated by a transition function

$$\mathsf{strategy} : \mathbb{T} \times \mathbb{N} \rightarrow \{1, 2\}$$

where 1 indicates "left absorption" and 2 indicates "right absorption". A sequence of states (T, n) is generated according to the following prescription.

- For $\mathsf{strategy}\,(T,n) = 1$ - Absorb the matrix M from the left argument $P = M\,\{R\}$.
$$T\,\{P,Q\} \to (T \bullet_1 M)\,\{R,Q\}$$
The next state is $(T \bullet_1 M, n+1)$.
- For $\mathsf{strategy}\,(T,n) = 2$ - Absorb the matrix N from the right argument $Q = N\,\{S\}$.
$$T\,\{P,Q\} \to (T \bullet_2 N)\,\{P,S\}$$
The next state is $(T \bullet_2 N, n+1)$.

However, if one or other of the arguments is finite (i.e. terminated by a vector) then a point may be reached at which the tensor mutates into a matrix and so the notion of strategy is no longer relevant.

- For $\mathsf{strategy}\,(T,n) = 1$ - Absorb the vector V from the left argument $P = V$ and mutate into a matrix
$$T\,\{P,Q\} \to (T \bullet_1 V)\,\{Q\}\,.$$
- For $\mathsf{strategy}\,(T,n) = 2$ - Absorb the vector W from the right argument $Q = W$ and mutate into a matrix
$$T\,\{P,Q\} \to (T \bullet_2 W)\,\{P\}\,.$$

In the next three sections, we explore three different strategies.

11.5.1 A Fair Strategy

Definition 56 *A **fair strategy** is any strategy such that endless left absorption implies that the meaning of the expression tree is independent of the right argument and endless right absorption implies that the meaning of the expression tree is independent of the left argument.*

A simple example of a fair strategy is one in which we alternately choose "left absorption" and "right absorption".

$$\mathsf{strategy}_{\mathrm{f}}\,(T,n) = \mathsf{mod}\,(n,2) + 1 \tag{11.9}$$

11.5.2 The Information Overlap Strategy

Left absorption into a tensor T increases the information in $\left(T^{\mathsf{T}}\right)_0$ and $\left(T^{\mathsf{T}}\right)_1$ because

$$\begin{aligned} T \bullet_1 V &= \left(\left(T^{\mathsf{T}}\right)_0 \bullet V, \left(T^{\mathsf{T}}\right)_1 \bullet V\right) \\ T \bullet_1 M &= \left(\left(T^{\mathsf{T}}\right)_0 \bullet M, \left(T^{\mathsf{T}}\right)_1 \bullet M\right)^{\mathsf{T}}, \end{aligned}$$

while right absorption into a tensor T increases the information in T_0 and T_1 because

$$\begin{aligned} T \bullet_2 V &= \left(T_0 \bullet V, T_1 \bullet V\right) \\ T \bullet_2 M &= \left(T_0 \bullet M, T_1 \bullet M\right) \end{aligned}$$

as demonstrated in lemma 30. So, consider the strategy in which we choose any fair strategy if the information in $\left(T^{\mathsf{T}}\right)_0$ is bottom or the information in $\left(T^{\mathsf{T}}\right)_1$ is bottom, we choose "right absorption" if the information in $\left(T^{\mathsf{T}}\right)_0$ and the information in $\left(T^{\mathsf{T}}\right)_1$ are disjoint and "left absorption" otherwise.

$$\mathsf{strategy}_{\mathrm{o}}(T,n) = \begin{cases} \mathsf{strategy}_{\mathrm{f}}(T,n) & \text{if } T \notin \mathbb{T}^+ \\ 1 & \text{if } \mathsf{info}\left(\left(T^{\mathsf{T}}\right)_0\right) \cap \\ & \mathsf{info}\left(\left(T^{\mathsf{T}}\right)_1\right) \neq \emptyset \\ 2 & \text{otherwise.} \end{cases} \tag{11.10}$$

We call this the *information overlap strategy.*

Proposition 57 *The information overlap strategy is a fair strategy.*

Proof : Consider a tensor $T \in \mathbb{T}$ with a left argument $E_0 \in \mathbb{E}^+$ and a right argument $F_0 \in \mathbb{E}^+$ defined by

$$\begin{aligned} E_i &= M_i\left\{E_{i+1}\right\} \\ F_j &= N_j\left\{F_{j+1}\right\}. \end{aligned}$$

We only need to consider $T \in \mathbb{T}^+$. This means that $T(x,y) \in [0,\infty]$ for all $x, y \in [0,\infty]$.

- Assume endless left absorption. Note that

$$T \bullet_1 \prod_{k=0}^{n} M_k = \left(\left(T^{\mathsf{T}}\right)_0 \bullet \prod_{k=0}^{n} M_k, \left(T^{\mathsf{T}}\right)_1 \bullet \prod_{k=0}^{n} M_k\right)^{\mathsf{T}}$$

for all $n \in \mathbb{N}$. Therefore

$$\bigcap_{i\in\{0,1\}} \mathsf{info}\left(\left(\left(T \bullet_1 \prod_{k=0}^{n} M_k\right)^{\mathsf{T}}\right)_i\right) \neq \emptyset$$

$$\bigcap_{i\in\{0,1\}} \mathsf{info}\left(\left(T^{\mathsf{T}}\right)_i \bullet \prod_{k=0}^{n} M_k\right) \neq \emptyset$$

for all $n \in \mathbb{N}$. Therefore

$$\bigcap_{i\in\{0,1\}} \mathsf{info}\left(\left(T^{\mathsf{T}}\right)_i \bullet V\right) \neq \emptyset$$

where

$$V = \bigcap_{n=0}^{\infty} \mathsf{info}\left(\prod_{k=0}^{n} M_k\right)$$

and V has real coefficients. But, this means that

$$\mathsf{info}\left(\left(T^{\mathsf{T}}\right)_0 \bullet V\right) = \mathsf{info}\left(\left(T^{\mathsf{T}}\right)_1 \bullet V\right).$$

Therefore

$$\mathsf{det}\left(\left(T^{\mathsf{T}}\right)_0 \bullet V, \left(T^{\mathsf{T}}\right)_1 \bullet V\right) = \mathsf{det}\left(T \bullet_1 V\right) = 0.$$

In other words, the expression tree $T\{E_0, F_0\}$ is independent of the right argument F_0.

- Assume endless right absorption. Note that

$$T \bullet_2 \prod_{k=0}^{n} N_k = \left(T_0 \bullet \prod_{k=0}^{n} N_k, T_1 \bullet \prod_{k=0}^{n} N_k\right)$$

for all $n \in \mathbb{N}$. Therefore

$$\bigcap_{i\in\{0,1\}} \mathsf{info}\left(\left(\left(T \bullet_2 \prod_{k=0}^{n} N_k\right)^{\mathsf{T}}\right)_i\right) = \emptyset$$

$$\bigcap_{i\in\{0,1\}} \mathsf{info}\left(\left(\left(T_0 \bullet \prod_{k=0}^{n} N_k, T_1 \bullet \prod_{k=0}^{n} N_k\right)^{\mathsf{T}}\right)_i\right) = \emptyset$$

for all $n \in \mathbb{N}$. Therefore

$$\bigcap_{i\in\{0,1\}} \mathsf{info}\left(\left(T_0 \bullet \prod_{k=0}^{n} N_k, T_1 \bullet \prod_{k=0}^{n} N_k\right)_i\right) \neq \emptyset$$

$$\bigcap_{i\in\{0,1\}} \mathsf{info}\left(T_i \bullet \prod_{k=0}^{n} N_k\right) \neq \emptyset$$

for all $n \in \mathbb{N}$. Therefore

$$\bigcap_{i \in \{0,1\}} \mathsf{info}(T_i \bullet V) \neq \emptyset$$

where

$$V = \bigcap_{n=0}^{\infty} \mathsf{info}\left(\prod_{k=0}^{n} N_k\right).$$

But, this means that

$$\mathsf{info}(T_0 \bullet V) = \mathsf{info}(T_1 \bullet V).$$

Therefore

$$\mathsf{det}(T_0 \bullet V, T_1 \bullet V) = \mathsf{det}(T \bullet_2 V) = 0.$$

In other words, the expression tree $T\{E_0, F_0\}$ is independent of the left argument E_0. ■

In practice, the information overlap strategy has proved the most efficient and effective.

11.5.3 The Outcome Minimization Strategy

The principle behind the outcome minimization strategy is to minimize the possible range of outcomes. The range of outcomes for a left absorption is given by

$$\mathsf{left}(T) = \mathsf{sup}\left\{ d\left(\left(T^{\mathsf{T}}\right)_0(x), \left(T^{\mathsf{T}}\right)_1(x)\right) \middle| x \in [0, \infty]\right\}$$

and the range of outcomes for a right absorption is given by

$$\mathsf{right}(T) = \mathsf{sup}\{d(T_0(y), T_1(y)) | y \in [0, \infty]\}$$

where $d(x, y)$ is a suitable metric. For instance, suppose $d(x, y) = d_{\mathrm{P}}(x, y)$ as defined in equation (3.5). Sadly, the exact formula for $\mathsf{left}(T)$ and $\mathsf{right}(T)$ involves square roots. However, a good approximation, in practice, for $T \in \mathbb{T}^+$, which is exact in many cases, is given by

$$\begin{aligned} \mathsf{left}_{\approx}(T) &= \omega\left(S_0 \bullet T^{\mathsf{T}}\right) \\ \mathsf{right}_{\approx}(T) &= \omega(S_0 \bullet T) \end{aligned}$$

where

$$\omega(T) = \mathsf{max}(|T_0(0) - T_1(0)|, |T_0(\infty) - T_1(\infty)|).$$

So, the outcome minimization strategy is given by

$$\mathsf{strategy}_{\mathrm{m}}(T, n) = \begin{cases} \mathsf{strategy}_{\mathrm{f}}(T, n) & \text{if } T \notin \mathbb{T}^+ \\ 1 & \text{if } \mathsf{left}_{\approx}(T) \leq \mathsf{right}_{\approx}(T) \\ 2 & \text{otherwise.} \end{cases}$$

We conjecture that the outcome minimization strategy is a fair strategy.

11.6 Straightforward Reduction Rules

In this section, we bring together the ideas of emission, absorption, exchange and strategy.

Firstly, let us define a *decision function* Δ_d that extends the strategy function to all linear fractional transformations. Let

$$\begin{aligned} \Delta_d &: \mathbb{L} \times \mathbb{N} \to \mathsf{boolean} \\ \Delta_1(M,n) &= \mathsf{true} \\ \Delta_1(T,n) &= (\mathsf{strategy}(T,n) = 1) \\ \Delta_2(T,n) &= (\mathsf{strategy}(T,n) = 2). \end{aligned} \tag{11.11}$$

The parameter d refers to the direction of absorption. This makes sense even for matrices because in fact $\bullet = \bullet_1$. The parameter n is best implemented as a hidden counter associated one per tensor and initialized to 0 at creation. In this way, we will drop the parameter n when it is convenient to do so.

Secondly, let us define a *sign emission function* sem that partially converts a signed expression tree E into the signed exact floating point representation. Let

$$\begin{aligned} \mathsf{sem} &: \mathbb{E} \times \mathbb{N} \to \mathbb{E} \\ \mathsf{sem}(E,i) &= \begin{cases} S_\sigma\left\{\mathsf{dem}\left(\mathfrak{D}_0^0, S_\sigma^\dagger[E], i\right)\right\} & \text{if } S_\sigma^\dagger \bullet L \in \mathbb{L}^+ \\ & \text{for some } \sigma \in \{+,\infty,-,0\} \\ \mathsf{sem}(L[F_1,\ldots,F_N], i) & \text{otherwise} \end{cases} \end{aligned} \tag{11.12}$$

where

$$L\{E_1,\ldots,E_N\} = E$$

and

$$F_d = \mathsf{ab}(L, E_d, \Delta_d(L)).$$

Note that

$$\mathfrak{D}_0^0 \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

In particular, $\mathsf{sem}(E,i)$ returns a signed expression tree of the form $S_\sigma\{\mathfrak{D}_c^i\{E'\}\}$ where S_σ is the required sign, $\mathfrak{D}_c^i$ is the i required digits compressed according to equation (9.4) and E' is an unsigned expression tree for the remaining digits (i.e. a continuation).

Thirdly, let us define a *digit emission function* dem that partially converts the unsigned expression tree $\mathfrak{D}_c^i\{E\}$ into the unsigned exact floating point representation.

$$\mathsf{dem} : \mathbb{M}^+ \times \mathbb{E}^+ \times \mathbb{N} \to \mathbb{E}^+ \tag{11.13}$$

$$\mathsf{dem}\left(\mathfrak{D}^i_c, E, j\right) = \begin{cases} \mathfrak{D}^i_c\{E\} & \text{if } j = 0 \text{ or } L \in \mathbb{V} \\ \mathsf{dem}\left(\mathfrak{D}^{i+1}_{2c+d}, D^\dagger_d[E], j-1\right) & \text{if } D^\dagger_d \bullet L \in \mathbb{L}^+ \text{ for some } d \in \mathbb{Z}(2) \\ \mathsf{dem}\left(\mathfrak{D}^i_c, L[F_1, \ldots, F_N], j\right) & \text{otherwise} \end{cases}$$

where

$$L\{E_1, \ldots, E_N\} = E$$

and

$$F_d = \mathsf{ab}\left(L, E_d, \Delta_d(L)\right).$$

In particular, $\mathsf{dem}\left(\mathfrak{D}^i_c, E, j\right)$ returns an unsigned expression tree of the form $\mathfrak{D}^{i+j}_{c'}\{E'\}$ where $\mathfrak{D}^{i+j}_{c'}$ is the $(i+j)$ required digits compressed according to equation (9.4) and E' is an unsigned expression tree for the remaining digits (i.e. a continuation).

Finally, let us define an *absorption function* ab that converts an unsigned expression tree E into another unsigned expression tree with a root node ready for absorption (by square bracket application as defined in equations (10.1) and (10.2)) into its parent node K.

$$\begin{aligned} \mathsf{ab} &: \mathbb{L} \times \mathbb{E}^+ \times \mathsf{boolean} \to \mathbb{E}^+ \\ \mathsf{ab}(K, E, b) &= \begin{cases} \mathfrak{D}^0_0\{E\} & \text{if } b = \mathsf{false} \\ \mathsf{dem}\left(\mathfrak{D}^0_0, E, 1\right) & \text{if } K \in \mathbb{T} \text{ and } L \in \mathbb{T} \\ E & \text{otherwise} \end{cases} \end{aligned} \tag{11.14}$$

where

$$L\{E_1, \ldots, E_N\} = E.$$

The boolean b is used to specify whether any information is actually required. The middle line in the definition corresponds to an information exchange between tensors.

11.7 Matrix Lazy Flow Analysis

For a matrix $M \in \mathbb{M}^+$ and natural number ϵ, we need to find the maximum units of information $\delta(M, \epsilon)$ that can be absorbed into M such that it contains at most ϵ units of information. In other words, we need the underestimate $\underline{\delta}(M, \epsilon)$ given in theorem 50. We need to convert this formula into an efficient algorithm. Let $\#(a)$ denote the number of bits required to represent the absolute value of the integer a.

Proposition 58 *For* $M = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathbb{M}^+$, *if*

$$\Delta_1^-(M,\epsilon) = \begin{cases} \epsilon - \#(\alpha^2) + \#(\mathsf{det}(M)) - 1 & \textit{if } \mathsf{det}(M) \textit{ is a power of } 2 \\ \epsilon - \#(\alpha^2) + \#(\mathsf{det}(M)) & \textit{otherwise.} \end{cases}$$

where

$$\alpha = \mathsf{max}(|a+b|, |c+d|)$$

then

$$\Delta_1^-(M,\epsilon) = \delta(M,\epsilon)$$

or

$$\Delta_1^-(M,\epsilon) = \delta(M,\epsilon) - 1.$$

Proof : We will use the easily verified identities

$$\begin{aligned} \lceil a \rceil &= -\lfloor -a \rfloor \\ \lfloor \mathsf{log}_2 a \rfloor &= \#(a) - 1 \\ \lceil \mathsf{log}_2 a \rceil &= \begin{cases} \#(a) - 1 & \text{if } a \text{ is a power of } 2 \\ \#(a) & \text{otherwise} \end{cases} \end{aligned}$$

and the fact that $\lfloor a+b \rfloor = \lfloor a \rfloor + \lfloor b \rfloor$ or $\lfloor a+b \rfloor = \lfloor a \rfloor + \lfloor b \rfloor + 1$. Consider $M = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathbb{M}^+$. So

$$\begin{aligned} \underline{\delta}(M,\epsilon) &= \epsilon + \lceil \mathsf{log}_2(\Xi(M)) \rceil \\ &= \epsilon + \left\lceil \mathsf{log}_2(|\mathsf{det}(M)|) - \mathsf{log}_2\left(\alpha^2\right) \right\rceil \\ &= \epsilon - \left\lfloor \mathsf{log}_2\left(\alpha^2\right) - \mathsf{log}_2(|\mathsf{det}(M)|) \right\rfloor. \end{aligned}$$

But

$$\begin{aligned} & \left\lfloor \mathsf{log}_2\left(\alpha^2\right) \right\rfloor + \lfloor -\mathsf{log}_2(|\mathsf{det}(M)|) \rfloor \\ = {} & \#\left(\alpha^2\right) - 1 - \lceil \mathsf{log}_2(|\mathsf{det}(M)|) \rceil \\ = {} & \begin{cases} \#(\alpha^2) - 1 - (\#(\mathsf{det}(M)) - 1) & \text{if } \mathsf{det}(M) \text{ is a power of } 2 \\ \#(\alpha^2) - 1 - \#(\mathsf{det}(M)) & \text{otherwise.} \end{cases} \\ = {} & \epsilon - 1 - \Delta_1^-(M,\epsilon) \end{aligned}$$

Therefore

$$\begin{aligned} \delta(M,\epsilon) &= \epsilon - \left(\epsilon - 1 - \Delta_1^-(M,\epsilon)\right) \\ &= \Delta_1^-(M,\epsilon) + 1 \end{aligned}$$

or

$$\begin{aligned}\delta(M,\epsilon) &= \epsilon - \left(\epsilon - 1 - \Delta_1^-(M,\epsilon) + 1\right) \\ &= \Delta_1^-(M,\epsilon). \blacksquare\end{aligned}$$

This proposition shows that $\Delta_1^-(M,\epsilon)$ is a conservative approximation of $\delta(M,\epsilon)$. In some cases, even though $\epsilon \geq 1$ and no digits can be emitted, the flow analysis given by Δ_1^- indicates that no information needs to be absorbed. So, we really need the function $\Delta_1^+(M,\epsilon) = \mathsf{max}\left(1, \Delta_1^-(M,\epsilon)\right)$ as well.

11.8 Tensor Lazy Flow Analysis

For a tensor $T \in \mathbb{T}^+$ and natural number ϵ, we need to find the maximum units of information $\delta_1(T,\epsilon)$ and $\delta_2(T,\epsilon)$ that can be left and right absorbed respectively into T such that it contains at most ϵ units of information. In other words, we need something similar to the underestimate $\underline{\delta}(M,\epsilon)$ given in theorem 50. By symmetry, it is clear that $\delta_1(T,\epsilon) = \delta_2\left(T^{\mathsf{T}},\epsilon\right)$. Note that for $U = S_0 \bullet T \bullet_1 S_0^\dagger \bullet_2 S_0^\dagger$

$$\begin{aligned}d_{\mathrm{P}}(T(w,x), T(w,y)) &= \left|(S_0 \bullet T)(w,x) - (S_0 \bullet T)(w,y)\right| \\ &= \left|U(S_0(w), S_0(x)) - U(S_0(w), S_0(y))\right| \\ &= \left|U_2(S_0(w), z)\right| \; d_{\mathrm{P}}(x,y) \;\text{ for some } z \in (-1,1)\end{aligned}$$

where $U_2(x,y)$ is the partial derivative of U with respect to y. It can be shown that

$$\mathsf{inf}\left(\left|U_2([-1,1],[-1,1])\right|\right) = \mathsf{min}\left(\Xi(T_0), \Xi(T_1)\right)$$

using the definition of the function $\Xi\colon \mathbb{M}^+ \to \mathbb{Q}$ in equation (11.3). This means that the algorithms

$$\begin{aligned}\Delta_1^-(T,\epsilon) &= \mathsf{min}\left(\Delta_1^-\left(\left(T^{\mathsf{T}}\right)_0,\epsilon\right), \Delta_1^-\left(\left(T^{\mathsf{T}}\right)_1,\epsilon\right)\right) \\ \Delta_2^-(T,\epsilon) &= \mathsf{min}\left(\Delta_1^-(T_0,\epsilon), \Delta_1^-(T_1,\epsilon)\right)\end{aligned}$$

provide conservative approximations for δ_1 and δ_2 respectively. Note that the overloading of Δ_1^- on matrices and tensors is intentional. In some cases, due to its conservative nature, even though $\epsilon \geq 1$ and no digits can be emitted, the flow analysis given by Δ_1^- and Δ_2^- indicate that no information needs to be absorbed from the left or from the right. So, in these cases, we need to decide whether to absorb just one from the left or just one from the right. So, let us define $\Delta_{\{1,2\}}^+(T,\epsilon)$ by

$$\Delta_n^+(T,\epsilon) = \text{if } \Delta_1^-(T,\epsilon) \leq 0 \text{ and } \Delta_2^-(T,\epsilon) \leq 0 \text{ then } \Delta_n(T) \text{ else } \Delta_n^-(T,\epsilon).$$

11.9 Efficient Linear Fractional Transformations

It should be noted that it is not always desirable to assimilate a vector or a matrix into its parent node. It depends on whether the vector or the matrix is spatially efficient relative to the information that it contains. The idea is that if a vector or a matrix is efficient then it should be assimilated directly otherwise only the digits $D_{\mathbb{Z}(2)}$ should be exchanged. Actually, it should also depend on the amount of information ϵ required as well. A particularly useful definition is

$$\begin{aligned}\mathsf{efficient} &: \mathbb{L}^+ \times \mathbb{N} \to \mathsf{boolean} \\ \mathsf{efficient}(L,\epsilon) &= \frac{\#(L)}{n} + \alpha \,\mathsf{min}(\epsilon, \mathsf{least}(L)) \leq \beta\end{aligned}$$

where α and β are adjustable parameters (e.g. $\alpha = 2$ and $\beta = 32$), $\#(L)$ denotes the total number of bits required to represent the coefficients in L and n is the number of coefficients in L.

11.10 Efficient Reduction Rules

In this section, we improve upon the straightforward reduction rules of section 11.6.

Firstly, let us improve the emission function dem with edem. The *efficient digit emission* function edem partially converts the unsigned expression tree $\mathfrak{D}^i_c\{E\}$ into the unsigned exact floating point representation. Let

$$\begin{aligned}\mathsf{edem} &: \mathbb{M} \times \mathbb{E}^+ \times \mathbb{N} \to \mathbb{E}^+ \\ \mathsf{edem}\left(\mathfrak{D}^i_c, E, j\right) &= \begin{cases} \mathfrak{D}^i_c\{E\} & \text{if } j = 0 \text{ or } L \in \mathbb{V} \\ \mathsf{edem}\left(\mathfrak{D}^{i+1}_{2c+d}, D^\dagger_d[E], j-1\right) & \text{if } D^\dagger_d \bullet L \in \mathbb{L}^+ \text{ for some } d \in \mathbb{Z}(2) \\ \mathsf{edem}\left(\mathfrak{D}^i_c, L\left[F^+_1, \ldots, F^+_N\right], j\right) & \text{otherwise}\end{cases}\end{aligned}$$

where

$$L\{E_1, \ldots, E_N\} = E$$

and

$$F^+_n = \mathsf{eab}\left(L, E_n, \Delta^+_n(L,j)\right).$$

In particular, $\mathsf{edem}\left(\mathfrak{D}^i_c, E, j\right)$ returns an unsigned expression tree of the form $\mathfrak{D}^{i+j}_{c'}\{E'\}$ where $\mathfrak{D}^{i+j}_{c'}$ is the $(i+j)$ required digits compressed according to equation (9.4) and E' is an unsigned expression tree for the remaining digits (i.e. a continuation).

Secondly, let us improve the absorption function ab with eab. The *efficient absorption* function converts the unsigned expression tree E into another unsigned expression tree with a root node ready for absorption (by square bracket application as defined in equations (10.1) and (10.2)) into its parent node K. Let

$$\begin{aligned}
\mathsf{eab} &: \mathbb{L} \times \mathbb{E}^{+} \times \mathbb{Z} \to \mathbb{E}^{+} \\
\mathsf{eab}\,(K, E, \epsilon) &= \begin{cases} \mathfrak{D}_0^0\,\{E\} & \text{if } \epsilon \leq 0 \\ L\left[F_1^-, \ldots, F_N^-\right] & \begin{array}{l}\text{if } \mathsf{efficient}\,(L,\epsilon) \text{ and} \\ (K \notin \mathbb{T} \text{ or } L \notin \mathbb{T})\end{array} \\ \mathsf{edem}\,(\mathfrak{D}_0^0, E, \epsilon) & \text{otherwise} \end{cases}
\end{aligned}$$

where

$$L\,\{E_1, \ldots, E_N\} = E$$

and

$$F_n^- = \mathsf{eab}\left(L, E_n, \Delta_n^-\,(L, \epsilon)\right).$$

The integer ϵ indicates the maximum required units of information in the root node. Note that Δ_n^+ is used when a specified number of digits $D_{\mathbb{Z}(2)}$ is required from an expression tree, whereas Δ_n^- is used when a conservative number of units of information is required from an expression tree.

11.11 Destructive Data Types

Any efficient implementation must avoid re-evaluating the same expression. This means that the language used must support references, pointers or destructive data types of some sort. For instance, there need only be one instantiation of the argument x in equation (10.4).

11.12 Scaling Invariance

Rescaling the coefficients of a linear fractional transformation down by their greatest common divisor is inefficient and generally unnecessary. In fact, absorption and emission of the radix 2 digit matrices means that only rescaling by 2 is necessary. The following theorem, a simpler variation of one by Reinhold Heckmann [32], effectively proves this conjecture.

Theorem 59 *If M is a matrix and V is a vector then every prime factor of $M \bullet V$ is a prime factor of* $\mathsf{det}\,(M)$ *or V.*

Proof : Let $W \in \mathbb{V}$ be any vector such that

$$M \bullet V = kW$$

for some prime number k. Note that

$$\begin{aligned} M^\dagger \bullet (M \bullet V) &= \left(M^\dagger \bullet M\right) \bullet V \\ &= \mathsf{det}\,(M)\, V \end{aligned}$$

and

$$M^\dagger \bullet (kW) = k\left(M^\dagger \bullet W\right).$$

Therefore k is a prime factor of $\mathsf{det}\,(M)$ or a prime factor of V. ■

Corollary 60 *If M is a matrix and L is a linear fractional transformation then every prime factor of $M \bullet L$ is a prime factor of $\mathsf{det}\,(M)$ or L.*

Proof : Use

$$\begin{aligned} M \bullet N &= (M \bullet N_0, M \bullet N_1) \\ M \bullet T &= (M \bullet T_0, M \bullet T_1) \ ■ \end{aligned}$$

Corollary 61 *If T is a tensor and M is a matrix then every prime factor of $T \bullet_n M$ is a prime factor of $\mathsf{det}\,(M)$ or T.*

Proof : Use

$$\begin{aligned} T \bullet_1 M &= \left(\left(T^\mathsf{T}\right)_0 \bullet M, \left(T^\mathsf{T}\right)_1 \bullet M\right)^\mathsf{T} \\ T \bullet_2 M &= (T_0 \bullet M, T_1 \bullet M) \ ■ \end{aligned}$$

This theorem means that if M is a matrix in its lowest terms then the only common factors of $M \bullet {}^bD_d$ and ${}^bD_d^\dagger \bullet M$ are the common factors of $\mathsf{det}\left({}^bD_d\right)$ and $\mathsf{det}\left({}^bD_d^\dagger\right)$. But

$$\mathsf{det}\left({}^bD_d\right) = \mathsf{det}\left({}^bD_d^\dagger\right) = 4b.$$

Therefore, only rescaling by 2 and b is necessary. In particular, for $b = 2$ only rescaling by 2 is necessary. Similarly for tensors.

Corollary 62 *If T is a tensor in its lowest terms and M is a matrices in its lowest terms then every common factor of $T \bullet_n M$ is a factor of $\mathsf{det}\,(M)$.*

Corollary 63 *If T is a tensor in its lowest terms and M is a matrices in its lowest terms then every common factor of $M \bullet T$ is a factor of $\mathsf{det}\,(M)$.*

Chapter 12

Implementation

All the algorithms in this thesis, from the lazy to the strict, have been tried and tested in a variety of languages including C/C++, Miranda, Haskell, Clean, Mathematica and CAML. Listed below is the most elegant out of all the implementations using the functional programming language called Miranda [36]. This implementation uses the straightforward reduction rules (see section 11.6) and the information overlap strategy (see equation (11.10)).

The strict implementations provide the best performance. However, the code is too long and complex to list in this thesis. Consequently, I will restrict myself to describing a fully lazy implementation of signed exact floating point in Miranda [36].

12.1 Type Definitions

12.1.1 Linear Fractional Transformations

```
vector  ==   (num,num)
matrix  ==   (vector,vector)
tensor  ==   (matrix,matrix)
lft     ::=  Vec vector | Mat matrix | Ten tensor num
```

Note that we maintain a hidden counter associated with each tensor in order to support a fair absorption strategy as discussed in section 11.6.

12.1.2 Expression Tree

```
expression  ::=  V vector |
                 M matrix expression |
                 T tensor num expression expression
```

12.1.3 Partial Exact Floating Point

Signed exact floating point is partially defined by `sefp` and unsigned exact floating point is partially defined by `uefp`. These types are partially defined in the sense that they are partially evaluated with an expression tree as a continuation. Sequences of digit matrices are stored compressed as two natural numbers as defined by `digits`. The compression technique used is described in section 9.1.

```
sefp    ::=  Spos uefp | Sinf uefp | Sneg uefp | Szer uefp
uefp    ==   (digits,expression)
digits  ==   (num,num)
```

12.2 Term Definitions

12.2.1 Basic Functions

Here are some basic functions that we need later.

- One function.

  ```
  one ::  * -> num  ->  *
  one x 1           =   x
  ```

- Identity matrix.

  ```
  identity  =  ((1,0),(0,1))
  ```

- Transpose of matrix or tensor.

  ```
  trans ::  ((*,*),(*,*))  ->  ((*,*),(*,*))
  trans ((a,b),(c,d))      =   ((a,c),(b,d))
  ```

- Determinant of matrix.

  ```
  determinant ::  matrix     ->  num
  determinant ((a,b),(c,d))  =   a*d-b*c
  ```

- Tame inverse of matrix (see equation (8.4)).

  ```
  inverse ::  matrix     ->  matrix
  inverse ((a,b),(c,d))  =   ((d,-b),(-c,a))
  ```

12.2.2 Binary Scaling Functions

Only rescaling of vectors, matrices and tensors by 2 is necessary as discussed in section 11.12.

```
vscale :: vector  ->  vector
vscale (a,b)      =   vscale (a div 2,b div 2),
                      a mod 2 = 0 & b mod 2 = 0
                  =   (a,b), otherwise

mscale :: matrix     ->  matrix
mscale ((a,b),(c,d))  =  mscale
                         ((a div 2,b div 2),
                         (c div 2,d div 2)),
                         a mod 2 = 0 & b mod 2 = 0 &
                         c mod 2 = 0 & d mod 2 = 0
                      =  ((a,b),(c,d)), otherwise

tscale :: tensor       ->  tensor
tscale (((a,b),(c,d)),
        ((e,f),(g,h)))  =  tscale
                           (((a div 2,b div 2),
                           (c div 2,d div 2)),
                           ((e div 2,f div 2),
                           (g div 2,h div 2))),
                           a mod 2 = 0 & b mod 2 = 0 &
                           c mod 2 = 0 & d mod 2 = 0 &
                           e mod 2 = 0 & f mod 2 = 0 &
                           g mod 2 = 0 & h mod 2 = 0
                        =  (((a,b),(c,d)),
                           ((e,f),(g,h))),
                           otherwise
```

12.2.3 Exact Floating Point

- Sign matrices for the exact floating point representation (see section 9.1).

```
spos  =  ((1,0),(0,1))
sinf  =  ((1,-1),(1,1))
sneg  =  ((0,1),(-1,0))
szer  =  ((1,1),(-1,1))
```

```
ispos  =  Mat (inverse spos)
isinf  =  Mat (inverse sinf)
isneg  =  Mat (inverse sneg)
iszer  =  Mat (inverse szer)
```

- Digit matrices for the exact floating point representation (see section 9.1).

```
dneg  =  ((1,1),(0,2))
dzer  =  ((3,1),(1,3))
dpos  =  ((2,0),(1,1))
```

```
idneg  =  Mat (inverse dneg)
idzer  =  Mat (inverse dzer)
idpos  =  Mat (inverse dpos)
```

12.2.4 Basic Arithmetic Operations

- Basic arithmetic tensors (see equations (8.7), (8.8), (8.9) and (8.10)).

```
tadd  =  (((0,0),(1,0)),((1,0),(0,1)))
tsub  =  (((0,0),(1,0)),((-1,0),(0,1)))
tmul  =  (((1,0),(0,0)),((0,0),(0,1)))
tdiv  =  (((0,0),(1,0)),((0,1),(0,0)))
```

- Reciprocal of signed exact floating point, unsigned exact floating point and expression tree (see section 10.1.2).

```
srec ::  sefp        ->  sefp
srec (Spos u)        =   Spos (urec u)
srec (Sneg u)        =   Sneg (urec u)
srec (Szer u)        =   Sinf (urec u)
srec (Sinf u)        =   Szer (urec u)
urec ::  uefp        ->  uefp
urec ((n,c),e)       =   ((n,-c),erec e)
erec ::  expression  ->  expression
erec e               =   M ((0,1),(1,0)) e
```

12.2.5 Linear Fractional Transformation Products

- Matrix products (see lemma 29).

```
mdotv :: matrix          ->  vector -> vector
mdotv ((a,b),(c,d)) (e,f) =  (a*e+c*f,b*e+d*f)
mdotm :: matrix          ->  matrix -> matrix
mdotm m (v,w)            =   (mdotv m v,mdotv m w)
mdott :: matrix          ->  tensor -> tensor
mdott m (n,o)            =   (mdotm m n,mdotm m o)
```

- Tensor products (see lemma 30).

```
tleftv :: tensor   ->  vector -> matrix
tleftv t v         =   trightv (trans t) v
tleftm :: tensor   ->  matrix -> tensor
tleftm t m         =   trans (trightm (trans t) m)
trightv :: tensor  ->  vector -> matrix
trightv (m,n) v    =   (mdotv m v,mdotv n v)
trightm :: tensor  ->  matrix -> tensor
trightm (m,n) o    =   (mdotm m o,mdotm n o)
```

- General dot product for linear fractional transformations. Note that for tensors, `dot 1` corresponds to a left product and `dot 2` corresponds to a right product.

```
dot                      ::  num -> lft -> lft -> lft
dot 1 (Mat m) (Vec v)    =   Vec (vscale (mdotv m v))
dot 1 (Mat m) (Mat n)    =   Mat (mscale (mdotm m n))
dot 1 (Mat m) (Ten t i)  =   Ten (tscale (mdott m t)) i
dot 1 (Ten t i) (Vec v)  =   Mat (mscale (tleftv t v))
dot 1 (Ten t i) (Mat m)  =   Ten t i, m = identity
                         =   Ten (tscale (tleftm t m))
                             (i+1), otherwise
dot 2 (Ten t i) (Vec v)  =   Mat (mscale (trightv t v))
dot 2 (Ten t i) (Mat m)  =   Ten t i, m = identity
                         =   Ten (tscale (trightm t m))
                             (i+1), otherwise
```

12.2.6 Type Cast Functions

- Convert a partial signed exact floating point into a matrix.

```
stom :: sefp   ->  matrix
stom (Spos u)  =   mdotm spos (dtom (fst u))
stom (Sinf u)  =   mdotm sinf (dtom (fst u))
stom (Sneg u)  =   mdotm sneg (dtom (fst u))
stom (Szer u)  =   mdotm szer (dtom (fst u))
```

- Convert a partial unsigned exact floating point into an expression tree.

```
utoe ::  uefp  ->  expression
utoe (d,e)     =   M (dtom d) e
```

- Convert compressed digit matrices into a single matrix (see equation (9.2)).

```
dtom ::  digits  ->  matrix
dtom (n,c)       =   mscale
                     ((2^n+c+1,2^n-c-1),
                     (2^n+c-1,2^n-c+1))
```

12.2.7 The Refinement Property

- The sign function enables a simple implementation of a refinement property predicate (see equation (8.1)).

```
sign ::  vector  ->  num
sign (a,b)       =   -1, a<0 & b<=0
                 =    0, a<0 & b>0
                 =   -1, a=0 & b<0
                 =    0, a=0 & b=0
                 =    1, a=0 & b>0
                 =    0, a>0 & b<0
                 =    1, a>0 & b>=0
```

- Vector, matrix and tensor refinement property predicates (see proposition 34).

$$\texttt{vrefine v} \equiv \texttt{v} \in \mathbb{V}^{+}$$
$$\texttt{mrefine m} \equiv \texttt{m} \in \mathbb{M}^{+}$$
$$\texttt{trefine t} \equiv \texttt{t} \in \mathbb{T}^{+}$$

```
vrefine ::  vector     ->  bool
vrefine v              =   sign v ~= 0
mrefine ::  matrix     ->  bool
mrefine (v,w)          =   a=b & b~=0
                            where  a  =  sign v
                                   b  =  sign w
trefine ::  tensor     ->  bool
trefine ((v,w),(x,y))  =   a=b & b=c &
                           c=d & d~=0
                            where  a  =  sign v
                                   b  =  sign w
                                   c  =  sign x
                                   d  =  sign y
```

- General refinement property predicate for linear fractional transformations.

$$\texttt{refine l} \equiv \texttt{l} \in \mathbb{L}^{+}$$

```
refine ::  lft    ->  bool
refine (Vec v)    =   vrefine v
refine (Mat m)    =   mrefine m
refine (Ten t i)  =   trefine t
```

12.2.8 Basic Expression Tree Functions

- Number of branches emanating from a linear fractional transformation if it were in an expression tree.

```
branch ::  lft    ->  num
branch (Vec v)    =   0
branch (Mat m)    =   1
branch (Ten t i)  =   2
```

- Type predicate for vectors, matrices and tensors.

```
vis ::  lft    ->  bool
vis (Vec v)    =   True
vis l          =   False
mis ::  lft    ->  bool
mis (Mat m)    =   True
mis l          =   False
tis ::  lft    ->  bool
tis (Ten t i)  =   True
tis l          =   False
```

- Cons to expression tree

```
cons :: lft       ->  (num -> expression) -> expression
cons (Vec v) f    =   V v
cons (Mat m) f    =   M m (f 1)
cons (Ten t i) f  =   T t i (f 1) (f 2)
```

- Head of expression tree

```
head :: expression  ->  lft
head (V v)          =   Vec v
head (M m e)        =   Mat m
head (T t i e f)    =   Ten t i
```

- Tail of expression tree

```
tail :: expression -> num  ->  expression
tail (M m e) 1             =   e
tail (T t i e f) 1         =   e
tail (T t i e f) 2         =   f
```

12.2.9 Square Bracket Application

Matrix application as defined in section 10.1.1 and tensor application as defined in section 10.1.3 are handled by a single generic function.

$$\begin{aligned} \texttt{app (Mat m) l} &\equiv \texttt{m}[\texttt{l}_1] \\ \texttt{app (Ten t i) l} &\equiv \texttt{t}[\texttt{l}_1,\texttt{l}_2] \end{aligned}$$

```
app :: lft       ->  (num -> expression) -> expression
app (Mat m) g    =   cons
                     (dot 1 (Mat m) (head (g 1)))
                     (tail (g 1))
app (Ten t i) g  =   cons
                     (dot 1 (dot 2 (Ten t i)
                     (head (g 2))) (head (g 1))) h
                      where  c    =  branch (head (g 1))
                             h i  =  tail (g 1) i, i<=c
                                  =  tail (g 2) (i-c),
                                     otherwise
```

12.2.10 Tensor Absorption Strategy

- The natural ordering on the special base interval $[0, \infty]$.

```
vlessv          ::  vector -> vector -> bool
vlessv v w      =   determinant (v,w) < 0
mlessv          ::  matrix -> vector -> bool
mlessv (v,w) x  =   vlessv v x & vlessv w x
mlessm          ::  matrix -> matrix -> bool
mlessm m (v,w)  =   mlessv m v & mlessv m w
```

- Disjoint information on the special base interval $[0, \infty]$.

```
mdisjointm      ::  matrix -> matrix -> bool
mdisjointm m n  =   mlessm m n \/ mlessm n m
```

- A fair strategy (see equation (11.9)).

$$\texttt{strategyf t i} \equiv \mathsf{strategy}_{\mathrm{f}}(\mathtt{t}, \mathtt{i})$$

```
strategyf ::  tensor -> num  ->  num
strategyf t i                =   i mod 2 + 1
```

- The information overlap strategy (see equation (11.10)).

$$\texttt{strategyo t i} \equiv \mathsf{strategy}_{\mathrm{o}}(\mathtt{t}, \mathtt{i})$$

```
strategyo      ::  tensor->num -> num
strategyo t i  =   strategyr t i, trefine t
               =   strategyf t i, otherwise
strategyr      ::  tensor -> num -> num
strategyr t i  =   2, mdisjointm
                   (fst (trans t)) (snd (trans t))
               =   1, otherwise
```

- Decision function (see equation (11.11)).

$$\begin{aligned}
\texttt{decision 1 (Mat m)} &\equiv \Delta_1(\mathtt{m}, \mathtt{i}) \\
\texttt{decision 1 (Ten t i)} &\equiv \Delta_1(\mathtt{t}, \mathtt{i}) \\
\texttt{decision 2 (Ten t i)} &\equiv \Delta_2(\mathtt{t}, \mathtt{i})
\end{aligned}$$

```
decision ::  num -> lft  ->  bool
decision 1 (Mat m)       =   True
decision 1 (Ten t i)     =   strategyo t i = 1
decision 2 (Ten t i)     =   strategyo t i = 2
```

12.2.11 Normalization Functions

- Sign emission function (see equation (11.1))

```
sem      ::  expression -> num -> sefp
sem e i  =   Spos (dem (0,0) (app ispos (one e)) i),
             refine (dot 1 ispos l)
         =   Sneg (dem (0,0) (app isneg (one e)) i),
             refine (dot 1 isneg l)
         =   Szer (dem (0,0) (app iszer (one e)) i),
             refine (dot 1 iszer l)
         =   Sinf (dem (0,0) (app isinf (one e)) i),
             refine (dot 1 isinf l)
         =   sem (app l f) i, otherwise
              where  l    =  head e
                     f d  =  ab l (tail e d)
                              (decision d l)
```

- Digit emission function (see equation (11.2))

```
dem            ::  digits -> expression -> num -> uefp
dem (i,c) e j  =   ((i,c),e), j=0 \/ vis l
               =   dem (i+1,2*c-1) (app idneg (one e))
                   (j-1), refine (dot 1 idneg l)
               =   dem (i+1,2*c+1) (app idpos (one e))
                   (j-1), refine (dot 1 idpos l)
               =   dem (i+1,2*c) (app idzer (one e))
               =   (j-1), refine (dot 1 idzer l)
               =   dem (i,c) (app l f) j, otherwise
                    where  l    =  head e
               =           f d  =  ab l (tail e d)
                                    (decision d l)
```

- Absorption function (see equation (11.14))

```
ab        ::  lft -> expression -> bool -> expression
ab k e b  =   utoe ((0,0),e), b = False
          =   utoe (dem (0,0) e 1),
              tis k & tis (head e)
          =   e, otherwise
```

12.2.12 Decimal Output Function

The function `eshow` takes an expression tree, `e` say, and a natural number, `i` say, as arguments and returns a string corresponding to the value of `e` in decimal evaluated to `i` exact floating point digits. In other words, the expression tree is evaluated according to the straightforward reduction rules until it has emitted a sign matrix followed by `i` digit matices. The partially evaluated exact floating point number is then converted into the decimal representation to as many decimal places as are valid. In particular, the last digit is guaranteed to be no more than 1 out. For instance

$$\begin{aligned}\texttt{0.314e1} &\equiv [3.13, 3.15]\\ \texttt{0.20} &\equiv [0.19, 0.21]\\ \texttt{0e-3} &\equiv [-0.001, 0.001].\end{aligned}$$

```
eshow ::  expression -> num  ->  [char]
eshow e i                    =   mshow (stom (sem e i))

mshow ::  matrix  ->  [char]
mshow m           =   show p, d=0 & q=1
                  =   (show p) ++ ''/'' ++ (show q),
                      d=0 & q~=1
                  =   sshow (scientific m 0),
                      d~=0
                       where d      = determinant m
                             (p,q)  = vscale (fst m)

sshow           ::  [num] -> [char]
sshow []        =   ''unbounded''
sshow (e : m)   =   (shows v) ++ (showm v) ++ (showe h)
                     where f        =  (foldr g 0).
                                       reverse
                           g d c    =  d+10*c
                           (h,l,v)  =  normalize
                                       (e,#m,f m)

normalize ::  (num,num,num)  ->  (num,num,num)
normalize (e,l,v)            =   normalize (e-1,l-1,v),
                                 l>0 & (abs v)<10^(l-1)
                             =   (e,l,v), otherwise

shows ::  num  ->  [char]
shows v        =   ''-'', v<0
                   '''', v>=0
```

```
showm ::  num  ->  [char]
showm v        =   ''0'', v=0
               =   ''0.''  ++ show (abs v), v~=0

showe ::  num  ->  [char]
showe e            ''e'' ++ (show e)

scientific      ::  matrix -> num -> [num]
scientific m n  =   [], vrefine (mdotv (inverse m) (1,0))
                =   n :  (mantissa (-9) 9 m),
                    mrefine (mdotm (inverse szer) m)
                =   scientific (mdotm ((1,0),(0,10)) m)
                    (n+1), otherwise

mantissa        ::  num -> num -> matrix -> [num]
mantissa i n m  =   i :  mantissa (-9) 9 (e i), c i
                =   mantissa (i+1) n m, i<n
                =   [], otherwise
                     where  c j  =  mrefine (mdotm
                                    (inverse (d j)) m)
                            d j  =  ((j+1,10),(j-1,10))
                            e j  =  mdotm
                                    ((10,0),(-j,1)) m
```

12.2.13 Elementary functions

- Convenient iterators for building expression trees (see equation (10.4)).

```
eiterate         ::  (num -> matrix) -> num ->
                     expression
eiterate i n     =   M (i n) (eiterate i (n+1))
eiteratex        ::  (num -> tensor) -> num ->
                     expression -> expression
eiteratex i n x  =   T (i n) 0 x (eiteratex i (n+1) x)
```

- Square root (see section 10.2.1).

$$\forall p, q \in \mathbb{N} \cdot \texttt{esqrtrat p q} \equiv \sqrt{\frac{p}{q}}$$

$$\forall x \in [0, \infty] \cdot \texttt{esqrtspos x} \equiv \sqrt{x}$$

```
esqrtrat         ::  num -> num -> expression
esqrtrat p q     =   rollover p q (p-q)
rollover         ::  num -> num -> num -> expression
rollover a b c   =   M dneg (rollover (4*a) d c), d >= 0
                 =   M dpos (rollover (-d) (4*b) c),
                     otherwise
                      where  d  =  2*(b-a)+c

esqrtspos        ::  expression -> expression
esqrtspos        =   eiteratex itersqrtspos 0
itersqrtspos     ::  num -> tensor
itersqrtspos n   =   (((1,0),(2,1)),((1,2),(0,1)))
```

- Logarithm (see section 10.2.2).

$$\forall \mathtt{x} \in [0,\infty] \cdot \texttt{elogspos x} \equiv \log\left(S_{+}\left(\mathtt{x}\right)\right)$$

Recall that

$$S_{+}\left(x\right) = x.$$

```
elogspos        ::  expression -> expression
elogspos        =   eiteratex iterlogspos 0
iterlogspos     ::  num -> tensor
iterlogspos 0   =   (((1,0),(1,1)),((-1,1),(-1,0)))
iterlogspos n   =   (((n,0),(2*n+1,n+1)),
                    ((n+1,2*n+1),(0,n)))
```

- Exponential (see section 10.2.3).

$$\texttt{ee} \equiv e$$

$$\forall \mathtt{x} \in [0,\infty] \cdot \texttt{eexpszer x} \equiv \exp\left(S_{0}\left(\mathtt{x}\right)\right)$$

Recall that

$$S_{0}\left(x\right) = \frac{x-1}{x+1}.$$

```
ee        ::  expression
ee        =   eiterate itere 0
itere     ::  num -> matrix
itere n   =   ((2*n+2,2*n+1),(2*n+1,2*n))

eexpszer        ::  expression -> expression
eexpszer        =   eiteratex iterexpszer 0
iterexpszer     ::  num -> tensor
iterexpszer n   =   (((2*n+2,2*n+1),(2*n+1,2*n)),
                    ((2*n,2*n+1),(2*n+1,2*n+2)))
```

- Pi (see section 10.2.4).

$$\mathtt{epi} \equiv \pi$$

```
epi           ::  expression
epi           =   T tdiv 0 (esqrtrat 10005 1) eomega
eomega        ::  expression
eomega        =   eiterate iteromega 0
iteromega     ::  num -> matrix
iteromega 0   =   ((6795705,213440),(6795704,213440))
iteromega n   =   ((e-d-c,e+d+c),(e+d-c,e-d+c))
                   where  b  =  (2*n-1)*(6*n-5)*(6*n-1)
                          c  =  b*(545140134*n+13591409)
                          d  =  b*(n+1)
                          e  =  10939058860032000*n^4
```

- Tangent (see section 10.2.5).

$$\forall \mathtt{x} \in [0,\infty] \cdot \mathtt{etanszer\ x} \equiv \tan\left(S_0\left(\mathtt{x}\right)\right)$$

```
etanszer        ::  expression -> expression
etanszer        =   eiteratex itertanszer 0
itertanszer     ::  num -> tensor
itertanszer 0   =   (((1,2),(1,0)),((-1,0),(-1,2)))
itertanszer n   =   (((2*n+1,2*n+3),(2*n-1,2*n+1)),
                    ((2*n+1,2*n-1),(2*n+3,2*n+1)))
```

- Inverse tangent (see section 10.2.6).

$$\forall \mathtt{x} \in [0,\infty] \cdot \mathtt{earctanszer\ x} \equiv \arctan\left(S_0\left(\mathtt{x}\right)\right)$$

```
earctanszer       ::  expression -> expression
earctanszer       =   eiteratex iterarctanszer 0
iterarctanszer    ::  num -> tensor
iterarctanszer 0  =   (((1,2),(1,0)),((-1,0),(-1,2)))
iterarctanszer n  =   (((2*n+1,n+1),(n,0)),
                      ((0,n),(n+1,2*n+1)))
```

Chapter 13

Theoretical Languages

In this chapter, following the work of Di Gianantonio [10] and Escardó [23], we incorporate the representation of the extended real numbers based on the composition of linear fractional transformations with integer coefficients into the Programming language for Computable Functions (PCF) [55, 29]. We present models for the extended language and show that they are computationally adequate with respect to the operational semantics.

Note that the proofs for the numerous lemmas and propositions in this chapter are located in appendix A for clarity.

13.1 PCF

The Programming Language for Computable Functions (PCF) devised by Plotkin [55] and described by Gunter [29] includes the terms of the simply-typed λ-calculus. The context-free grammar for PCF is given in BNF by

$$\begin{array}{rcl} x & \in & \mathsf{variable} \\ t & ::= & \mathsf{num} \mid \mathsf{bool} \mid t \to t \\ P & ::= & x \mid 0 \mid \mathsf{true} \mid \mathsf{false} \mid \\ & & \mathsf{succ}\,(P) \mid \mathsf{pred}\,(P) \mid \mathsf{zero}\,(P) \mid \\ & & \mathsf{if}\; P \;\mathsf{then}\; P \;\mathsf{else}\; P \mid \\ & & \lambda x \cdot P \mid PP \mid \mu x \cdot P \end{array}$$

where variable is the primitive syntax class of *variables*. The expressions in the syntax class over which t ranges are called *types*, and those over which P ranges are called *term trees*. The types num and bool are called the *ground* types for the *natural numbers* $\mathbb{N}$ and the *booleans* $\mathbb{B} = \{\mathsf{true}, \mathsf{false}\}$. The remaining types are called the *higher* types.

Term trees of the form $\lambda x.P$ are called *abstractions*, and those of the form PP are called *applications*. The other constructs of PCF include the *successor* – $\mathsf{succ}(P)$, *predecessor* – $\mathsf{pred}(P)$, *test for zero* – $\mathsf{zero}(P)$, *conditional* – $\mathsf{if}\, P\, \mathsf{then}\, P\, \mathsf{else}\, P$ and *recursion* – $\mu x.P$. The equivalence class of term trees modulo renaming of bound variables are called just *terms* and we refer to closed terms of ground type as *programs*. We will use the notation $[Q/x]P$ for *substitution* to indicate the result of replacing all free occurrences of the variable x in P by Q, making the appropriate changes in the bound variables of P so that no free variables in Q become bound.

There are two systems of rules describing PCF. The first of these determines which of the terms described by the syntax above are to be considered *well-typed*. These are the terms to which we will assign a meaning in our semantic model. The second set of rules form the *operational semantics* for evaluation.

13.1.1 Typing Rules

A *type assignment* is a list $H = x_1 : t_1, x_2 : t_2, \ldots, x_n : t_n$ of pairs of variables and types such that the variables are distinct. A *typing judgement* is a triple, denoted $H \vdash P : t$, consisting of a type assignment H, a term P and a type t such that all the free variables of P appear in the list H. We read this triple as "given the assignment H, the term P has type t". It is defined to be the least relation satisfying the well known typing rules [29]

$$G, x : t, H \vdash x : t$$

$$H \vdash \mathsf{0{:}num}$$

$$H \vdash \mathsf{true{:}bool}$$

$$H \vdash \mathsf{false{:}bool}$$

$$\frac{H \vdash P : \mathsf{num}}{H \vdash \mathsf{succ}\,(P) : \mathsf{num}}$$

$$\frac{H \vdash P : \mathsf{num}}{H \vdash \mathsf{pred}\,(P) : \mathsf{num}}$$

$$\frac{H \vdash P : \mathsf{num}}{H \vdash \mathsf{zero}\,(P) : \mathsf{bool}}$$

$$\frac{H \vdash P : \mathsf{bool} \quad H \vdash Q : t \quad H \vdash R : t}{H \vdash \mathsf{if}\, P\, \mathsf{then}\, Q\, \mathsf{else}\, R : t}$$

$$\frac{H, x : s \vdash P : t}{H \vdash \lambda x \cdot M : s \to t}$$

$$\frac{H \vdash P : s \to t \quad H \vdash Q : s}{H \vdash PQ : t}$$

$$\frac{H, x : t \vdash P : t}{H \vdash \mu x \cdot M : t}.$$

13.1.2 Operational Semantics

The strategy for evaluating a term is called an *operational semantics* for the language. One approach to describing such a semantics is to indicate how a term P evaluates to another term Q by defining a relation $P \rightarrow Q$ between terms using a set of *one-step reduction rules.* Let the symbols P, Q, R and S indicates arbitrary terms and let a *terminal value*, indicated using the symbols X and Y, be a term generated by

$$X = 0 \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{succ}\,(X) \mid \lambda x \cdot P \ \ .$$

The *one-step reduction relation* $\rightarrow$ is defined to be the least relation satisfying the one-step reduction rules for *call-by-name evaluation* of PCF given by

$$\frac{P \rightarrow Q}{\mathsf{succ}\,(P) \rightarrow \mathsf{succ}\,(Q)} \tag{13.1}$$

$$\mathsf{pred}\,(0) \rightarrow 0 \tag{13.2}$$

$$\mathsf{pred}\,(\mathsf{succ}\,(X)) \rightarrow X \tag{13.3}$$

$$\frac{P \rightarrow Q}{\mathsf{pred}\,(P) \rightarrow \mathsf{pred}\,(Q)} \tag{13.4}$$

$$\mathsf{zero}\,(0) \rightarrow \mathsf{true} \tag{13.5}$$

$$\mathsf{zero}\,(\mathsf{succ}\,(X)) \rightarrow \mathsf{false} \tag{13.6}$$

$$\frac{P \rightarrow Q}{\mathsf{zero}\,(P) \rightarrow \mathsf{zero}\,(Q)} \tag{13.7}$$

$$\mathsf{if}\;\mathsf{true}\;\mathsf{then}\;P\;\mathsf{else}\;Q \rightarrow P \tag{13.8}$$

$$\mathsf{if}\;\mathsf{false}\;\mathsf{then}\;P\;\mathsf{else}\;Q \rightarrow Q \tag{13.9}$$

$$\frac{P \rightarrow S}{\mathsf{if}\;P\;\mathsf{then}\;Q\;\mathsf{else}\;R \rightarrow \mathsf{if}\;S\;\mathsf{then}\;Q\;\mathsf{else}\;R} \tag{13.10}$$

$$(\lambda x \cdot P)\,Q \rightarrow [Q/x]\,P \tag{13.11}$$

$$\frac{P \rightarrow R}{PQ \rightarrow RQ} \tag{13.12}$$

$$\mu x \cdot P \rightarrow [\mu x \cdot P/x]\,P. \tag{13.13}$$

The *reduction relation* $\rightarrow^*$ is defined as the reflexive, transitive closure of the one-step reduction relation. We say that a term P evaluates to a term Q whenever $P \rightarrow^* Q$. Note that the one-step reduction relation is deterministic.

13.1.3 Denotational Semantics

We will use the *semantic brackets* $[\![\,]\!]$ to distinguish between terms in the language and expressions in the model. In the *standard model* for PCF, the number type num is interpreted as the flat dcpo of natural numbers - $[\![\mathsf{num}]\!] = \mathbb{N}_\perp = \mathbb{N} \cup \{\perp\}$, the boolean type bool is interpreted as the flat dcpo of booleans - $[\![\mathsf{bool}]\!] = \mathbb{B}_\perp = \{\mathsf{true}, \mathsf{false}, \perp\}$ and the function type $s \to t$ is interpreted as the dcpo of continuous functions from $[\![s]\!]$ to $[\![t]\!]$ - $[\![s \to t]\!] = [[\![s]\!] \to [\![t]\!]]$.

$$\begin{aligned}
[\![\mathsf{num}]\!] &= \mathbb{N}_\perp \\
[\![\mathsf{bool}]\!] &= \mathbb{B}_\perp \\
[\![s \to t]\!] &= [[\![s]\!] \to [\![t]\!]]
\end{aligned}$$

While a type assignment associates *types* with variables, an *environment* associates *values* to variables. If H is a type assignment, then an *H-environment* is a function ρ on variables that maps each $x : t \in H$ to a value $\rho(x) \in [\![t]\!]$.

Let us use the notation $[\![H \rhd P : t]\!]$ for the interpretation of term P relative to type assignment H and type t. Thus $[\![H \rhd P : t]\!]$ is a function from H-environments to $[\![t]\!]$ defined by induction on the type derivation of $H \vdash P : t$. For convenience, we will sometimes abbreviate $[\![H \rhd P : t]\!]\rho$ to just $[\![P]\!]$.

$$\begin{aligned}
[\![H \rhd x : t]\!]\rho &= \rho(x) \\
[\![H \rhd 0 : \mathsf{num}]\!]\rho &= 0 \\
[\![H \rhd \mathsf{true} : \mathsf{bool}]\!]\rho &= \mathsf{true} \\
[\![H \rhd \mathsf{false} : \mathsf{bool}]\!]\rho &= \mathsf{false}
\end{aligned}$$

$$\begin{aligned}
[\![H \rhd \mathsf{succ}(P) : \mathsf{num}]\!]\rho &= \begin{cases} [\![H \rhd P : \mathsf{num}]\!]\rho + 1 & \text{if } [\![H \rhd P : \mathsf{num}]\!]\rho \neq \perp \\ \perp & \text{if } [\![H \rhd P : \mathsf{num}]\!]\rho = \perp \end{cases} \\
[\![H \rhd \mathsf{pred}(P) : \mathsf{num}]\!]\rho &= \begin{cases} 0 & \text{if } [\![H \rhd P : \mathsf{num}]\!]\rho = 0 \\ [\![H \rhd P : \mathsf{num}]\!]\rho - 1 & \text{if } [\![H \rhd P : \mathsf{num}]\!]\rho > 0 \\ \perp & \text{if } [\![H \rhd P : \mathsf{num}]\!]\rho = \perp \end{cases} \\
[\![H \rhd \mathsf{zero}(P) : \mathsf{bool}]\!]\rho &= \begin{cases} \mathsf{true} & \text{if } [\![H \rhd P : \mathsf{num}]\!]\rho = 0 \\ \mathsf{false} & \text{if } [\![H \rhd P : \mathsf{num}]\!]\rho > 0 \\ \perp & \text{if } [\![H \rhd P : \mathsf{num}]\!]\rho = \perp \end{cases}
\end{aligned}$$

$$\begin{aligned}
[\![H \rhd \mathsf{if}\, P\, \mathsf{then}\, Q\, \mathsf{else}\, R : t]\!]\rho &= \begin{cases} [\![H \rhd Q : t]\!]\rho & \text{if } [\![H \rhd P : \mathsf{bool}]\!]\rho = \mathsf{true} \\ [\![H \rhd R : t]\!]\rho & \text{if } [\![H \rhd P : \mathsf{bool}]\!]\rho = \mathsf{false} \\ \perp & \text{if } [\![H \rhd P : \mathsf{bool}]\!]\rho = \perp \end{cases} \\
[\![H \rhd \lambda x \cdot P : s \to t]\!]\rho &= d \mapsto [\![H, x : s \rhd P : t]\!]\rho\,[x \mapsto d] \\
[\![H \rhd PQ : t]\!]\rho &= ([\![H \rhd P : s \to t]\!]\rho)\,([\![H \rhd Q : s]\!]\rho) \\
[\![H \rhd \mu x \cdot P : t]\!]\rho &= \mathsf{fix}\,(d \mapsto [\![H, x : t \rhd P : t]\!]\rho\,[x \mapsto d])
\end{aligned}$$

13.1.4 Computational Adequacy

Computational adequacy is a property of a given denotational semantics of a programming language with respect to its operational semantics. Roughly speaking, it says that the two coincide in the sense that anything that can be derived from the operational semantics can be derived from the denotational semantics and visa versa. Computational adequacy can be deduced from the combination of two other properties; namely *soundness* and *completeness.*

Proposition 64 (Soundness) *If P is a program and $P \rightarrow^* X$ then $[\![P]\!] = [\![X]\!]$.*

Theorem 65 (Completeness) *If P is a program and $[\![P]\!] = [\![X]\!]$ then $P \rightarrow^* X$.*

It is well known that the standard model for PCF is computationally adequate. In order to establish *completeness*, it is usual to consider Plotkin's notion of computability [57] and show that every term is computable.

Definition 66 *The* computable terms *of PCF form the least set of terms such that*

- *If $P : t$ is a program then P is computable whenever $[\![P]\!] = [\![X]\!]$ implies $P \rightarrow^* X$.*
- *If $\vdash P : s \rightarrow t$ then P is computable whenever PQ is computable for every closed computable term Q of type s.*
- *If $x_1 : t_1, x_2 : t_2, \ldots, x_n : t_n \vdash P : t$ then P is computable whenever*

  $$[P_1, P_2, \ldots, P_n/x_1, x_2, \ldots, x_n]\, P$$

  is computable for all closed computable terms P_i such that $\vdash P_i : t_i$.

Lemma 67 *Every PCF term P is computable.*

13.2 Language for Positive Reals

The Language for Positive Reals (LPR) [63] includes the syntax and conventions of PCF as described in section 13.1. The context-free grammar for LPR is given in BNF by

$$\begin{array}{rcl} x & \in & \mathsf{variable} \\ t & ::= & \mathsf{num} \mid \mathsf{bool} \mid \mathsf{real}^+ \mid t \times t \mid t \rightarrow t \\ P & ::= & x \mid 0 \mid \mathsf{true} \mid \mathsf{false} \mid \end{array}$$

$$\begin{array}{l}
\mathsf{succ}\,(P)\ |\, \mathsf{pred}\,(P)\ |\, \mathsf{zero}\,(P)\ | \\
\mathsf{if}\ P\ \mathsf{then}\ P\ \mathsf{else}\ P\ | \\
\lambda x\cdot P\ |\ PP\ |\ \mu x\cdot P\ | \\
(P,P)\ |\ \mathsf{fst}\,(P)\ |\ \mathsf{snd}\,(P)\ | \\
\langle P\rangle\,.
\end{array}$$

The new ground type real^+ represents the set $\mathbb{I}^{C}\mathbb{R}^+$ of closed intervals in $\mathbb{R}^+$. The types generated by

$$t = \mathsf{num}\ \big|\ \mathsf{bool}\ \big|\ \mathsf{real}^+\ |\ t\times t \tag{13.14}$$

are called the *ground* types. The new constructs are *pairing* – (P,Q), *first projection* – $\mathsf{fst}\,(P)$, *second projection* – $\mathsf{snd}\,(P)$ and *transformation* – $\langle P\rangle$. For convenience,

vector^+	denotes	$\mathsf{num}\times\mathsf{num}$,
matrix^+	denotes	$\mathsf{vector}^+\times\mathsf{vector}^+$ and
tensor^+	denotes	$\mathsf{matrix}^+\times\mathsf{matrix}^+$.

For example, the real number $\frac{3}{4}$ may be expressed as

$$\begin{aligned}
\tfrac{3}{4} &\equiv \langle(3,4)\rangle \\
&\equiv \left\langle\begin{array}{c}3\\4\end{array}\right\rangle
\end{aligned}$$

and the real number $\sqrt{2}$ may be expressed as

$$\begin{aligned}
\sqrt{2} &= \mu x\cdot\langle((1,1),(2,1))\rangle\,x \\
&\equiv \mu x\cdot\left\langle\begin{array}{cc}1&2\\1&1\end{array}\right\rangle x
\end{aligned}$$

and $1+2$ may be expressed as

$$\begin{aligned}
1+2 &\equiv \langle(((0,0),(1,0)),((1,0),(0,1)))\rangle\,(\langle(1,1)\rangle,\langle(2,1)\rangle) \\
&\equiv \left\langle\begin{array}{cccc}0&1&1&0\\0&0&0&1\end{array}\right\rangle\left(\left\langle\begin{array}{c}1\\1\end{array}\right\rangle,\left\langle\begin{array}{c}2\\1\end{array}\right\rangle\right).
\end{aligned}$$

13.2.1 Typing Rules

The typing judgement $H\vdash P:t$ for LPR is the least relation satisfying the typing rules for PCF as listed in section 13.1.1 together with

$$\frac{H\vdash P:s\quad H\vdash Q:t}{H\vdash(P,Q):s\times t}$$

$$\frac{H \vdash P : s \times t}{H \vdash \mathsf{fst}\,(P) : s}$$

$$\frac{H \vdash P : s \times t}{H \vdash \mathsf{snd}\,(P) : t}$$

$$\frac{H \vdash V{:}\mathsf{vector}^{+}}{H \vdash \langle V\rangle : \mathsf{real}^{+}}$$

$$\frac{H \vdash M : \mathsf{matrix}^{+}}{H \vdash \langle M\rangle : \mathsf{real}^{+} \to \mathsf{real}^{+}}$$

$$\frac{H \vdash T : \mathsf{tensor}^{+}}{H \vdash \langle T\rangle : \mathsf{real}^{+} \times \mathsf{real}^{+} \to \mathsf{real}^{+}}.$$

13.2.2 Operational Semantics

Let a *terminal value* be a term that cannot be reduced to another term. Let the symbols

P,Q, R, S	indicate	arbitrary terms,
X, Y	indicate	terminal values,
V, W	indicate	terminal values of type vector^{+},
V^{+}, W^{+}	indicate	terminal values of type vector^{+} excluding $(0,0)$,
M, N, O	indicate	(V, W) ,
M^{+}, N^{+}, O^{+}	indicate	(V^{+}, W^{+}) ,
T, U	indicate	(M, N) ,
T^{+}, U^{+}	indicate	(M^{+}, N^{+})

and the generated term

$$A \quad \text{indicates} \quad \langle V^{+}\rangle \mid \langle M^{+}\rangle\, P$$

and

L°	indicates	not L where $L \in \{X, Y, V, W, M, N, O, T, U, A\}$,
L^{-}	indicates	$(L^{+})^{\circ}$ where $L \in \{V, W, M, N, O, T, U\}$.

For example, V° indicates any term that is not a terminal value of type vector^{+} and V^{-} indicates $(0,0)$ or not a terminal value of type vector^{+}.

The *one-step reduction relation* $\to$ for LPR is defined as the least relation satisfying the one-step reduction rules for PCF as listed in section 13.1.2 together with the *product rules*

$$\frac{P \to R}{(P, Q) \to (R, Q)}$$

$$\frac{Q \to S}{(P, Q) \to (P, S)}$$
$$\mathsf{fst}\,(P, Q) \to P$$
$$\mathsf{snd}\,(P, Q) \to Q$$
$$\frac{P \to Q}{\mathsf{fst}\,(P) \to \mathsf{fst}\,(Q)}$$
$$\frac{P \to Q}{\mathsf{snd}\,(P) \to \mathsf{snd}\,(Q)},$$

the *transformation rule*

$$\frac{P \to Q}{\langle P \rangle \to \langle Q \rangle},$$

the *matrix rules*

$$\begin{array}{rcl} \langle M \rangle \left\langle V^{+} \right\rangle & \to & \left\langle M \bullet V^{+} \right\rangle \\ \langle M \rangle \left(\left\langle N^{+} \right\rangle P \right) & \to & \left\langle M \bullet N^{+} \right\rangle P \end{array}$$

$$\frac{P \to Q}{\langle M \rangle\, P \to \langle M \rangle\, Q}$$

and the *tensor rules*

$$\left\langle T^{+} \right\rangle P \to \left\langle \left(T^{+}\right)^{\mathrm{head}} \right\rangle \left(\left\langle \left(T^{+}\right)^{\mathrm{tail}} \right\rangle P \right)$$

$$\begin{array}{rcl} \langle T \rangle \left(\left\langle V^{+} \right\rangle, Q \right) & \to & \left\langle T \bullet_1 V^{+} \right\rangle Q \\ \langle T \rangle \left(P, \left\langle V^{+} \right\rangle \right) & \to & \left\langle T \bullet_2 V^{+} \right\rangle P \\ \langle T \rangle \left(\left\langle M^{+} \right\rangle P, Q \right) & \to & \left\langle T \bullet_1 M^{+} \right\rangle (P, Q) \\ \langle T \rangle \left(P, \left\langle M^{+} \right\rangle Q \right) & \to & \left\langle T \bullet_2 M^{+} \right\rangle (P, Q) \end{array}$$

$$\frac{P \to Q}{\langle T \rangle\, P \to \langle T \rangle\, Q}.$$

The product rules are non-deterministic. The transformation rule together with equation (13.12) ensures that the parameter in a transformation construct is completely evaluated before any matrix or tensor rule can be applied. The matrix rules allow information to be absorbed into a matrix. The tensor rules allow information to be emitted from and absorbed into a tensor.

13.2.3 Denotational Semantics

Let us extend the *standard model* for PCF as described in section 13.1.3. The real type real^+ may be interpreted as the continuous real domain $\mathbb{C}(\mathbb{R}^+)$ or the algebraic real domain $\mathbb{A}(\mathbb{Q}^+)$. The product type $s \times t$ will be interpreted as the lifted product of dcpos - $(\llbracket s \rrbracket \times \llbracket t \rrbracket)_\perp$.

$$\begin{aligned}
\llbracket \mathsf{real}^+ \rrbracket &= \mathbb{C}\left(\mathbb{R}^+\right) \\
\llbracket s \times t \rrbracket &= \left(\llbracket s \rrbracket \times \llbracket t \rrbracket\right)_\perp
\end{aligned}$$

The interpretation of the new constructs (P, P), $\mathsf{fst}(P)$, $\mathsf{snd}(P)$ and $\langle P \rangle$ are given by

$$\begin{aligned}
\llbracket (P, Q) \rrbracket &= \mathsf{up}\left(\llbracket P \rrbracket, \llbracket Q \rrbracket\right) \\
\llbracket \mathsf{fst}\,(P) \rrbracket &= \mathsf{fst}\left(\mathsf{down}\left(\llbracket P \rrbracket\right)\right) \\
\llbracket \mathsf{snd}\,(P) \rrbracket &= \mathsf{snd}\left(\mathsf{down}\left(\llbracket P \rrbracket\right)\right)
\end{aligned}$$

$$\begin{aligned}
\llbracket \langle V \rangle \rrbracket &= \begin{cases} [0,\infty] & \text{if } \llbracket V \rrbracket = \perp \\ \{\llbracket V \rrbracket\} & \text{otherwise} \end{cases} \\
\llbracket \langle M \rangle \rrbracket (X) &= \begin{cases} [0,\infty] & \text{if } \exists x \in X \cdot \llbracket M \rrbracket (x) = \perp \\ \llbracket M \rrbracket (X) & \text{otherwise} \end{cases} \\
\llbracket \langle T \rangle \rrbracket (X, Y) &= \begin{cases} [0,\infty] & \text{if } \exists x \in X \cdot \exists y \in Y \cdot \llbracket T \rrbracket (x, y) = \perp \\ \llbracket T \rrbracket (X, Y) & \text{otherwise.} \end{cases}
\end{aligned}$$

Note that

$$\begin{aligned}
\llbracket \langle V^+ \rangle \rrbracket &= \mathsf{info}\left(V^+\right) \\
\llbracket \langle M^+ \rangle \rrbracket \left([0,\infty]\right) &= \mathsf{info}\left(M^+\right) \\
\llbracket \langle T^+ \rangle \rrbracket \left([0,\infty], [0,\infty]\right) &= \mathsf{info}\left(T^+\right)
\end{aligned}$$

and

$$\begin{aligned}
\llbracket \langle V^- \rangle \rrbracket &= [0,\infty] \neq \mathbb{R}^\infty = \mathsf{info}\left(V^-\right) \\
\llbracket \langle M^- \rangle \rrbracket \left([0,\infty]\right) &= [0,\infty] \neq \mathbb{R}^\infty = \mathsf{info}\left(M^-\right) \\
\llbracket \langle T^- \rangle \rrbracket \left([0,\infty], [0,\infty]\right) &= [0,\infty] \neq \mathbb{R}^\infty = \mathsf{info}\left(T^-\right).
\end{aligned}$$

13.2.4 Computational Adequacy

LPR is a non-deterministic extension of PCF with meaningful programs that never terminate. However, Plotkin's notion of computability assumes that non-terminating programs are meaningless. Escardó [23] circumvented this problem by generalizing Plotkin's notion of computability by considering a program P to be computable whenever

$$[\![P]\!] \sqsubseteq \mathsf{eval}\,(P)$$

where

$$\mathsf{eval}\,(P) = \bigsqcup \{\mathsf{value}\,(Q) |\, P \to^* Q\}$$

and

$$\mathsf{value}\,(P) = \begin{cases} n & \text{if } P = \mathsf{succ}^n\,(0) \\ \mathsf{true} & \text{if } P = \mathsf{true} \\ \mathsf{false} & \text{if } P = \mathsf{false} \\ \bot & \text{otherwise.} \end{cases}$$

The evaluation function eval is well defined provided that the one-step reduction relation $\to$ is *sound*

$$\frac{P \to Q}{\mathsf{value}\,(P) \sqsubseteq \mathsf{value}\,(Q)}$$

and *weakly Church-Rosser*

$$\frac{P \to Q \quad P \to R}{\exists S \cdot Q \to^* S \quad R \to^* S}.$$

For PCF, the one-step reduction relation is sound because only terminal values have non-bottom values and the one-step reduction relation is weakly Church-Rosser because the reduction rules are deterministic.

Definition 68 *The* computable terms *of LPR form the least set of terms such that*

- *If $P : t$ is a program then P is computable whenever $[\![P]\!] \sqsubseteq \mathsf{eval}\,(P)$.*
- *If $\vdash P : s \to t$ then P is computable whenever PQ is computable for every closed computable term Q of type s.*
- *If $x_1 : t_1, x_2 : t_2, \ldots, x_n : t_n \vdash P : t$ then P is computable whenever*

$$[P_1, P_2, \ldots, P_n/x_1, x_2, \ldots, x_n]\,P$$

is computable for all closed computable terms P_i such that $\vdash P_i : t_i$.

Lemma 69 *If P is a PCF program then $[\![P]\!] \sqsubseteq \mathsf{eval}\,(P)$ iff $[\![P]\!] = [\![X]\!]$ implies $P \to^* X$.*

For LPR, the value function value is extended to

$$\mathsf{value}(P) = \begin{cases} n & \text{if } P = \mathsf{succ}^n(0) \\ \mathsf{true} & \text{if } P = \mathsf{true} \\ \mathsf{false} & \text{if } P = \mathsf{false} \\ \mathsf{info}(V^+) & \text{if } P = \langle V^+ \rangle \\ \mathsf{info}(M^+) & \text{if } P = \langle M^+ \rangle R \\ (\mathsf{value}(R), \mathsf{value}(S)) & \text{if } P = (R, S) \\ \bot & \text{otherwise.} \end{cases}$$

Lemma 70 *The one-step reduction relation for LPR is sound*

$$\frac{P \to Q}{\mathsf{value}(P) \sqsubseteq \mathsf{value}(Q)}.$$

Lemma 71 *The one-step reduction relation for LPR is weakly Church-Rosser*

$$\frac{P \to Q \quad P \to R}{\exists S \cdot Q \to^* S \quad R \to^* S}.$$

The last two lemmas ensure that the evaluation function eval is well defined. In order to establish soundness, namely

$$\mathsf{eval}(P) \sqsubseteq [\![P]\!],$$

the following two lemmas are required.

Lemma 72 *If P is an LPR program then $P \to Q$ implies $[\![P]\!] = [\![Q]\!]$.*

Lemma 73 *If P is an LPR program then* $\mathsf{value}(P) \sqsubseteq [\![P]\!]$.

Proposition 74 (Soundness) *If P is an LPR program then* $\mathsf{eval}(P) \sqsubseteq [\![P]\!]$.

In order to establish completeness of LPR, namely

$$[\![P]\!] \sqsubseteq \mathsf{eval}(P),$$

the following two lemmas are required.

Lemma 75 *An LPR program P is* computable *iff*

$$\forall a \ll [\![P]\!] \cdot \exists Q : P \to^* Q \cdot a \sqsubseteq \mathsf{value}(Q).$$

Lemma 76 *For any closed LPR term $P : s \to t$, where s and t are ground types, if $[\![P]\!]$ is continuous and*

$$\forall R : Q \to^* R \cdot \exists S : PQ \to^* S \cdot [\![P]\!] \left(\mathsf{value}\left(R\right)\right) \sqsubseteq \mathsf{value}\left(S\right)$$

for every closed computable term $Q : s$ then $P : s \to t$ is computable.

So, we need to prove that every LPR term is computable, establishing theorem 113 below, by extending the inductive proof of Plotkin [56] with the following lemmas.

Lemma 77 *The pairing construct is computable.*

Lemma 78 *The projection constructs are computable.*

Lemma 79

$$\forall Q : P \to^* Q \cdot \exists R : \langle M \rangle\, P \to^* R \cdot [\![\langle M \rangle]\!] \left(\mathsf{value}\left(Q\right)\right) = \mathsf{value}\left(R\right).$$

Lemma 80

$$\forall Q : P \to^* Q \cdot \exists R : \langle T \rangle\, P \to^* R \cdot [\![\langle T \rangle]\!] \left(\mathsf{value}\left(Q\right)\right) = \mathsf{value}\left(R\right).$$

Lemma 81 *The transformation construct is computable.*

Lemma 82 *Every term P of LPR is computable.*

Theorem 83 (Completeness) *If P is a program then*

$$[\![P]\!] \sqsubseteq \mathsf{eval}\left(P\right).$$

The soundness and completeness properties are together called the *computational adequacy property*. We have shown that LPR is computational adequate. This result means that a mathematical proof of correctness of a recursive algorithm is sufficient to conclude that a program induced by it produces the correct result. In other words, a syntactic proof for the program by appealing to the operational semantics is unnecessary. This language does not incorporate any strategy for tensor absorption and therefore says nothing about their correctness. However, computational adequacy does say that a correct strategy exists.

13.3 Language for All Reals

The Language for All Reals (LAR) is a further development of work by Edalat, Sünderhauf and myself [21] to capture multi-valued functions. It includes the syntax and conventions of PCF as described in section 13.1. The context-free grammar for LAR is given in BNF by

$$\begin{array}{rcl}
x & \in & \mathsf{variable} \\
t & ::= & \mathsf{num} \mid \mathsf{bool} \mid \mathsf{real}^{\infty} \mid \mathsf{real}^{+} \mid t \times t \mid t \to t \\
P & ::= & x \mid 0 \mid \mathsf{true} \mid \mathsf{false} \mid \\
 & & \mathsf{succ}\,(P) \mid \mathsf{pred}\,(P) \mid \mathsf{zero}\,(P) \mid \\
 & & \mathsf{if}\; P\, \mathsf{then}\; P\, \mathsf{else}\; P \mid \\
 & & \lambda x \cdot P \mid PP \mid \mu x \cdot P \mid \\
 & & (P, P) \mid \mathsf{fst}\,(P) \mid \mathsf{snd}\,(P) \mid \\
 & & \{P, P\} \mid P \lhd P \mid \\
 & & [P] \mid [P\rangle \mid \langle P\rangle\,.
\end{array}$$

The new ground types real^{∞} and real^{+} represent the sets $\mathbb{I}^{\mathrm{C}}\mathbb{R}^{\infty}$ and $\mathbb{I}^{\mathrm{C}}\mathbb{R}^{+}$ of closed intervals in $\mathbb{R}^{\infty}$ and $\mathbb{R}^{+}$ respectively. The types generated by

$$t = \mathsf{num} \mid \mathsf{bool} \mid \mathsf{real}^{\infty} \mid \mathsf{real}^{+} \mid t \times t \tag{13.15}$$

are called the *ground* types. The new constructs are *pairing* – (P, Q), *first projection* – $\mathsf{fst}\,(P)$, *second projection* – $\mathsf{snd}\,(P)$, *parallel* - $\{P, P\}$, *squeeze* $P \lhd P$ and *transformation* – $[P]$, $[P\rangle$, $\langle P\rangle$. For convenience, int denotes $\mathsf{num} \times \mathsf{num}$ representing an integer in the usual way (i.e. (n, m) represents $n - m$),

vector^{∞}	denotes	$\mathsf{int} \times \mathsf{int}$,
vector^{+}	denotes	$\mathsf{num} \times \mathsf{num}$,
matrix^{∞}	denotes	$\mathsf{vector}^{\infty} \times \mathsf{vector}^{\infty}$
matrix^{+}	denotes	$\mathsf{vector}^{+} \times \mathsf{vector}^{+}$
tensor^{∞}	denotes	$\mathsf{matrix}^{\infty} \times \mathsf{matrix}^{\infty}$
tensor^{+}	denotes	$\mathsf{matrix}^{+} \times \mathsf{matrix}^{+}$.

For example, a typical finite general normal product for $-\frac{7}{32}$ is

$$-\tfrac{7}{16} \equiv \left[\begin{array}{cc} -2 & 1 \\ 1 & 1 \end{array}\right\rangle \left\langle\begin{array}{cc} 1 & 3 \\ 2 & 4 \end{array}\right\rangle \left\langle\begin{array}{cc} 1 & 3 \\ 2 & 4 \end{array}\right\rangle \left\langle\begin{array}{c} 1 \\ 3 \end{array}\right\rangle .$$

Note the use of brackets to indicate type. Furthermore, $\left(-\frac{7}{16}\right)^{-1}$ may be expressed as

$$\left(-\tfrac{7}{16}\right)^{-1} = \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}\right] \left[\begin{array}{cc} -2 & 1 \\ 1 & 1 \end{array}\right\rangle \left\langle\begin{array}{cc} 1 & 3 \\ 2 & 4 \end{array}\right\rangle \left\langle\begin{array}{cc} 1 & 3 \\ 2 & 4 \end{array}\right\rangle \left\langle\begin{array}{c} 1 \\ 3 \end{array}\right\rangle .$$

The parallel and squeeze constructors are used to form the redundant if operator as defined in equation (7.7)

$$\begin{aligned}
&\mathsf{rif}\, x < (M([0,\infty]), N([0,\infty]))\ \mathsf{then}\, f\ \mathsf{else}\, g \\
=& \begin{cases} f(x) & \text{if } [0,\infty] \ll M^{\dagger}(x) \\ g(x) & \text{if } [0,\infty] \ll N^{\dagger}(x) \end{cases} \\
\equiv& \left\{ f \lhd \left[M^{\dagger}\right](x), g \lhd \left[N^{\dagger}\right](x) \right\}.
\end{aligned}$$

The parallel construct allows the two if conditions to be evaluated in parallel while the squeeze operator allows the if conditions to be incrementally tested.

13.3.1 Typing Rules

The typing judgement $H \vdash P : t$ for LAR is the least relation satisfying the typing rules for PCF as listed in section 13.1.1 together with

$$\frac{H \vdash P : s \quad H \vdash Q : t}{H \vdash (P, Q) : s \times t}$$

$$\frac{H \vdash P : s \times t}{H \vdash \mathsf{fst}(P) : s}$$

$$\frac{H \vdash P : s \times t}{H \vdash \mathsf{snd}(P) : t}$$

$$\frac{P : t \quad Q : t}{H \vdash \{P, Q\} : t}$$

$$\frac{H \vdash P{:}\mathsf{real}^{+} \to t \quad H \vdash Q{:}\mathsf{real}^{\infty}}{H \vdash P \lhd Q{:}\, t}$$

$$\frac{H \vdash V{:}\mathsf{vector}^{\infty}}{H \vdash [V\rangle : \mathsf{real}^{\infty}}$$

$$\frac{H \vdash V{:}\mathsf{vector}^{+}}{H \vdash \langle V\rangle : \mathsf{real}^{+}}$$

$$\frac{H \vdash M : \mathsf{matrix}^{\infty}}{H \vdash [M] : \mathsf{real}^{\infty} \to \mathsf{real}^{\infty}}$$

$$\frac{H \vdash M : \mathsf{matrix}^{\infty}}{H \vdash [M\rangle : \mathsf{real}^{+} \to \mathsf{real}^{\infty}}$$

$$\frac{H \vdash M : \mathsf{matrix}^{+}}{H \vdash \langle M\rangle : \mathsf{real}^{+} \to \mathsf{real}^{+}}$$

$$\frac{H \vdash T : \mathsf{tensor}^{\infty}}{H \vdash [T] : \mathsf{real}^{\infty} \times \mathsf{real}^{\infty} \to \mathsf{real}^{\infty}}$$

$$\frac{H \vdash T : \mathsf{tensor}^{\infty}}{H \vdash [T\rangle : \mathsf{real}^{+} \times \mathsf{real}^{+} \to \mathsf{real}^{\infty}}$$

$$\frac{H \vdash T : \mathsf{tensor}^+}{H \vdash \langle T \rangle : \mathsf{real}^+ \times \mathsf{real}^+ \to \mathsf{real}^+}.$$

13.3.2 Operational Semantics

Let a *terminal value* be a term that cannot be reduced to another term. Let the symbols

P, Q, R, S	indicate	arbitrary terms,
X, Y	indicate	terminal values,
V, W	indicate	terminal values of type vector^∞ or vector^+,
V^+, W^+	indicate	terminal values of type vector^+ excluding $(0, 0)$,
M, N, O	indicate	(V, W),
M^+, N^+, O^+	indicate	(V^+, W^+),
T, U	indicate	(M, N),
T^+, U^+	indicate	(M^+, N^+),

and the generated terms

A	indicates	$[V\rangle \mid \langle V^+\rangle \mid [M\rangle\, P \mid \langle M^+\rangle\, P$,
B	indicates	$([V\rangle, [V\rangle) \mid ([M\rangle\, P, [V\rangle) \mid ([V\rangle, [M\rangle\, P) \mid ([M\rangle\, P, [M\rangle\, P)$,
C	indicates	$(\langle V^+\rangle, P) \mid (\langle M^+\rangle\, P, P) \mid (P, \langle V^+\rangle) \mid (P, \langle M^+\rangle\, P)$

and

L°	indicates	not L where $L \in \{X, Y, V, W, M, N, O, T, U, A, B, C\}$,
L^-	indicates	$(L^+)^\circ$ where $L \in \{V, W, M, N, O, T, U\}$.

For example, V° indicates any term that is not a terminal value of type vector^∞ or vector^+.

The *one-step reduction relation* $\to$ for LAR is defined as the least relation satisfying the one-step reduction rules for PCF as listed in section 13.1.2 together with the *product rules*

$$\frac{P \to R \quad Q \to S}{(P, Q) \to (R, S)}$$

$$\frac{P \to Q}{(P, X) \to (Q, X)}$$

$$\frac{P \to Q}{(X, P) \to (X, Q)}$$

$$\mathsf{fst}\,(P, Q) \to P$$

$$\mathsf{snd}\,(P, Q) \to Q$$

$$\frac{P \neq (R,S) \quad P \to Q}{\mathsf{fst}\,(P) \to \mathsf{fst}\,(Q)}$$
$$\frac{P \neq (R,S) \quad P \to Q}{\mathsf{snd}\,(P) \to \mathsf{snd}\,(Q)},$$

the *parallel rules*

$$\begin{array}{rcl}
\mathsf{succ}\,(\{P,Q\}) & \to & \{\mathsf{succ}\,(P)\,,\mathsf{succ}\,(Q)\} \\
\mathsf{pred}\,(\{P,Q\}) & \to & \{\mathsf{pred}\,(P)\,,\mathsf{pred}\,(Q)\} \\
\mathsf{zero}\,(\{P,Q\}) & \to & \{\mathsf{zero}\,(P)\,,\mathsf{zero}\,(Q)\} \\
\mathsf{if}\ \{P,Q\}\ \mathsf{then}\,R\,\mathsf{else}\,S & \to & \{\mathsf{if}\,P\,\mathsf{then}\,R\,\mathsf{else}\,S, \mathsf{if}\,Q\,\mathsf{then}\,R\,\mathsf{else}\,S\} \\
\lambda x \cdot \{P,Q\} & \to & \{\lambda x \cdot P, \lambda x \cdot Q\} \\
\{P,Q\}\,R & \to & \{PR, QR\} \\
P\,\{Q,R\} & \to & \{PQ, PR\} \\
(\{P,Q\}\,,R) & \to & \{(P,R)\,,(Q,R)\} \\
(P,\{Q,R\}) & \to & \{(P,Q)\,,(P,R)\} \\
\mathsf{fst}\,(\{P,Q\}) & \to & \{\mathsf{fst}\,(P)\,,\mathsf{fst}\,(Q)\} \\
\mathsf{snd}\,(\{P,Q\}) & \to & \{\mathsf{snd}\,(P)\,,\mathsf{snd}\,(Q)\} \\
P \lhd \{Q,R\} & \to & \{P \lhd Q, P \lhd R\} \\
[\{P,Q\}] & \to & \{[P]\,,[Q]\} \\
[\{P,Q\}\rangle & \to & \{[P\rangle\,,[Q\rangle\} \\
\langle\{P,Q\}\rangle & \to & \{\langle P\rangle\,,\langle Q\rangle\}
\end{array}$$

$$\frac{P \to R \quad Q \to S}{\{P,Q\} \to \{R,S\}}$$
$$\frac{P \to Q}{\{P,X\} \to \{Q,X\}}$$
$$\frac{P \to Q}{\{X,P\} \to \{X,Q\}},$$

(note that the parallel construct does not distribute through the recursion operator, the "then" part of conditional, the "else" part of conditional and the function part of squeeze), the *squeeze rules*

$$\frac{A^\circ \to Q}{P \lhd A^\circ \to P \lhd Q}$$
$$\frac{\mathsf{info}\,(V) \subseteq (0,\infty)}{P \lhd [V\rangle \to P\,\langle|V|\rangle}$$

$$\frac{\mathsf{info}\,(M) \subseteq (0,\infty)}{P \lhd [M\rangle\, Q \to P\,(\langle|M|\rangle\, Q)}$$
$$\frac{\mathsf{info}\,(M) \not\subseteq (0,\infty) \quad [M\rangle\, Q \to R}{P \lhd [M\rangle\, Q \to P \lhd R},$$

the *transformation rules*

$$\frac{P \to Q}{[P] \to [Q]}$$
$$\frac{P \to Q}{[P\rangle \to [Q\rangle}$$
$$\frac{P \to Q}{\langle P\rangle \to \langle Q\rangle},$$

the *matrix rules*

$$\begin{array}{rcl} [M]\,[V\rangle & \to & [M \bullet V\rangle \\ [M]\,([N\rangle\, P) & \to & [M \bullet N\rangle\, P \end{array}$$
$$\frac{A^\circ \to P}{[M]\,A^\circ \to [M]\,P}$$

$$\begin{array}{rcl} [M\rangle\,\left\langle V^+\right\rangle & \to & \left[M \bullet V^+\right\rangle \\ [M\rangle\,\left(\left\langle N^+\right\rangle P\right) & \to & \left[M \bullet N^+\right\rangle P \end{array}$$
$$\frac{A^\circ \to P}{[M\rangle\,A^\circ \to [M\rangle\,P}$$

$$\begin{array}{rcl} \langle M\rangle\,\left\langle V^+\right\rangle & \to & \left\langle M \bullet V^+\right\rangle \\ \langle M\rangle\,\left(\left\langle N^+\right\rangle P\right) & \to & \left\langle M \bullet N^+\right\rangle P \end{array}$$
$$\frac{A^\circ \to P}{\langle M\rangle\,A^\circ \to \langle M\rangle\,P}$$

and the *tensor rules*

$$\begin{array}{rcl} [T]\,([V\rangle\,, [W\rangle) & \to & [T \bullet_1 V \bullet W\rangle \\ [T]\,([V\rangle\,, [N\rangle\, Q) & \to & [T \bullet_1 V \bullet N\rangle\, Q \\ [T]\,([M\rangle\, P, [W\rangle) & \to & [T \bullet_1 M \bullet_2 W\rangle\, P \\ [T]\,([M\rangle\, P, [N\rangle\, Q) & \to & [T \bullet_1 M \bullet_2 N\rangle\,(P, Q) \end{array}$$
$$\frac{B^\circ \to P}{[T]\,B^\circ \to [T]\,P}$$

$$\frac{\mathsf{info}\,(T) = \mathbb{R}^\infty}{[T\rangle \to [T\rangle'}$$

$$\frac{\mathsf{info}\,(T) \neq \mathbb{R}^\infty}{[T\rangle\, P \to [T^{\text{head}}\rangle\, \big(\langle T^{\text{tail}}\rangle'\, P\big)}$$

$$\begin{aligned}
[T\rangle'\,\big(\big\langle V^+\big\rangle, \big\langle W^+\big\rangle\big) &\quad\to\quad \big[T \bullet_1 V^+ \bullet W^+\big\rangle \\
[T\rangle'\,\big(\big\langle V^+\big\rangle, \big\langle N^+\big\rangle\, Q\big) &\quad\to\quad \big[T \bullet_1 V^+ \bullet N^+\big\rangle\, Q \\
[T\rangle'\,\big(\big\langle M^+\big\rangle\, P, \big\langle W^+\big\rangle\big) &\quad\to\quad \big[T \bullet_1 M^+ \bullet_2 W^+\big\rangle\, P \\
[T\rangle'\,\big(\big\langle M^+\big\rangle\, P, \big\langle N^+\big\rangle\, Q\big) &\quad\to\quad \big[T \bullet_1 M^+ \bullet_2 N^+\big\rangle\, (P, Q)
\end{aligned}$$

$$\begin{aligned}
[T\rangle'\,\big(\big\langle V^+\big\rangle, A^\circ\big) &\quad\to\quad \big[T \bullet_1 V^+\big\rangle\, A^\circ \\
[T\rangle'\,\big(\big\langle M^+\big\rangle\, P, A^\circ\big) &\quad\to\quad \big[T \bullet_1 M^+\big\rangle\, (P, A^\circ) \\
[T\rangle'\,\big(A^\circ, \big\langle W^+\big\rangle\big) &\quad\to\quad \big[T \bullet_2 W^+\big\rangle\, A^\circ \\
[T\rangle'\,\big(A^\circ, \big\langle N^+\big\rangle\, Q\big) &\quad\to\quad \big[T \bullet_2 N^+\big\rangle\, (A^\circ, Q)
\end{aligned}$$

$$\frac{C^\circ \to P}{[T\rangle'\, C^\circ \to [T\rangle'\, P}$$

$$\begin{aligned}
\langle T^+\rangle\, P &\quad\to\quad \Big\langle \big(T^+\big)^{\text{head}} \Big\rangle \left(\Big\langle \big(T^+\big)^{\text{tail}} \Big\rangle'\, P \right) \\
\langle T^-\rangle\, P &\quad\to\quad \left\langle \begin{matrix} \mathsf{succ}\,(0) & 0 \\ 0 & \mathsf{succ}\,(0) \end{matrix} \right\rangle \big(\langle T^-\rangle'\, P\big)
\end{aligned}$$

$$\begin{aligned}
\langle T\rangle'\,\big(\big\langle V^+\big\rangle, \big\langle W^+\big\rangle\big) &\quad\to\quad \big\langle T \bullet_1 V^+ \bullet W^+\big\rangle \\
\langle T\rangle'\,\big(\big\langle V^+\big\rangle, \big\langle N^+\big\rangle\, Q\big) &\quad\to\quad \big\langle T \bullet_1 V^+ \bullet N^+\big\rangle\, Q \\
\langle T\rangle'\,\big(\big\langle M^+\big\rangle\, P, \big\langle W^+\big\rangle\big) &\quad\to\quad \big\langle T \bullet_1 M^+ \bullet_2 W^+\big\rangle\, P \\
\langle T\rangle'\,\big(\big\langle M^+\big\rangle\, P, \big\langle N^+\big\rangle\, Q\big) &\quad\to\quad \big\langle T \bullet_1 M^+ \bullet_2 N^+\big\rangle\, (P, Q)
\end{aligned}$$

$$\begin{aligned}
\langle T\rangle'\,\big(\big\langle V^+\big\rangle, A^\circ\big) &\quad\to\quad \big\langle T \bullet_1 V^+\big\rangle\, A^\circ \\
\langle T\rangle'\,\big(\big\langle M^+\big\rangle\, P, A^\circ\big) &\quad\to\quad \big\langle T \bullet_1 M^+\big\rangle\, (P, A^\circ) \\
\langle T\rangle'\,\big(A^\circ, \big\langle W^+\big\rangle\big) &\quad\to\quad \big\langle T \bullet_2 W^+\big\rangle\, A^\circ \\
\langle T\rangle'\,\big(A^\circ, \big\langle N^+\big\rangle\, Q\big) &\quad\to\quad \big\langle T \bullet_2 N^+\big\rangle\, (A^\circ, Q)
\end{aligned}$$

$$\frac{C^\circ \to P}{\langle T\rangle'\, C^\circ \to \langle T\rangle'\, P}.$$

Note that the one-step reduction relation $\rightarrow$ for LAR is deterministic in contrast to the one-step reduction relation $\rightarrow$ for LPR. This reflects the fact that a *fair strategy* for tensor absorption as explained in section 11.9 has been incorporated into the product rules and the tensor rules. The transformation rule together with equation (13.12) ensures that the parameter in a transformation construct is completely evaluated before any matrix or tensor rule can be applied. The matrix rules allow information to be absorbed into a matrix. The tensor rules allow information to be emitted from and absorbed into a tensor. The unprimed transformation constructs $[T\rangle$ and $\langle T\rangle$ indicate "emission required next", while the primed transformation constructs $[T\rangle'$ and $\langle T\rangle'$ indicate "absorption required next". The primed constructs are not intended for direct use by the programmer.

13.3.3 Denotational Semantics

Let us extend the *standard model* for PCF as described in section 13.1.3. The real type real^{∞} may be interpreted as the continuous real domain $\mathbb{C}(\mathbb{R}^{\infty})$ or the algebraic real domain $\mathbb{A}(\mathbb{Q}^{\infty})$. Note that $[\![Q]\!] \subseteq (0,\infty)$ is equivalent to $[0,\infty] \ll [\![Q]\!]$ in the continuous real domain $\mathbb{C}(\mathbb{R}^{\infty})$. The real type real^{+} may be interpreted as the continuous real domain $\mathbb{C}(\mathbb{R}^{+})$ or the algebraic real domain $\mathbb{A}(\mathbb{Q}^{+})$. The product type $s \times t$ will be interpreted as the lifted product of dcpos - $([\![s]\!] \times [\![t]\!])_{\perp}$.

$$\begin{aligned}
[\![\mathsf{real}^{\infty}]\!] &= \mathbb{C}(\mathbb{R}^{\infty}) \\
[\![\mathsf{real}^{+}]\!] &= \mathbb{C}(\mathbb{R}^{+}) \\
[\![s \times t]\!] &= ([\![s]\!] \times [\![t]\!])_{\perp}
\end{aligned}$$

The interpretation of the new constructs (P,P), $\mathsf{fst}(P)$, $\mathsf{snd}(P)$, $P \lhd P$, $[P]$, $[P\rangle$ and $\langle P\rangle$ are given by

$$\begin{aligned}
[\![(P,Q)]\!] &= \mathsf{up}([\![P]\!],[\![Q]\!]) \\
[\![\mathsf{fst}(P)]\!] &= \mathsf{fst}(\mathsf{down}([\![P]\!])) \\
[\![\mathsf{snd}(P)]\!] &= \mathsf{snd}(\mathsf{down}([\![P]\!]))
\end{aligned}$$

$$\begin{aligned}
[\![P \lhd Q]\!] &= \begin{cases} [\![P]\!][\![Q]\!] & \text{if } [\![Q]\!] \subseteq (0,\infty) \\ \perp & \text{otherwise} \end{cases} \\
&= ([0,\infty] \searrow ([\![P]\!][\![Q]\!]))([\![Q]\!])
\end{aligned}$$

$$\begin{aligned}
[\![[M]]\!](X) &= \begin{cases} \mathbb{R}^{\infty} & \text{if } \exists x \in X \cdot [\![M]\!](x) = \perp \\ [\![M]\!](X) & \text{otherwise} \end{cases} \\
[\![[T]]\!](X,Y) &= \begin{cases} \mathbb{R}^{\infty} & \text{if } \exists x \in X \cdot \exists y \in Y \cdot [\![T]\!](x,y) = \perp \\ [\![T]\!](X,Y) & \text{otherwise} \end{cases}
\end{aligned}$$

$$\begin{aligned}
\llbracket [V\rangle \rrbracket &= \begin{cases} \mathbb{R}^\infty & \text{if } \llbracket V \rrbracket = \bot \\ \{\llbracket V \rrbracket\} & \text{otherwise} \end{cases} \\
\llbracket [M\rangle \rrbracket (X) &= \begin{cases} \mathbb{R}^\infty & \text{if } \exists x \in X \cdot \llbracket M \rrbracket (x) = \bot \\ \llbracket M \rrbracket (X) & \text{otherwise} \end{cases} \\
\llbracket [T\rangle \rrbracket (X, Y) &= \begin{cases} \mathbb{R}^\infty & \text{if } \exists x \in X \cdot \exists y \in Y \cdot \llbracket T \rrbracket (x, y) = \bot \\ \llbracket T \rrbracket (X, Y) & \text{otherwise} \end{cases} \\
\llbracket \langle V\rangle \rrbracket &= \begin{cases} [0, \infty] & \text{if } \llbracket V \rrbracket = \bot \\ \{\llbracket V \rrbracket\} & \text{otherwise} \end{cases} \\
\llbracket \langle M\rangle \rrbracket (X) &= \begin{cases} [0, \infty] & \text{if } \exists x \in X \cdot \llbracket M \rrbracket (x) = \bot \\ \llbracket M \rrbracket (X) & \text{otherwise} \end{cases} \\
\llbracket \langle T\rangle \rrbracket (X, Y) &= \begin{cases} [0, \infty] & \text{if } \exists x \in X \cdot \exists y \in Y \cdot \llbracket T \rrbracket (x, y) = \bot \\ \llbracket T \rrbracket (X, Y) & \text{otherwise.} \end{cases}
\end{aligned}$$

Note that

$$\begin{aligned}
\llbracket [V\rangle \rrbracket &= \mathsf{info}\,(V) \\
\llbracket [M\rangle \rrbracket ([0, \infty]) &= \mathsf{info}\,(M) \\
\llbracket [T\rangle \rrbracket ([0, \infty], [0, \infty]) &= \mathsf{info}\,(T)
\end{aligned}$$

and

$$\begin{aligned}
\llbracket \langle V^+\rangle \rrbracket &= \mathsf{info}\left(V^+\right) \\
\llbracket \langle M^+\rangle \rrbracket ([0, \infty]) &= \mathsf{info}\left(M^+\right) \\
\llbracket \langle T^+\rangle \rrbracket ([0, \infty], [0, \infty]) &= \mathsf{info}\left(T^+\right)
\end{aligned}$$

and

$$\begin{aligned}
\llbracket \langle V^-\rangle \rrbracket &= [0, \infty] \\
&\neq \mathbb{R}^\infty \\
&= \mathsf{info}\left(V^+\right) \\
\llbracket \langle M^-\rangle \rrbracket ([0, \infty]) &= [0, \infty] \\
&\neq \mathbb{R}^\infty \\
&= \mathsf{info}\left(M^+\right) \\
\llbracket \langle T^-\rangle \rrbracket ([0, \infty], [0, \infty]) &= [0, \infty] \\
&\neq \mathbb{R}^\infty \\
&= \mathsf{info}\left(T^+\right).
\end{aligned}$$

At this point, consider the monad $(\text{LOWER}, \eta, \mu)$ in the category of continuous Scott domains and continuous functions where LOWER is the lower powerdomain operator and

$$\begin{aligned}
\eta_D &: D \to \mathrm{LOWER}(D) \\
\eta_D(x) &= \downarrow\{x\} \\
\mu_D &: \mathrm{LOWER}(\mathrm{LOWER}(D)) \to \mathrm{LOWER}(D) \\
\mu_D(X) &= \bigsqcup X.
\end{aligned}$$

Recall Theorem 4, which states that the lower powerdomain of a continuous domain $(D, \sqsubseteq)$ is isomorphic to the lattice of all non-empty Scott closed subsets of D and so

$$\bigsqcup X = \mathsf{closure}\left(\bigcup X\right).$$

This monad induces a new model for LAR, which we will call the *parallel model.* We will use the *semantic brackets* $(\!|\ |\!)$ to distinguish between terms in the language, expressions in the standard model and expressions in the parallel model. In the parallel model, the ground types are interpreted as the lower powerdomain of the interpretation in the standard model, while the higher types merely reflect this change. In other words, for any ground type t

$$(\!|t|\!) = \mathrm{LOWER}(\llbracket t \rrbracket)$$

and for any higher type $s \to t$

$$(\!|s \to t|\!) = [(\!|s|\!) \to (\!|t|\!)].$$

Definition 84 *For any $x : \llbracket t \rrbracket$ where t is a ground type*

$$\begin{aligned}
\hat{x} &: (\!|t|\!) \\
\hat{x} &= \downarrow\{x\} : (\!|t|\!)
\end{aligned}$$

and for any $f : \llbracket s \to t \rrbracket$ where $s \to t$ is a higher type

$$\begin{aligned}
\hat{f} &: (\!|s \to t|\!) \\
\hat{f} &= X \mapsto \bigsqcup_{x : \hat{x} \sqsubseteq X} \widehat{f(x)}.
\end{aligned}$$

Lemma 85 *For any $f : \llbracket s \to t \rrbracket$ where $s \to t$ is a higher type*

$$\hat{f}(\hat{x}) = \widehat{f(x)}.$$

The lower powerdomain LOWER($\mathbb{N}_\perp$) of $\mathbb{N}_\perp$ is isomorphic to the powerset $\mathcal{P}(\mathbb{N})$ of $\mathbb{N}$ ordered by inclusion. The lower powerdomain LOWER($\mathbb{B}_\perp$) of $\mathbb{B}_\perp$ is isomorphic to the powerset $\mathcal{P}(\mathbb{B})$ of $\mathbb{B}$ ordered by inclusion. The lower powerdomain LOWER($\mathbb{C}(\mathbb{R}^\sigma)$) of $\mathbb{C}(\mathbb{R}^\sigma)$ is a continuous Scott domain with a basis consisting of the set $\mathcal{P}_f\left(\mathbb{I}^{\mathrm{C}}\mathbb{F}\right)$ of finite subsets of the set $\mathbb{I}^{\mathrm{C}}\mathbb{F}$ of closed intervals with end points in the dense subset $\mathbb{F}$ of $\mathbb{R}^\sigma$.

The general formula giving the interpretation $(\!|H \rhd P : t|\!)\mu$ of an LAR term P without the parallel construct in the parallel model in terms of its interpretation $[\![H \rhd P : t]\!]\rho$ in the standard model is given by

$$(\!|H \rhd P : t|\!)\mu = \bigsqcup_{\rho:\forall x\in\mathsf{variable}\cdot\widehat{\rho(x)}\sqsubseteq\mu(x)} [\![H \widehat{\rhd P :} t]\!]\rho.$$

In particular, for any closed LAR term P without the parallel construct

$$(\!|P|\!) = \widehat{[\![P]\!]}.$$

The interpretation of the remaining new construct $\{P, P\}$ in the parallel model is given by

$$(\!|\,\{P, Q\}\,|\!) = (\!|P|\!) \sqcup (\!|Q|\!).$$

13.3.4 Computational Adequacy

LAR is a deterministic extension of PCF with meaningful programs that never terminate. However, Plotkin's notion of computability assumes that non-terminating programs are meaningless. Escardó [23] circumvented this problem by generalizing Plotkin's notion of computability as described in section 13.2.4. This notion can be specialized by considering chains instead of directed sets. In particular, a program P is computable whenever

$$[\![P]\!] \sqsubseteq \mathsf{eval}(P)$$

where

$$\mathsf{eval}(P) = \bigsqcup_{n=0}^{\infty} \mathsf{step}(P, n)$$

and

$$\begin{array}{rcl}
\mathsf{step}(P, n) & = & \mathsf{value}(\mathsf{reduce}(P, n)) \\
\mathsf{value}(P) & = & \left\{\begin{array}{ll} n & \text{if } P = \mathsf{succ}^n(0) \\ \mathsf{true} & \text{if } P = \mathsf{true} \\ \mathsf{false} & \text{if } P = \mathsf{false} \\ \perp & \text{otherwise} \end{array}\right. \\
\mathsf{reduce}(P, n) & = & \left\{\begin{array}{ll} \mathsf{reduce}(Q, n-1) & \text{if } n > 0 \text{ and } P \to Q \\ P & \text{otherwise.} \end{array}\right.
\end{array}$$

The evaluation function eval is well defined provided that the one-step reduction relation $\rightarrow$ is *sound*

$$\frac{P \rightarrow Q}{\mathsf{value}\,(P) \sqsubseteq \mathsf{value}\,(Q)}$$

and *weakly Church-Rosser*

$$\frac{P \rightarrow Q \quad P \rightarrow R}{\exists S \cdot Q \rightarrow^{*} S \quad R \rightarrow^{*} S}.$$

For LAR, the one-step reduction relation is weakly Church-Rosser because the reduction rules are deterministic.

$\boxed{\textbf{LAR without the parallel construct in the standard model}}$

For LAR without the parallel construct in the standard model, the value function value is extended to

$$\mathsf{value}\,(P) = \begin{cases} n & \text{if } P = \mathsf{succ}^n\,(0) \\ \mathsf{true} & \text{if } P = \mathsf{true} \\ \mathsf{false} & \text{if } P = \mathsf{false} \\ \mathsf{info}\,(V) & \text{if } P = [V\rangle \\ \mathsf{info}\,(M) & \text{if } P = [M\rangle\, Q \\ \mathsf{info}\,(V^{+}) & \text{if } P = \langle V^{+}\rangle \\ \mathsf{info}\,(M^{+}) & \text{if } P = \langle M^{+}\rangle\, Q \\ (\mathsf{value}\,(Q)\,, \mathsf{value}\,(R)) & \text{if } P = (Q, R) \\ \bot & \text{otherwise.} \end{cases}$$

Lemma 86 *The one-step reduction relation for LAR without the parallel construct in the standard model is sound*

$$\frac{P \rightarrow Q}{\mathsf{value}\,(P) \sqsubseteq \mathsf{value}\,(Q)}.$$

The last lemma and the determinism of the reduction rules ensure that the evaluation function eval is well defined. In order to establish soundness, namely

$$\mathsf{eval}\,(P) \sqsubseteq [\![P]\!],$$

the following two lemmas are required.

Lemma 87 *If P is an LAR program without the parallel construct then $P \rightarrow Q$ implies $[\![P]\!] = [\![Q]\!]$.*

Lemma 88 *If P is an LAR program without the parallel construct then* $\mathsf{value}(P) \sqsubseteq [\![P]\!]$.

Proposition 89 (Soundness) *If P is an LAR program without the parallel construct then* $\mathsf{eval}(P) \sqsubseteq [\![P]\!]$.

In order to establish completeness of LAR without the parallel construct, namely

$$[\![P]\!] \sqsubseteq \mathsf{eval}(P),$$

the following two lemmas are required.

Lemma 90 *An LAR program P without the parallel construct is* computable *iff*

$$\forall a \ll [\![P]\!] \cdot \exists n \in \mathbb{N} \cdot a \sqsubseteq \mathsf{step}(P, n).$$

Lemma 91 *For any closed LAR term $P : s \to t$ without the parallel construct, where s and t are ground types, if*

$$\forall n \in \mathbb{N} \cdot \exists m \in \mathbb{N} \cdot [\![P]\!] (\mathsf{step}(Q, n)) \sqsubseteq \mathsf{step}(PQ, m)$$

for every closed computable term $Q : s$ then $P : s \to t$ is computable.

So, we need to prove that every LAR term without the parallel construct is computable, establishing theorem 113 below, by extending the inductive proof of Plotkin [56] with the following lemmas.

Lemma 92 *The PCF constructs are computable.*

Lemma 93 *The pairing construct is computable.*

Lemma 94 *The projection constructs are computable.*

Lemma 95 *The squeeze construct is computable.*

Lemma 96

$$\begin{aligned}
\mathsf{reduce}([M]\,(\mathsf{reduce}(P,1)),1) &= \mathsf{reduce}([M]\,P,2)\\
\mathsf{reduce}([M\rangle\,(\mathsf{reduce}(P,1)),1) &= \mathsf{reduce}([M\rangle\,P,2)\\
\mathsf{reduce}(\langle M\rangle\,(\mathsf{reduce}(P,1)),1) &= \mathsf{reduce}(\langle M\rangle\,P,2).
\end{aligned}$$

Lemma 97

$$\begin{aligned}
\forall n &\in \mathbb{N} \cdot \mathsf{reduce}([M]\,(\mathsf{reduce}(P,n)),1) = \mathsf{reduce}([M]\,P,n+1)\\
\forall n &\in \mathbb{N} \cdot \mathsf{reduce}([M\rangle\,(\mathsf{reduce}(P,n)),1) = \mathsf{reduce}([M\rangle\,P,n+1)\\
\forall n &\in \mathbb{N} \cdot \mathsf{reduce}(\langle M\rangle\,(\mathsf{reduce}(P,n)),1) = \mathsf{reduce}(\langle M\rangle\,P,n+1).
\end{aligned}$$

Lemma 98

$$\begin{array}{rcl} [\![[M]]\!]\,(\mathsf{value}\,(P)) & = & \mathsf{step}\,([M]\,P, 1) \\ [\![[M\rangle]\!]\,(\mathsf{value}\,(P)) & = & \mathsf{step}\,([M\rangle\,P, 1) \\ [\![\langle M\rangle]\!]\,(\mathsf{value}\,(P)) & = & \mathsf{step}\,(\langle M\rangle\,P, 1)\,. \end{array}$$

Lemma 99

$$\begin{array}{rcl} \forall n & \in & \mathbb{N}\cdot [\![[M]]\!]\,(\mathsf{step}\,(P,n)) = \mathsf{step}\,([M]\,P, n+1) \\ \forall n & \in & \mathbb{N}\cdot [\![[M\rangle]\!]\,(\mathsf{step}\,(P,n)) = \mathsf{step}\,([M\rangle\,P, n+1) \\ \forall n & \in & \mathbb{N}\cdot [\![\langle M\rangle]\!]\,(\mathsf{step}\,(P,n)) = \mathsf{step}\,(\langle M\rangle\,P, n+1)\,. \end{array}$$

Lemma 100

$$\begin{array}{rcl} \forall n & \in & \mathbb{N}\cdot\exists m\in\mathbb{N}\cdot\mathsf{step}\,([T]\,(\mathsf{reduce}\,(P,1))\,,n)\sqsubseteq\mathsf{step}\,([T]\,P,m) \\ \forall n & \in & \mathbb{N}\cdot\exists m\in\mathbb{N}\cdot\mathsf{step}\,([T\rangle\,(\mathsf{reduce}\,(P,1))\,,n)\sqsubseteq\mathsf{step}\,([T\rangle\,P,m) \\ \forall n & \in & \mathbb{N}\cdot\exists m\in\mathbb{N}\cdot\mathsf{step}\,(\langle T\rangle\,(\mathsf{reduce}\,(P,1))\,,n)\sqsubseteq\mathsf{step}\,(\langle T\rangle\,P,m)\,. \end{array}$$

Lemma 101

$$\begin{array}{rcl} \forall n,m & \in & \mathbb{N}\cdot\exists r\in\mathbb{N}\cdot\mathsf{step}\,([T]\,(\mathsf{reduce}\,(P,n))\,,m)\sqsubseteq\mathsf{step}\,([T]\,P,r) \\ \forall n,m & \in & \mathbb{N}\cdot\exists r\in\mathbb{N}\cdot\mathsf{step}\,\big([T\rangle'\,(\mathsf{reduce}\,(P,n))\,,m\big)\sqsubseteq\mathsf{step}\,\big([T\rangle'\,P,r\big) \\ \forall n,m & \in & \mathbb{N}\cdot\exists r\in\mathbb{N}\cdot\mathsf{step}\,\big(\langle T\rangle'\,(\mathsf{reduce}\,(P,n))\,,m\big)\sqsubseteq\mathsf{step}\,\big(\langle T\rangle'\,P,r\big)\,. \end{array}$$

Lemma 102

$$\begin{array}{rcl} \exists n & \in & \mathbb{N}\cdot [\![[T]]\!]\,(\mathsf{value}\,(P)) = \mathsf{step}\,([T]\,P,n) \\ \exists n & \in & \mathbb{N}\cdot [\![[T\rangle']\!]\,(\mathsf{value}\,(P)) = \mathsf{step}\,\big([T\rangle'\,P,n\big) \\ \exists n & \in & \mathbb{N}\cdot [\![\langle T\rangle']\!]\,(\mathsf{value}\,(P)) = \mathsf{step}\,\big(\langle T\rangle'\,P,n\big)\,. \end{array}$$

Lemma 103

$$\begin{array}{rcl} \forall n & \in & \mathbb{N}\cdot\exists m\in\mathbb{N}\cdot [\![[T]]\!]\,(\mathsf{step}\,(P,n))\sqsubseteq\mathsf{step}\,([T]\,P,m) \\ \forall n & \in & \mathbb{N}\cdot\exists m\in\mathbb{N}\cdot [\![[T\rangle']\!]\,(\mathsf{step}\,(P,n))\sqsubseteq\mathsf{step}\,\big([T\rangle'\,P,m\big) \\ \forall n & \in & \mathbb{N}\cdot\exists m\in\mathbb{N}\cdot [\![\langle T\rangle']\!]\,(\mathsf{step}\,(P,n))\sqsubseteq\mathsf{step}\,\big(\langle T\rangle'\,P,m\big)\,. \end{array}$$

Lemma 104 *The transformation constructs are computable.*

Lemma 105 *Every term P of LAR without the parallel construct is computable.*

Theorem 106 (Completeness) *If P is an LAR program without the parallel construct then*

$$[\![P]\!] \sqsubseteq \mathsf{eval}\,(P)\,.$$

LAR without the parallel construct in the parallel model

Let

$$\mathsf{pvalue}\,(P) = \downarrow \{\mathsf{value}\,(P)\} \tag{13.16}$$

and let

$$\mathsf{pstep} = \mathsf{pvalue} \circ \mathsf{reduce}.$$

Lemma 107 *The one-step reduction relation for LAR without the parallel construct in the parallel model is sound*

$$\frac{P \to Q}{\mathsf{pvalue}\,(P) \sqsubseteq \mathsf{pvalue}\,(Q)}.$$

Therefore

$$\mathsf{peval}\,(P) = \bigsqcup_{n=0}^{\infty} \mathsf{pstep}\,(P, n)$$

is well defined. But

$$\begin{aligned}
\mathsf{peval}\,(P) &= \bigsqcup_{n=0}^{\infty} \mathsf{pstep}\,(P, n) \\
&= \bigsqcup_{n=0}^{\infty} \downarrow \{\mathsf{step}\,(P, n)\} \\
&= \downarrow \left\{ \bigsqcup_{n=0}^{\infty} \mathsf{step}\,(P, n) \right\} \\
&= \downarrow \{\mathsf{eval}\,(P)\} \\
&= \downarrow \{[\![P]\!]\} \\
&= (\!|P|\!)
\end{aligned}$$

and so LAR without the parallel construct is computationally adequate in the parallel model.

LAR in the parallel model

Extend the definition of pvalue in equation (13.16) with

$$\mathsf{pvalue}\,(\{P, Q\}) = \mathsf{pvalue}\,(P) \sqcup \mathsf{pvalue}\,(Q)\,.$$

Lemma 108 *The one-step reduction relation for LAR in the parallel model is sound*

$$\frac{P \to Q}{\mathsf{pvalue}\,(P) \sqsubseteq \mathsf{pvalue}\,(Q)}.$$

Therefore

$$\mathsf{peval}\,(P) = \bigsqcup_{n=0}^{\infty} \mathsf{pstep}\,(P, n)$$

is still well defined. In order to establish soundness, namely

$$\mathsf{peval}\,(P) \sqsubseteq (\!|P|\!),$$

the following two lemmas are required.

Lemma 109 *If P is an LAR program then $P \to Q$ implies $(\!|P|\!) = (\!|Q|\!)$.*

Lemma 110 *If P is an LAR program then* $\mathsf{pvalue}\,(P) \sqsubseteq (\!|P|\!)$.

Proposition 111 (Soundness) *If P is an LAR program then* $\mathsf{peval}\,(P) \sqsubseteq (\!|P|\!)$.

In order to establish *completeness*, namely

$$(\!|P|\!) \sqsubseteq \mathsf{peval}\,(P)\,,$$

we consider Plotkin's notion of computability [57] as extended by Escardó [23]. So, we need to prove that every term is computable, establishing proposition 113 below, by extending the inductive proof of Plotkin [56] with the following lemma.

Lemma 112 *The parallel construct is computable.*

Theorem 113 (Completeness) *If P is an LAR program then*

$$(\!|P|\!) \sqsubseteq \mathsf{peval}\,(P)\,.$$

Finally, proposition 111 and theorem 113 together show that LAR is computationally adequate, namely

$$(\!|P|\!) = \mathsf{peval}\,(P)\,.$$

Computational adequacy verifies that the incorporated fair strategy for tensor absorption as described in section 11.9 is correct. LAR allows a very wide range of mathematical functions to be defined elegantly from the rich theory of continued fractions.

13.3.5 Worked Examples

Example 114 *Consider the LAR program* $P : \mathsf{num}$

$$\begin{aligned} P &= \mu x \cdot Q \\ Q &= \{0, \mathsf{succ}\,(x)\} \end{aligned}$$

$$\begin{aligned}
\mathsf{pstep}\,(P,0) &= \mathsf{pvalue}\,(P) = \{\bot\} \\
\mathsf{pstep}\,(P,1) &= \mathsf{pvalue}\,(\{0, \mathsf{succ}\,(P)\}) = \{\bot, 0\} \\
\mathsf{pstep}\,(P,2) &= \mathsf{pvalue}\,(\{0, \mathsf{succ}\,(\{0, \mathsf{succ}\,(P)\})\}) = \{\bot, 0\} \\
\mathsf{pstep}\,(P,3) &= \mathsf{pvalue}\,\left(\left\{0, \left\{\mathsf{succ}\,(0), \mathsf{succ}^2\,(P)\right\}\right\}\right) = \{\bot, 0, 1\} \\
\mathsf{pstep}\,(P,4) &= \mathsf{pvalue}\,\left(\left\{0, \left\{\mathsf{succ}\,(0), \mathsf{succ}^2\,(\{0, \mathsf{succ}\,(P)\})\right\}\right\}\right) = \{\bot, 0, 1\} \\
\mathsf{pstep}\,(P,5) &= \mathsf{pvalue}\,\left(\left\{0, \left\{\mathsf{succ}\,(0), \mathsf{succ}\,\left(\left\{\mathsf{succ}\,(0), \mathsf{succ}^2\,(P)\right\}\right)\right\}\right\}\right) \\
&= \{\bot, 0, 1\} \\
\mathsf{pstep}\,(P,6) &= \mathsf{pvalue}\,\left(\left\{0, \left\{\mathsf{succ}\,(0), \left\{\mathsf{succ}^2\,(0), \mathsf{succ}^3\,(P)\right\}\right\}\right\}\right) \\
&= \{\bot, 0, 1, 2\} \\
\mathsf{pstep}\,(P,n) &= \{\bot\} \cup \mathbb{N}\left(\left\lfloor \frac{-1+\sqrt{8n+1}}{2} \right\rfloor\right) \\
\mathsf{peval}\,(P) &= \mathbb{N}_\bot
\end{aligned}$$

$$\begin{aligned}
(\!|P|\!) &= \mathsf{fix}\,(D \mapsto (\!|x : \mathsf{num} \rhd Q : \mathsf{num}|\!)\,[x \mapsto D]) \\
&= \mathsf{fix}\left(D \mapsto \begin{array}{l} (\!|x : \mathsf{num} \rhd 0 : \mathsf{num}|\!)\,[x \mapsto D] \sqcup \\ (\!|x : \mathsf{num} \rhd \mathsf{succ}\,(x) : \mathsf{num}|\!)\,[x \mapsto D] \end{array}\right) \\
&= \mathsf{fix}\left(D \mapsto \{\bot, 0\} \sqcup \bigsqcup_{d \in D} \downarrow \{[\![x : \mathsf{num} \rhd \mathsf{succ}\,(x) : \mathsf{num}]\!]\,[x \mapsto d]\}\right)
\end{aligned}$$

$$\begin{aligned}
[\![x &: \mathsf{num} \rhd \mathsf{succ}\,(x) : \mathsf{num}]\!]\rho = \begin{cases} [\![\Lambda]\!]\rho + 1 & \text{\textit{if} } [\![\Lambda]\!]\rho \neq \bot \\ \bot & \text{\textit{if} } [\![\Lambda]\!]\rho = \bot \end{cases} \\
&= \begin{cases} \rho\,(x) + 1 & \text{\textit{if} } \rho\,(x) \neq \bot \\ \bot & \text{\textit{if} } \rho\,(x) = \bot \end{cases} \\
&= \begin{cases} d + 1 & \text{\textit{if} } d \neq \bot \\ \bot & \text{\textit{if} } d = \bot \end{cases} \\
\textit{where } \Lambda &= x : \mathsf{num} \rhd x : \mathsf{num} \\
\textit{and } \rho &= [x \mapsto d]\,.
\end{aligned}$$

So, we have shown explicitly that

$$(\!|\mu x \cdot \{0, \mathsf{succ}\,(x)\}\,|\!) = \mathsf{eval}\,(\mu x \cdot \{0, \mathsf{succ}\,(x)\}) = \mathbb{N}_\bot.$$

Example 115 *Consider the LAR term $P : \mathsf{real}^+ \to \mathsf{real}^+$*

$$\begin{aligned} P &= \mu x \cdot Q \\ Q &= \{R_{1-b}, \ldots, R_{-1}, R_0, R_1, \ldots, R_{b-1}\} \\ R_i &= \left\langle {}^b D_i \right\rangle \circ x \lhd \left[{}^b D_i^\dagger \right\rangle \end{aligned}$$

where $\circ$ denotes function composition. This interesting example corresponds to the normalization of a general normal product to exact floating point. Let

$$\begin{aligned} t &= \mathsf{real}^+ \to \mathsf{real}^+ \\ g &= [\![x : t \rhd \left\langle {}^b D_i \right\rangle \circ x : t]\!] [x \mapsto f] \\ &= {}^b D_i \circ f \\ h &= [\![x : t \rhd \left[{}^b D_i^\dagger \right\rangle : \mathsf{real}^+ \to \mathsf{real}^\infty]\!] [x \mapsto f] \\ &= {}^b D_i^\dagger \end{aligned}$$

and so

$$(\!| P |\!) = \mathsf{fix}\left(F \mapsto (\!| x : t \rhd Q : t |\!) [x \mapsto F]\right)$$

$$(\!| x : t \rhd Q : t |\!) [x \mapsto F] = \bigsqcup_{i \in \mathbb{Z}(b)} (\!| x : t \rhd R_i : t |\!) [x \mapsto F]$$

$$(\!| x : t \rhd R_i : t |\!) [x \mapsto F] =$$

$$D \mapsto \bigsqcup_{\substack{f : \forall e \in C\left(\mathbb{R}^+\right) \cdot f(e) \in F(\downarrow\{e\}) \\ d \in D}} \downarrow \{ [\![x : t \rhd R_i : t]\!] [x \mapsto f] (d) \}$$

$$[\![x : t \rhd R_i : t]\!] [x \mapsto f] = d \mapsto \begin{cases} g(d) & \textit{if } h(d) \subseteq (0, \infty) \\ \bot & \textit{otherwise.} \end{cases}$$

Let

$$\begin{aligned} \theta(f, i) &= d \mapsto \begin{cases} {}^b D_i (f(d)) & \textit{if } \mathsf{info}\left({}^b D_i\right) \ll d \\ [0, \infty] & \textit{otherwise} \end{cases} \\ \mathcal{F}(F) &= D \mapsto \bigsqcup_{\substack{f : \forall e \in C\left(\mathbb{R}^+\right) \cdot f(e) \in F(\downarrow\{e\}) \\ i \in \mathbb{Z}(b) \\ d \in D}} \downarrow \{ \theta(f, i)(d) \} \end{aligned}$$

and so

$$[\![x : t \rhd R_i : t]\!] [x \mapsto f] = \theta(f, i)$$

$$(\!| x : t \rhd R_i : t |\!) [x \mapsto F] =$$

$$D \mapsto \bigsqcup_{\substack{f:\forall e\in C\left(\mathbb{R}^+\right)\cdot f(e)\in F(\downarrow\{e\}) \\ d\in D}} \downarrow\{\theta(f,i)(d)\}$$

$$(\!| x : t \rhd Q : t |\!)\,[x \mapsto F] = \mathcal{F}(F)$$

$$(\!|P|\!) = \bigsqcup_{n=0}^{\infty} \mathcal{F}^n\left(D \mapsto \{[0,\infty]\}\right).$$

But

$$\begin{aligned}
\mathcal{F}\left(D \mapsto \{[0,\infty]\}\right) &= D \mapsto \bigsqcup_{\substack{i\in\mathbb{Z}(2)\\ d\in D}} \downarrow\{\theta(e\mapsto[0,\infty],i)(d)\} \\
\theta(e\mapsto[0,\infty],i) &= d \mapsto \begin{cases} \textsf{info}\left({}^bD_i\right) & \textit{if } \textsf{info}\left({}^bD_i\right) \ll d \\ [0,\infty] & \textit{otherwise} \end{cases} \\
\mathcal{F}\left(D \mapsto \{[0,\infty]\}\right) &= D \mapsto \bigsqcup_{\substack{i\in\mathbb{Z}(b)\\ d\in D}} \begin{cases} \downarrow\left\{\textsf{info}\left({}^bD_i\right)\right\} & \textit{if } \textsf{info}\left({}^bD_i\right) \ll d \\ \{[0,\infty]\} & \textit{otherwise} \end{cases} \\
\mathcal{F}^2\left(D \mapsto \{[0,\infty]\}\right) &= \\
D &\mapsto \bigsqcup_{\substack{i\in\mathbb{Z}(b)\\ j\in\mathbb{Z}(b)\\ d\in D}} \begin{cases} \downarrow\left\{\textsf{info}\left({}^bD_i \bullet {}^bD_j\right)\right\} & \textit{if } \textsf{info}\left({}^bD_i \bullet {}^bD_j\right) \ll d \\ \downarrow\left\{\textsf{info}\left({}^bD_i\right)\right\} & \textit{if } \textsf{info}\left({}^bD_i\right) \ll d \\ \{[0,\infty]\} & \textit{otherwise} \end{cases} \\
\mathcal{F}^n\left(D \mapsto \{[0,\infty]\}\right) &= D \mapsto \bigsqcup_{\substack{m\in\mathbb{N}(n+1)\\ c\in\mathbb{Z}(b^m)\\ d\in D}} \downarrow\left\{\textsf{info}\left({}^b\mathfrak{D}^n_c\right) \middle| \textsf{info}\left({}^b\mathfrak{D}^n_c\right) \ll d\right\}
\end{aligned}$$

where ${}^b\mathfrak{D}^n_c$ *is defined in equation (9.3). Therefore*

$$(\!|P|\!) = D \mapsto \bigsqcup_{\substack{m\in\mathbb{N}\\ c\in\mathbb{Z}(b^m)\\ d\in D}} \downarrow\left\{\textsf{info}\left({}^b\mathfrak{D}^n_c\right) \middle| \textsf{info}\left({}^b\mathfrak{D}^n_c\right) \ll d\right\}$$

and in particular, for all $x \in (0,\infty)$

$$(\!|P|\!)\left(\downarrow\{x\}\right) = \downarrow\{x\}.$$

In other words, the program P *effectively converts an arbitrary unsigned real number into the unbiased exact floating point representation as defined in section 9.1 (i.e. normalization).*

Example 116 *A continued fraction by Lambert [44] for tangent is*

$$\mathsf{tan}\,(x) = \cfrac{1}{\frac{1}{x} + \cfrac{1}{-\frac{3}{x} + \cfrac{1}{\frac{5}{x} + \cfrac{1}{-\frac{7}{x} + \cdots}}}}$$

This can be transformed to

$$\begin{aligned}\mathsf{tan}\left(\frac{y-1}{y+1}\right) &= \begin{pmatrix} y-1 & y-1 \\ 2y & 2 \end{pmatrix} \\ &\prod_{n=0}^{\infty} \begin{pmatrix} (2n+3)(y+1) & (2n+1)y+(2n+5) \\ (2n+5)y+(2n+1) & (2n+3)(y+1) \end{pmatrix}\end{aligned}$$

or put another way

$$\begin{aligned}&\mathsf{tan}\left(\left(\begin{matrix} 1 & -1 \\ 1 & 1 \end{matrix}\right) y\right) \\ = &\left(\begin{matrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{matrix}\right)\left(y, \left(\begin{matrix} 3 & 1 & 3 & 5 \\ 5 & 3 & 1 & 3 \end{matrix}\right)\left(y, \left(\begin{matrix} 5 & 3 & 5 & 7 \\ 7 & 5 & 3 & 5 \end{matrix}\right)(y, \cdots)\right)\right)\end{aligned}$$

for $y \in [0, \infty]$. *This caters for* $\mathsf{tan}\,(x)$ *in the range* $x \in [-1, 1]$. *Thus, a restricted program for* tan *in LAR is*

$$\begin{aligned}\mathsf{tan} &= P \lhd \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \\ P &= \lambda x \cdot \left[\begin{matrix} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{matrix}\right\rangle (x, (\mu f.\lambda n.Q\,(x, f\,(n+1)))\,0) \\ Q &= \left\langle \begin{matrix} 2n+3 & 2n+1 & 2n+3 & 2n+5 \\ 2n+5 & 2n+3 & 2n+1 & 2n+3 \end{matrix} \right\rangle\end{aligned}$$

where for all $x \in \mathbb{R}^{\infty}$

$$[\![\mathsf{tan}]\!]\,(x) = \begin{cases} \mathsf{tan}\,(x) & \textit{if } x \in (-1,1) \\ \perp & \textit{otherwise.} \end{cases}$$

For $x \in [1, -1]$, *we must repeatedly apply the trigonometric identity*

$$\mathsf{tan}\,(x) = \frac{2\,\mathsf{tan}\left(\frac{x}{2}\right)}{1 - \mathsf{tan}\left(\frac{x}{2}\right)^2} = [T]\left(\mathsf{tan}\left(\frac{x}{2}\right), \mathsf{tan}\left(\frac{x}{2}\right)\right)$$

where

$$T = \left(\begin{array}{cccc} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{array} \right)$$

until $x \in (-1, 1)$. *A recursive application of the parallel and squeeze construct can be used to accomplish this provided that we use an interval way below* $[1, -1]$*; for instance* $\left[\frac{1}{2}, -\frac{1}{2}\right]$. *So, bringing all this together, we have*

$$\begin{aligned}
\mathsf{tan} &= \mu g \cdot \left\{ P \lhd \left[\begin{array}{cc} 1 & 1 \\ -1 & 1 \end{array} \right], R \lhd \left[\begin{array}{cc} 2 & -1 \\ 2 & 1 \end{array} \right] \right\} \\
P &= \lambda x \cdot \left[\begin{array}{cccc} 1 & 1 & -1 & -1 \\ 2 & 0 & 0 & 2 \end{array} \right\rangle (x, (\mu f \cdot \lambda n \cdot Q (x, f (n+1))) \, 0) \\
Q &= \left\langle \begin{array}{cccc} 2n+3 & 2n+1 & 2n+3 & 2n+5 \\ 2n+5 & 2n+3 & 2n+1 & 2n+3 \end{array} \right\rangle \\
R &= \lambda x \cdot \left[\begin{array}{cccc} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{array} \right] \left(g \left(\left[\begin{array}{cc} 1 & 1 \\ -4 & 4 \end{array} \right\rangle x \right), g \left(\left[\begin{array}{cc} 1 & 1 \\ -4 & 4 \end{array} \right\rangle x \right) \right)
\end{aligned}$$

where for all $x \in \mathbb{R}^\infty$

$$(\!|\mathsf{tan}|\!) (\downarrow \{x\}) = \downarrow \{\mathsf{tan}\,(x)\}\,.$$

13.3.6 Redundant If Operator

The redundant if operator is defined in equation (7.7) by

$$\begin{aligned}
\mathsf{rif} &: \mathbb{I}^{\mathrm{C}} K \times \mathbb{I}^{\mathrm{C}} \mathbb{F}^2 \times \left(\mathbb{I}^{\mathrm{C}} K \to t\right)^2 \to t \\
\mathsf{rif}\, x &< (I, J) \;\mathsf{then}\, f \;\mathsf{else}\, g = \left\{ \begin{array}{ll} f(x) & \text{if } I \ll x \\ g(x) & \text{if } J \ll x \end{array} \right.
\end{aligned}$$

where $K \in \mathbb{I}^{\mathrm{C}} \mathbb{R}^\infty$ and $\mathbb{F}$ is a dense subset of K. In practice, this definition is not useful because f and g are usually only defined over I and J respectively. This means that we need to insert some domain restricting functions between f and it's argument x and g and it's argument x. In terms of the language for all reals, this gives rise to two flavors of the redundant if operator; namely

$$\begin{aligned}
\mathsf{srif} &: \mathsf{real}^\infty \times \left(\mathsf{matrix}^\infty\right)^2 \times \left(\mathsf{real}^+ \to t\right)^2 \to t \\
\mathsf{srif}\,(x, M, N, f, g) &= \left\{ f \lhd \left[M^\dagger\right] x, g \lhd \left[N^\dagger\right] x \right\}
\end{aligned}$$

and

$$\begin{aligned}
\mathsf{urif} &: \mathsf{real}^+ \times \left(\mathsf{matrix}^+\right)^2 \times \left(\mathsf{real}^+ \to t\right)^2 \to t \\
\mathsf{urif}\,(x, M, N, f, g) &= \left\{ f \lhd \left[M^\dagger\right\rangle x, g \lhd \left[N^\dagger\right\rangle x \right\} \\
&=
\end{aligned}$$

where M and N must be non-singular. The interpretations of srif and urif are

$$\begin{aligned}
& (\!|\mathsf{srif}|\!)\,(x, M, N, f, g) \\
= \ & (\!|\mathsf{urif}|\!)\,(x, M, N, f, g) \\
= \ & \begin{cases} f\left(M^{\dagger}(x)\right) \sqcup g\left(N^{\dagger}(x)\right) & \text{if } \mathsf{info}\,(M) \ll x \text{ and } \mathsf{info}\,(N) \ll x \\ f\left(M^{\dagger}(x)\right) & \text{if } \mathsf{info}\,(M) \ll x \text{ only} \\ g\left(N^{\dagger}(x)\right) & \text{if } \mathsf{info}\,(N) \ll x \text{ only} \\ \{\bot\} & \text{otherwise.} \end{cases}
\end{aligned}$$

The first side condition

$$\mathsf{interior}\,(I) \cup \mathsf{interior}\,(J) = \mathsf{interior}\,(K)$$

becomes

$$\mathsf{interior}\,(\mathsf{info}\,(M)) \cup \mathsf{interior}\,(\mathsf{info}\,(N)) = \mathbb{R}^{\infty}$$

and

$$\mathsf{interior}\,(\mathsf{info}\,(M)) \cup \mathsf{interior}\,(\mathsf{info}\,(N)) = (0, \infty)$$

respectively. The second side condition

$$f\,(\{x\}) = g\,(\{x\}) \ \text{ if } I \ll \{x\} \ \text{ and } J \ll \{x\}$$

becomes

$$f\left(M^{\dagger}(\downarrow\{x\})\right) = g\left(N^{\dagger}(\downarrow\{x\})\right) \ \text{ if } \mathsf{info}\,(M) \ll \{x\} \ \text{ and } \mathsf{info}\,(N) \ll \{x\}\,.$$

In practice, these two conditions ensure that one argument in the parallel construct can be discarded in finite time. In particular, the program $\mathsf{srif}\,(P, M, N, F, G)$, where $(\!|P|\!) = \downarrow\{x\}$ for some $x \in \mathbb{R}^{\infty}$, can be reduced with impunity in the following manner: If $\mathsf{info}\left(M^{\dagger}\right) \subseteq (0, \infty)$ then

$$\mathsf{srif}\,(P, M, N, F, G) \to F\left(\left\langle\left|M^{\dagger}\right|\right\rangle P\right)$$

else if $\mathsf{info}\left(N^{\dagger}\right) \subseteq (0, \infty)$ then

$$\mathsf{srif}\,(P, M, N, F, G) \to G\left(\left\langle\left|N^{\dagger}\right|\right\rangle P\right)$$

else

$$\mathsf{srif}\,(A^{\circ}, M, N, F, G) \to \mathsf{srif}\,(\mathsf{reduce}\,(A^{\circ}, 1)\,, M, N, F, G)$$

$$\frac{\mathsf{info}\left(M^{\dagger} \bullet V^{+}\right) \subseteq (0, \infty)}{\mathsf{srif}\,(\langle V^{+}\rangle\,, M, N, F, G) \to F\,(\langle|M^{\dagger} \bullet V^{+}|\rangle)}$$

$$\frac{\mathsf{info}\left(N^{\dagger} \bullet V^{+}\right) \subseteq (0, \infty)}{\mathsf{srif}\,(\langle V^{+}\rangle\,, M, N, F, G) \to G\,(\langle|N^{\dagger} \bullet V^{+}|\rangle)}$$

$$\mathsf{srif}\left(\left\langle O^{+}\right\rangle Q, M, N, F, G\right) \to \mathsf{srif}\left(Q, \left(O^{+}\right) \bullet M, \left(O^{+}\right) \bullet N, F, G\right).$$

Chapter 14

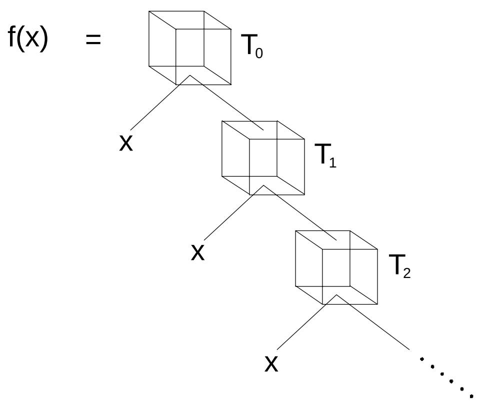
Conclusion

We started this thesis by noting that any real number can be represented by a sequence of nested closed intervals. We then put this idea into the formal context of an incremental digit representation. In particular, we presented the decimal, continued fraction, redundant binary, general normal product and exact floating point representations in this framework. We pointed out that any such sequence can be seen as a chain in the continuous domain of extended real numbers.

We then introduced the notion of an expression tree and linked it to the concept of a directed set in the continuous domain of extended real numbers. We then outlined a general two part procedure for converting any function with a power series representation into an expression tree. This general procedure was applied to the transcendental functions culminating in the introduction of new algorithms for π, e and the transcendental functions.

We presented the redundant if statement, which provides a simple and efficient means to overcome various computability problems during the construction of various expression trees. Straightforward reduction rules were presented to allow any valid expression tree to be converted incrementally into the exact floating point representation.

The digit set in the exact floating point representation has properties that enable the size of integers to be controlled and the flow of information to be analyzed in an expression tree. As a result, we introduced an efficient algorithm for converting an expression tree into the exact floating point representation. The quantization of information means that the complexity of these reduction rules is amenable to analysis. Although, both the spacial and temporal aspects of exact real arithmetic have been tackled in this thesis, for many applications the spacial overhead is still unavoidably prohibitive. This is essentially because, in general, the entire history of every variable must be remembered, not just the last value as in a conventional floating point application. Nevertheless, exact real arithmetic is still useful for many small applications where the user wants guaranteed correct answers, such as verification of algorithms. The efficiency of the exact real arithmetic presented in this thesis can undoubtedly be improved even further by hardware assisted software and parallel processes. The universality of the Language for All Reals is still an open question. In other words, can any computable real function be defined in this language? Another open question is whether the class of expression trees of the form



is the meromorphic functions.

Appendix A

Computational Adequacy Proofs

This appendix contains the proofs for each lemma and proposition in chapter 13.

Proof of Lemma 69:

"$\Rightarrow$". Assume $[\![P]\!] \sqsubseteq \mathsf{eval}\,(P)$. Suppose $[\![P]\!] = [\![X]\!]$. Note that $[\![X]\!] \neq \bot$. Therefore $[\![P]\!] = \mathsf{eval}\,(P)$. Note that $[\![X]\!] = \mathsf{eval}\,(X)$. Therefore $P \to^* X$. "$\Leftarrow$". Assume $[\![P]\!] = [\![X]\!]$ implies $P \to^* X$. Either $[\![P]\!] = \bot$ or $[\![P]\!] \in \mathbb{B} \cup \mathbb{N}$. Consider $[\![P]\!] = \bot$. Clearly $[\![P]\!] \sqsubseteq \mathsf{eval}\,(P)$. Consider $[\![P]\!] \in \mathbb{B} \cup \mathbb{N}$. Clearly $[\![P]\!] = [\![X]\!]$. Therefore $P \to^* X$. So, $\mathsf{eval}\,(P) = \mathsf{eval}\,(X)$. But $[\![X]\!] = \mathsf{eval}\,(X)$. Therefore $[\![P]\!] = \mathsf{eval}\,(P)$. ∎

For LPR, the value function value is extended to

$$\mathsf{value}\,(P) = \begin{cases} n & \text{if } P = \mathsf{succ}^n\,(0) \\ \mathsf{true} & \text{if } P = \mathsf{true} \\ \mathsf{false} & \text{if } P = \mathsf{false} \\ \mathsf{info}\,(V^+) & \text{if } P = \langle V^+ \rangle \\ \mathsf{info}\,(M^+) & \text{if } P = \langle M^+ \rangle\, R \\ \mathsf{value}\,(R) \times \mathsf{value}\,(S) & \text{if } P = (R, S) \\ \bot & \text{otherwise.} \end{cases}$$

Proof of Lemma 70:

The proof is by induction on the definition of $\to$. The rules $P \to Q$ with $\mathsf{value}\,(P) = \bot$ are trivial.

Basis:

$$\begin{aligned} \mathsf{value}\left(\langle M \rangle \left\langle V^+ \right\rangle\right) &= [\![\langle M \rangle]\!]\,([0,\infty]) \\ &\sqsubseteq [\![\left\langle M \bullet V^+ \right\rangle]\!] \\ &= \mathsf{value}\left(\left\langle M \bullet V^+ \right\rangle\right) \end{aligned}$$

$$\begin{aligned}\mathsf{value}\left(\langle M\rangle\left(\left\langle N^{+}\right\rangle P\right)\right) &= [\![\langle M\rangle]\!]\,([0,\infty]) \\ &\sqsubseteq [\![\left\langle M\bullet N^{+}\right\rangle]\!]\,([0,\infty]) \\ &= \mathsf{value}\left(\left\langle M\bullet N^{+}\right\rangle P\right).\end{aligned}$$

Inductive Step:

$$\frac{\mathsf{value}\,(P)\sqsubseteq\mathsf{value}\,(R)}{\mathsf{value}\,(P,Q)=(\mathsf{value}\,(P)\,,\mathsf{value}\,(Q))\sqsubseteq(\mathsf{value}\,(R)\,,\mathsf{value}\,(Q))=\mathsf{value}\,(R,Q)}$$

$$\frac{\mathsf{value}\,(Q)\sqsubseteq\mathsf{value}\,(S)}{\mathsf{value}\,(P,Q)=(\mathsf{value}\,(P)\,,\mathsf{value}\,(Q))\sqsubseteq(\mathsf{value}\,(P)\,,\mathsf{value}\,(S))=\mathsf{value}\,(P,S)}$$

$$\frac{\mathsf{value}\,(P)\sqsubseteq\mathsf{value}\,(Q)}{\mathsf{value}\,(\langle M\rangle\,P)=[\![\langle M\rangle]\!]\,([0,\infty])=\mathsf{value}\,(\langle M\rangle\,Q)}.\ \blacksquare$$

Proof of Lemma 71:

The proof is by induction on the definition of $\rightarrow$. The deterministic rules are trivial.

Basis: Trivial

Inductive Step:

- Pairing - If $P\rightarrow R$ and $Q\rightarrow S$ then

$$\begin{aligned}(P,Q) &\rightarrow (R,Q) \\ &\rightarrow (R,S) \\ (P,Q) &\rightarrow (P,S) \\ &\rightarrow (R,S)\,.\end{aligned}$$

- First projection - If $Q\rightarrow S$ then

$$\begin{aligned}\mathsf{fst}\,(P,Q) &\rightarrow \mathsf{fst}\,(P,S) \\ &\rightarrow P \\ \mathsf{fst}\,(P,Q) &\rightarrow P.\end{aligned}$$

- First projection - If $P\rightarrow R$ then

$$\begin{aligned}\mathsf{fst}\,(P,Q) &\rightarrow \mathsf{fst}\,(R,Q) \\ &\rightarrow R \\ \mathsf{fst}\,(P,Q) &\rightarrow P \\ &\rightarrow R.\end{aligned}$$

- Transformation - If $P \to Q$ then

$$\begin{aligned}
\langle M\rangle \left(\left\langle N^{+}\right\rangle P\right) &\to \left\langle M \bullet N^{+}\right\rangle P \\
&\to \left\langle M \bullet N^{+}\right\rangle Q \\
\langle M\rangle \left(\left\langle N^{+}\right\rangle P\right) &\to \langle M\rangle \left(\left\langle N^{+}\right\rangle Q\right) \\
&\to \left\langle M \bullet N^{+}\right\rangle Q.
\end{aligned}$$

- Transformation - If $P = \langle V^{+}\rangle$ then

$$\begin{aligned}
\langle M\rangle \left(\left\langle N^{+}\right\rangle P\right) &\to \left\langle M \bullet N^{+}\right\rangle P \\
&\to \left\langle M \bullet N^{+} \bullet V^{+}\right\rangle \\
\langle M\rangle \left(\left\langle N^{+}\right\rangle P\right) &\to \langle M\rangle \left\langle N^{+} \bullet V^{+}\right\rangle \\
&\to \left\langle M \bullet N^{+} \bullet V^{+}\right\rangle.
\end{aligned}$$

- Transformation - If $P = \langle O^{+}\rangle Q$ then

$$\begin{aligned}
\langle M\rangle \left(\left\langle N^{+}\right\rangle P\right) &\to \left\langle M \bullet N^{+}\right\rangle P \\
&\to \left\langle M \bullet N^{+} \bullet O^{+}\right\rangle Q \\
\langle M\rangle \left(\left\langle N^{+}\right\rangle P\right) &\to \langle M\rangle \left(\left\langle N^{+} \bullet O^{+}\right\rangle Q\right) \\
&\to \left\langle M \bullet N^{+} \bullet O^{+}\right\rangle Q.
\end{aligned}$$

- Transformation - If $P \to Q$ then

$$\begin{aligned}
\left\langle T^{+}\right\rangle P &\to \left\langle \left(T^{+}\right)^{\text{head}}\right\rangle \left(\left\langle \left(T^{+}\right)^{\text{tail}}\right\rangle P\right) \\
&\to \left\langle \left(T^{+}\right)^{\text{head}}\right\rangle \left(\left\langle \left(T^{+}\right)^{\text{tail}}\right\rangle Q\right) \\
\left\langle T^{+}\right\rangle P &\to \left\langle T^{+}\right\rangle Q \\
&\to \left\langle \left(T^{+}\right)^{\text{head}}\right\rangle \left(\left\langle \left(T^{+}\right)^{\text{tail}}\right\rangle Q\right).
\end{aligned}$$

- Transformation - If $Q \to S$ then

$$\begin{aligned}
\langle T\rangle \left(\left\langle V^{+}\right\rangle, Q\right) &\to \left\langle T \bullet_1 V^{+}\right\rangle Q \\
&\to \left\langle T \bullet_1 V^{+}\right\rangle S \\
\langle T\rangle \left(\left\langle V^{+}\right\rangle, Q\right) &\to \langle T\rangle \left(\left\langle V^{+}\right\rangle, S\right) \\
&\to \left\langle T \bullet_1 V^{+}\right\rangle S.
\end{aligned}$$

- Transformation - If $Q = \langle W^+ \rangle$ then

$$\begin{array}{rcl} \langle T \rangle \left(\left\langle V^+ \right\rangle, Q \right) & \rightarrow & \left\langle T \bullet_1 V^+ \right\rangle Q \\ & \rightarrow & \left\langle T \bullet_1 V^+ \bullet W^+ \right\rangle \\ \langle T \rangle \left(\left\langle V^+ \right\rangle, Q \right) & \rightarrow & \left\langle T \bullet_2 W^+ \right\rangle \left\langle V^+ \right\rangle \\ & \rightarrow & \left\langle T \bullet_2 W^+ \bullet V^+ \right\rangle. \end{array}$$

- Transformation - If $Q = \langle O^+ \rangle S$ then

$$\begin{array}{rcl} \langle T \rangle \left(\left\langle V^+ \right\rangle, Q \right) & \rightarrow & \left\langle T \bullet_1 V^+ \right\rangle Q \\ & \rightarrow & \left\langle T \bullet_1 V^+ \bullet O^+ \right\rangle S \\ \langle T \rangle \left(\left\langle V^+ \right\rangle, Q \right) & \rightarrow & \left\langle T \bullet_2 O^+ \right\rangle \left(\left\langle V^+ \right\rangle, S \right) \\ & \rightarrow & \left\langle T \bullet_2 O^+ \bullet_1 V^+ \right\rangle S. \end{array}$$

- Transformation - If $Q \rightarrow S$ then

$$\begin{array}{rcl} \langle T \rangle \left(\left\langle M^+ \right\rangle P, Q \right) & \rightarrow & \left\langle T \bullet_1 M^+ \right\rangle (P, Q) \\ & \rightarrow & \left\langle T \bullet_1 M^+ \right\rangle (P, S) \\ \langle T \rangle \left(\left\langle M^+ \right\rangle P, Q \right) & \rightarrow & \langle T \rangle \left(\left\langle M^+ \right\rangle P, S \right) \\ & \rightarrow & \left\langle T \bullet_1 M^+ \right\rangle (P, S). \end{array}$$

- Transformation - If $Q = \langle W^+ \rangle$ then

$$\begin{array}{rcl} \langle T \rangle \left(\left\langle M^+ \right\rangle P, Q \right) & \rightarrow & \left\langle T \bullet_1 M^+ \right\rangle (P, Q) \\ & \rightarrow & \left\langle T \bullet_1 M^+ \bullet_2 W^+ \right\rangle P \\ \langle T \rangle \left(\left\langle M^+ \right\rangle P, Q \right) & \rightarrow & \left\langle T \bullet_2 W^+ \right\rangle \left(\left\langle M^+ \right\rangle P \right) \\ & \rightarrow & \left\langle T \bullet_2 W^+ \bullet M^+ \right\rangle P. \end{array}$$

- Transformation - If $P = \langle O^+ \rangle S$ then

$$\begin{array}{rcl} \langle T \rangle \left(\left\langle M^+ \right\rangle P, Q \right) & \rightarrow & \left\langle T \bullet_1 M^+ \right\rangle (P, Q) \\ & \rightarrow & \left\langle T \bullet_1 M^+ \bullet_2 O^+ \right\rangle (P, S) \\ \langle T \rangle \left(\left\langle M^+ \right\rangle P, Q \right) & \rightarrow & \left\langle T \bullet_2 O^+ \right\rangle \left(\left\langle M^+ \right\rangle P, S \right) \\ & \rightarrow & \left\langle T \bullet_2 O^+ \bullet_1 M^+ \right\rangle (P, S). \ \blacksquare \end{array}$$

Proof of Lemma 72:
The proof is by induction on the definition of $\rightarrow$. Trivial. $\blacksquare$

Proof of Lemma 73:
The proof is by induction on the structure of P. Trivial. $\blacksquare$

Proof of Proposition 74:
Assume P is an LPR program. But $P \rightarrow Q$ implies $[\![P]\!] = [\![Q]\!]$. Therefore, $P \rightarrow^* Q$ implies $[\![P]\!] = [\![Q]\!]$. But $\mathsf{value}\,(P) \sqsubseteq [\![P]\!]$ and $\mathsf{value}\,(Q) \sqsubseteq [\![Q]\!]$. Therefore $P \rightarrow^* Q$ implies $\mathsf{value}\,(Q) \sqsubseteq [\![P]\!]$. Therefore $\bigsqcup \{\mathsf{value}\,(Q) | \, P \rightarrow^* Q\} \sqsubseteq [\![P]\!]$. But, $\mathsf{eval}\,(P) = \bigsqcup \{\mathsf{value}\,(Q) | \, P \rightarrow^* Q\}$. Therefore, $\mathsf{eval}\,(P) \sqsubseteq [\![P]\!]$. $\blacksquare$

Proof of Lemma 75:
"$\Rightarrow$". Assume P is computable. Therefore, $[\![P]\!] \sqsubseteq \mathsf{eval}\,(P)$. Consider $a \ll [\![P]\!]$. So, $a \ll \mathsf{eval}\,(P) = \bigsqcup \{\mathsf{value}\,(Q) | \, P \rightarrow^* Q\}$. Therefore, $\exists Q : P \rightarrow^* Q$ such that $a \sqsubseteq \mathsf{value}\,(Q)$. "$\Leftarrow$". Trivial. $\blacksquare$

Proof of Lemma 76:
Consider computable Q. Consider $a \ll [\![PQ]\!]$. But, $[\![PQ]\!] = [\![P]\!][\![Q]\!]$. By continuity of $[\![P]\!]$, $\exists b \ll [\![Q]\!]$ such that $a \ll [\![P]\!]b$. By computability of Q, $\exists R : Q \rightarrow^* R$ such that $b \sqsubseteq \mathsf{value}\,(R)$. But, $\exists S : PQ \rightarrow^* S$ such that $[\![P]\!]\,(\mathsf{value}\,(R)) \sqsubseteq \mathsf{value}\,(S)$. By monotonicity of $[\![P]\!]$, $[\![P]\!]b \sqsubseteq [\![P]\!]\,(\mathsf{value}\,(R))$. So, $a \ll [\![P]\!]b \sqsubseteq [\![P]\!]\,(\mathsf{value}\,(R)) \sqsubseteq \mathsf{value}\,(S)$. Therefore, $a \sqsubseteq \mathsf{value}\,(S)$. $\blacksquare$

Proof of Lemma 77:
Consider (P, Q). The function $x \mapsto y \mapsto (x, y)$ is clearly continuous. So, we only need to show that

$$\forall R : P \rightarrow^* R \cdot \forall S : Q \rightarrow^* S \cdot \exists D : (P, Q) \rightarrow^* D \cdot$$

$$(\mathsf{value}\,(R)\,, \mathsf{value}\,(S)) \sqsubseteq \mathsf{value}\,(D)$$

for all closed computable terms P and Q. Let P and Q be closed computable terms and consider $R : P \rightarrow^* R$ and $S : Q \rightarrow^* S$.

$$(P, Q) \rightarrow^* (P, S) \rightarrow^* (R, S)\,. \blacksquare$$

Proof of Lemma 78:

Consider $\mathsf{fst}(P)$. The function $x \mapsto \mathsf{fst}\,(x)$ is clearly continuous. So, we only need to show that

$$\forall Q : P \to^* Q \cdot \exists R : \mathsf{fst}\,(P) \to^* R \cdot \mathsf{fst}\,(\mathsf{value}\,(Q)) \sqsubseteq \mathsf{value}\,(R)$$

for any closed computable term P. Let P be a closed computable term and consider $Q : P \to^* Q$.

- Consider $\mathsf{value}\,(Q) = \bot$.
$$\mathsf{fst}\,(P) \to^* \mathsf{fst}\,(Q)\,.$$
- Consider $Q = (R, S)$.
$$\mathsf{fst}\,(P) \to^* \mathsf{fst}\,((R, S)) \to R.$$

Similarly for $\mathsf{snd}\,(P)$. ■

Proof of Lemma 79:
Consider $P \to^* Q$. Clearly $\langle M\rangle\, P \to^* \langle M\rangle\, Q$.

- Consider $\mathsf{value}\,(Q) = [0, \infty]$.
$$\langle M\rangle\, Q.$$
- Consider $Q = \langle V^+\rangle$.
$$\langle M\rangle\, Q = \langle M\rangle \left\langle V^+\right\rangle \to \left\langle M \bullet V^+\right\rangle.$$
- Consider $Q = \langle N^+\rangle\, S$.
$$\langle M\rangle\, Q = \langle M\rangle \left(\left\langle N^+\right\rangle S\right) \to \left\langle M \bullet N^+\right\rangle S. \blacksquare$$

Proof of Lemma 80:
Consider $P \to^* Q$. Clearly $\langle T\rangle\, P \to^* \langle T\rangle\, Q$.

- Consider $\mathsf{value}\,(Q) = \bot$ or $([0, \infty]\,, [0, \infty])$.
  - Consider $[\![T]\!] \in \mathbb{T}^+$.
$$\langle T\rangle\, Q \to \left\langle T^{\text{head}}\right\rangle \left(\left\langle T^{\text{tail}}\right\rangle Q\right).$$

- Consider $[\![T]\!] \notin \mathbb{T}^+$.

$$\langle T \rangle \, Q.$$

- Consider $Q = (A_1^\circ, \langle V_2 \rangle)$.

$$\begin{array}{rcl} \langle T \rangle \, Q & = & \langle T \rangle \, (A_1^\circ, \langle V_2 \rangle) \\ & \rightarrow & \langle T \bullet_2 V_2 \rangle \, A_1^\circ. \end{array}$$

- Consider $Q = (A_1^\circ, \langle N_2 \rangle \, S_2)$.

  - Consider $[\![T \bullet_2 N_2]\!] \in \mathbb{T}^+$.

$$\begin{array}{rcl} \langle T \rangle \, Q & = & \langle T \rangle \, (A_1^\circ, \langle N_2 \rangle \, S_2) \\ & \rightarrow & \langle T \bullet_2 N_2 \rangle \, (A_1^\circ, S_2) \\ & \rightarrow & \left\langle (T \bullet_2 N_2)^{\text{head}} \right\rangle \left(\left\langle (T \bullet_2 N_2)^{\text{tail}} \right\rangle (A_1^\circ, S_2) \right). \end{array}$$

  - Consider $[\![T \bullet_2 N_2]\!] \notin \mathbb{T}^+$.

$$\begin{array}{rcl} \langle T \rangle \, Q & = & \langle T \rangle \, (A_1^\circ, \langle N_2 \rangle \, S_2) \\ & \rightarrow & \langle T \bullet_2 N_2 \rangle \, (A_1^\circ, S_2). \end{array}$$

- Consider $Q = (\langle V_1 \rangle, A_2^\circ)$. Similar to $Q = (A_1^\circ, \langle V_2 \rangle)$.
- Consider $Q = (\langle V_1 \rangle, \langle V_2 \rangle)$.

$$\begin{array}{rcl} \langle T \rangle \, Q & = & \langle T \rangle \, (\langle V_1 \rangle, \langle V_2 \rangle) \\ & \rightarrow & \langle T \bullet_1 V_1 \rangle \, \langle V_2 \rangle \\ & \rightarrow & \langle T \bullet_1 V_1 \bullet V_2 \rangle. \end{array}$$

- Consider $Q = (\langle V_1 \rangle, \langle N_2 \rangle \, S_2)$.

$$\begin{array}{rcl} \langle T \rangle \, Q & = & \langle T \rangle \, (\langle V_1 \rangle, \langle N_2 \rangle \, S_2) \\ & \rightarrow & \langle T \bullet_1 V_1 \rangle \, (\langle N_2 \rangle \, S_2) \\ & \rightarrow & \langle T \bullet_1 V_1 \bullet N_2 \rangle \, S_2. \end{array}$$

- Consider $Q = (\langle N_1 \rangle \, S_1, A_2^\circ)$. Similar to $Q = (A_1^\circ, \langle N_2 \rangle \, S_2)$.
- Consider $Q = (\langle N_1 \rangle \, S_1, \langle V_2 \rangle)$. Similar to $Q = (\langle V_1 \rangle, \langle N_2 \rangle \, S_2)$.
- Consider $Q = (\langle N_1 \rangle \, S_1, \langle N_2 \rangle \, S_2)$.

– Consider $[\![T \bullet_1 N_1 \bullet N_2]\!] \in \mathbb{T}^+$.

$$\begin{aligned}\langle T\rangle\, Q &= \langle T\rangle\, (\langle N_1\rangle\, S_1, \langle N_2\rangle\, S_2)\\ &\rightarrow \langle T \bullet_1 N_1\rangle\, (S_1, \langle N_2\rangle\, S_2)\\ &\rightarrow \langle T \bullet_1 N_1 \bullet N_2\rangle\, (S_1, S_2)\\ &\rightarrow \left\langle (T \bullet_1 N_1 \bullet N_2)^{\mathrm{head}}\right\rangle \left(\left\langle (T \bullet_1 N_1 \bullet N_2)^{\mathrm{tail}}\right\rangle (S_1, S_2)\right).\end{aligned}$$

– Consider $[\![T \bullet_1 N_1 \bullet N_2]\!] \notin \mathbb{T}^+$.

$$\begin{aligned}\langle T\rangle\, Q &= \langle T\rangle\, (\langle N_1\rangle\, S_1, \langle N_2\rangle\, S_2)\\ &\rightarrow \langle T \bullet_1 N_1\rangle\, (S_1, \langle N_2\rangle\, S_2)\\ &\rightarrow \langle T \bullet_1 N_1 \bullet N_2\rangle\, (S_1, S_2)\,. \;\blacksquare\end{aligned}$$

Proof of Lemma 81:

- Consider $\langle P\rangle : \mathsf{real}^+$. The function $[\![\lambda x \cdot \langle x\rangle]\!]$ is continuous. So, we only need to show that

$$\forall Q : P \rightarrow^* Q \cdot \exists R : (\lambda x \cdot \langle x\rangle)\, P \rightarrow^* R \cdot [\![\lambda x \cdot \langle x\rangle]\!]\, (\mathsf{value}\,(Q)) \sqsubseteq \mathsf{value}\,(R)$$

for every closed computable term P. So, let P be a closed computable term and consider $Q : P \rightarrow^* Q$.

  – Consider $Q = X^\circ$.

$$(\lambda x \cdot \langle x\rangle)\, P \rightarrow \langle P\rangle \rightarrow^* \langle Q\rangle = \langle X^\circ\rangle\,.$$

  – Consider $Q = X$.

$$(\lambda x \cdot \langle x\rangle)\, P \rightarrow \langle P\rangle \rightarrow^* \langle Q\rangle = \langle X\rangle\,.$$

- Consider $\langle P\rangle\, Q : \mathsf{real}^+$. The function $[\![\lambda x \cdot \langle x\rangle]\!]$ is continuous because Möbius transformations are continuous. So, we only need to show that

$$\forall R : P \rightarrow^* R \cdot \forall S : Q \rightarrow^* S \cdot \exists D : (\lambda x \cdot \langle x\rangle)\, PQ \rightarrow^* D \cdot$$

$$[\![\lambda x \cdot \langle x\rangle]\!]\, (\mathsf{value}\,(R))\, (\mathsf{value}\,(S)) \sqsubseteq \mathsf{value}\,(D)$$

for all closed computable terms P and Q. So, let P and Q be closed computable terms and consider $R : P \rightarrow^* R$ and $S : Q \rightarrow^* S$.

– Consider $R = X^\circ$.

$$(\lambda x \cdot \langle x \rangle)\, PQ \to \langle P \rangle\, Q \to^* \langle R \rangle\, Q = \langle X^\circ \rangle\, Q.$$

– Consider $R = X$. By Lemma 79 and Lemma 80, there exists D : $\langle X \rangle\, Q \to^* D$ such that $[\![\langle X \rangle]\!]\,(\mathsf{value}\,(S)) = \mathsf{value}\,(D)$.

$$(\lambda x \cdot \langle x \rangle)\, PQ \to \langle P \rangle\, Q \to^* \langle X \rangle\, Q \to^* D. \blacksquare$$

Proof of Lemma 82:
The proof is by structural induction on P. Let σ be a substitution of closed computable terms for the free variables in P. Thus, we must show that σP is computable.

- For $P = x$, 0, true, false, $\mathsf{succ}(Q)$, $\mathsf{pred}(Q)$, $\mathsf{zero}(Q)$, $\mathsf{if}\, Q\, \mathsf{then}\, R\, \mathsf{else}\, S$ or QR, P is computable because it is in PCF.
- For $P = \lambda x.Q$, we must show that $R = (\lambda x.\sigma Q)P_1P_2 \ldots P_n$ is computable if $P_1, P_2, \ldots, P_n$ are closed computable terms and R has ground type t.
  - For $t = \mathsf{num}$ or bool, R is computable because it is in PCF.
  - For $t = \mathsf{real}^+$ or $s \times t$, observe that

    $$R \to ([P_1/x](\sigma Q))P_2P_3 \ldots P_n = ((\sigma[P_1/x])Q)P_2P_3 \ldots P_n,$$

    which we will call S. But, S is computable because Q is computable. Therefore, $[\![R]\!] \sqsubseteq \mathsf{eval}\,(R)$ because $[\![R]\!] = [\![S]\!]$, $\mathsf{eval}\,(R) = \mathsf{eval}\,(S)$ and $[\![S]\!] \sqsubseteq \mathsf{eval}\,(S)$.
- For $P = \mu x.Q$, we must show that $R = SP_1P_2 \ldots P_n$ where $S \equiv \mu x.\sigma Q$ is computable if $P_1, P_2, \ldots, P_n$ are closed computable terms and R has ground type t.
  - For $t = \mathsf{num}$ or bool, R is computable because it is in PCF.
  - For $t = \mathsf{real}^+$ or $r \times s$, define S^n by

    $$\begin{aligned} S^0 &= \mu x.x \\ S^{n+1} &= (\lambda x.\sigma Q)\, S^n. \end{aligned}$$

It is easy to show by induction on n that $[\![S]\!] = \bigsqcup_{n=0}^{\infty}[\![S^n]\!]$. Clearly, S^n is computable for all $n \in \mathbb{N}$. Let $a \ll [\![R]\!]$.

$$\begin{aligned}
[\![R]\!] &= [\![S]\!][\![P_1]\!][\![P_2]\!]\dots[\![P_n]\!] \\
&= \left(\bigsqcup_{n=0}^{\infty}[\![S^n]\!]\right)[\![P_1]\!][\![P_2]\!]\dots[\![P_n]\!] \\
&= \bigsqcup_{n=0}^{\infty}\left([\![S^n]\!][\![P_1]\!][\![P_2]\!]\dots[\![P_n]\!]\right) \\
&= \bigsqcup_{n=0}^{\infty}[\![S^n P_1 P_2 \dots P_n]\!].
\end{aligned}$$

However, $\exists n \in \mathbb{N}$ such that $a \ll [\![S^n P_1 P_2 \dots P_n]\!]$, therefore $\exists S_n$: $S^n P_1 P_2 \dots P_n \to^* S_n$ such that $a \sqsubseteq \mathsf{value}\,(S_n)$ because S^n is computable. Therefore $\exists R_n : R \to^* R_n$ such that $\mathsf{value}\,(R_n) = \mathsf{value}\,(S_n)$ by a straightforward extension of the Unwinding Theorem [29].

- For $P = (Q, R)$. Proved in Lemma 77.
- For $P = \mathsf{fst}(Q)$ or $\mathsf{snd}(Q)$. Proved in Lemma 78.
- For $P = \langle Q \rangle$. Proved in Lemma 81. ∎

Proof of Lemma 85:

$$\hat{f}\,(\hat{x}) = \bigsqcup_{x:\hat{x}\sqsubseteq\hat{y}} \widehat{f\,(y)} = \widehat{f\,(x)}.\ \blacksquare$$

Proof of Lemma 86:
The proof is by induction on the definition of $\to$. The rules $P \to Q$ with $\mathsf{value}\,(P) = \bot$ are trivial.
Basis:

$$\begin{aligned}
\mathsf{value}\,(\mathsf{fst}\,(P, Q)) &= \mathsf{fst}\,(\mathsf{value}\,(P)\,, \mathsf{value}\,(Q)) \\
&= \mathsf{value}\,(P) \\
\mathsf{value}\,(\mathsf{snd}\,(P, Q)) &= \mathsf{snd}\,(\mathsf{value}\,(P)\,, \mathsf{value}\,(Q)) \\
&= \mathsf{value}\,(Q) \\
\mathsf{value}\,\left([M\rangle\,\langle V^+\rangle\right) &= \mathsf{info}\,(M) \\
&\sqsubseteq \mathsf{info}\,\left(M \bullet V^+\right)
\end{aligned}$$

$$\begin{array}{rcl}
 & = & \mathsf{value}\left(\left[M \bullet V^{+}\right\rangle\right) \\
\mathsf{value}\left([M\rangle\left(\left\langle N^{+}\right\rangle P\right)\right) & = & \mathsf{info}\,(M) \\
 & \sqsubseteq & \mathsf{info}\left(M \bullet N^{+}\right) \\
 & = & \mathsf{value}\left(\left[M \bullet N^{+}\right\rangle P\right) \\
\mathsf{value}\,([M\rangle\, A^{\circ}) & = & \mathsf{info}\,(M) \\
 & = & \mathsf{value}\,([M\rangle\, P) \\
\mathsf{value}\left(\langle M\rangle \left\langle V^{+}\right\rangle\right) & = & [\![\langle M\rangle]\!]\,([0,\infty]) \\
 & \sqsubseteq & [\![\left\langle M \bullet V^{+}\right\rangle]\!] \\
 & = & \mathsf{value}\left(\left\langle M \bullet V^{+}\right\rangle\right) \\
\mathsf{value}\left(\langle M\rangle\left(\left\langle N^{+}\right\rangle P\right)\right) & = & [\![\langle M\rangle]\!]\,([0,\infty]) \\
 & \sqsubseteq & [\![\left\langle M \bullet N^{+}\right\rangle]\!]\,([0,\infty]) \\
 & = & \mathsf{value}\left(\left\langle M \bullet N^{+}\right\rangle P\right) \\
\mathsf{value}\,(\langle M\rangle\, A^{\circ}) & = & [\![\langle M\rangle]\!]\,([0,\infty]) \\
 & = & \mathsf{value}\,(\langle M\rangle\, P)
\end{array}$$

Inductive Step:

$$\frac{\mathsf{value}\,(P) \sqsubseteq \mathsf{value}\,(R) \quad \mathsf{value}\,(Q) \sqsubseteq \mathsf{value}\,(S)}{\mathsf{value}\,(P,Q) = (\mathsf{value}\,(P)\,, \mathsf{value}\,(Q)) \sqsubseteq (\mathsf{value}\,(R)\,, \mathsf{value}\,(S)) = \mathsf{value}\,(R,S)}$$

$$\frac{\mathsf{value}\,(P) \sqsubseteq \mathsf{value}\,(Q)}{\mathsf{value}\,(P,X) = (\mathsf{value}\,(P)\,, \mathsf{value}\,(X)) \sqsubseteq (\mathsf{value}\,(Q)\,, \mathsf{value}\,(X)) = \mathsf{value}\,(Q,X)}$$

$$\frac{\mathsf{value}\,(P) \sqsubseteq \mathsf{value}\,(Q)}{\mathsf{value}\,(X,P) = (\mathsf{value}\,(X)\,, \mathsf{value}\,(P)) \sqsubseteq (\mathsf{value}\,(X)\,, \mathsf{value}\,(Q)) = \mathsf{value}\,(X,Q)}. \quad \blacksquare$$

Proof of Lemma 87:
The proof is by induction on the definition of $\rightarrow$. Trivial. ■

Proof of Lemma 88:
The proof is by induction on the structure of P. Trivial. ■

Proof of Proposition 89:
Assume P is an LAR program without the parallel construct. But $P \rightarrow Q$ implies $[\![P]\!] = [\![Q]\!]$. Therefore, $[\![\mathsf{reduce}\,(P,n)]\!] = [\![\mathsf{reduce}\,(P,n+1)]\!]$. Therefore, $[\![P]\!] = [\![\mathsf{reduce}\,(P,n)]\!]$ for all $n \in \mathbb{N}$. But $\mathsf{value}\,(\mathsf{reduce}\,(P,n)) \sqsubseteq [\![\mathsf{reduce}\,(P,n)]\!]$ for all $n \in \mathbb{N}$. Therefore, $\mathsf{step}\,(P,n) \sqsubseteq [\![P]\!]$ for all $n \in \mathbb{N}$. Therefore $\bigsqcup_{n=0}^{\infty} \mathsf{step}\,(P,n) \sqsubseteq [\![P]\!]$. But, $\mathsf{eval}\,(P) = \bigsqcup_{n=0}^{\infty} \mathsf{step}\,(P,n)$. Therefore, $\mathsf{eval}(P) \sqsubseteq [\![P]\!]$. ■

Proof of Lemma 90:

"$\Rightarrow$". Assume P is computable. Therefore, $[\![P]\!] \sqsubseteq \mathsf{eval}(P)$. Consider $a \ll [\![P]\!]$. So, $a \ll \mathsf{eval}(P) = \bigsqcup_{n=0}^{\infty} \mathsf{step}(P, n)$. Therefore, $\exists n \in \mathbb{N}$ such that $a \sqsubseteq \mathsf{step}(P, n)$. "$\Leftarrow$". Trivial. ■

Proof of Lemma 91:

Consider computable Q. Consider $a \ll [\![PQ]\!]$. But, $[\![PQ]\!] = [\![P]\!][\![Q]\!]$. By continuity of $[\![P]\!]$, there exists $b \ll [\![Q]\!]$ such that $a \ll [\![P]\!]b$. By computability of Q, there exists $n \in \mathbb{N}$ such that $b \sqsubseteq \mathsf{step}(Q, n)$. But, there exists $m \in \mathbb{N}$ such that $[\![P]\!](\mathsf{step}(Q, n)) \sqsubseteq \mathsf{step}(PQ, m)$. By monotonicity of $[\![P]\!]$, $[\![P]\!]b \sqsubseteq [\![P]\!]\mathsf{step}(Q, n)$. So, $a \ll [\![P]\!]b \sqsubseteq [\![P]\!](\mathsf{step}(Q, n)) \sqsubseteq \mathsf{step}(PQ, m)$. Therefore, $a \sqsubseteq \mathsf{step}(PQ, m)$. ■

Proof of Lemma 92:

We only need to check the computability of those constructs that appear in the new reduction rules. Consider $\mathsf{succ}(P)$. The function $x \mapsto \mathsf{succ}(x)$ is clearly continuous. So, we only need to show that

$$\forall n \in \mathbb{N} \cdot \exists m \in \mathbb{N} \cdot \mathsf{succ}(\mathsf{step}(P, n)) \sqsubseteq \mathsf{step}(\mathsf{succ}(P), m)$$

for any closed computable term P. Let P be a closed computable term and $n \in \mathbb{N}$. Let

$$i = \mu n \cdot (\exists s \in \mathbb{N} \cdot \mathsf{step}(P, n) = \mathsf{succ}^s(0)).$$

- Consider $i > n$.

$$\begin{aligned} & \mathsf{succ}(\mathsf{step}(P, n)) \\ = {} & \bot \\ = {} & \mathsf{step}(\mathsf{succ}(P), 0). \end{aligned}$$

- Consider $i \leq n$. Clearly $i \neq \infty$.

$$\begin{aligned} & \mathsf{succ}(\mathsf{step}(P, n)) \\ = {} & \mathsf{succ}(\mathsf{step}(\mathsf{reduce}(P, i), n - i)) \\ = {} & \mathsf{succ}(\mathsf{step}(\mathsf{succ}^s(0), n - i)) \\ = {} & \mathsf{step}\left(\mathsf{succ}^{s+1}(0), n - i\right) \\ = {} & \mathsf{step}(\mathsf{succ}(\mathsf{succ}^s(0)), n + 1 - i) \\ = {} & \mathsf{step}(\mathsf{succ}(\mathsf{reduce}(P, i)), n + 1 - i) \\ = {} & \mathsf{step}(\mathsf{succ}(P), n + 1). \end{aligned}$$

Similarly for $\mathsf{pred}\,(P)$, $\mathsf{zero}\,(P)$ and $\mathsf{if}\,P\,\mathsf{then}\,Q\,\mathsf{else}\,R$. ■

Proof of Lemma 93:

Consider (P, Q). The function $x \mapsto y \mapsto (x, y)$ is clearly continuous. So, we only need to show that

$$\forall n, m \in \mathbb{N} \cdot \exists r \in \mathbb{N} \cdot (\mathsf{step}\,(P, n)\,, \mathsf{step}\,(Q, m)) \sqsubseteq \mathsf{step}\,((P, Q)\,, r)$$

for all closed computable terms P and Q. Let P and Q be closed computable terms and consider $n, m \in \mathbb{N}$.

$$\begin{aligned}
& (\mathsf{step}\,(P, n)\,, \mathsf{step}\,(Q, m)) \\
\sqsubseteq\ & (\mathsf{step}\,(P, \mathsf{max}\,(n, m))\,, \mathsf{step}\,(Q, \mathsf{max}\,(n, m))) \\
=\ & \mathsf{value}\,((\mathsf{reduce}\,(P, \mathsf{max}\,(n, m))\,, \mathsf{reduce}\,(Q, \mathsf{max}\,(n, m)))) \\
=\ & \mathsf{step}\,((P, Q)\,, \mathsf{max}\,(n, m))\,. \ ■
\end{aligned}$$

Proof of Lemma 94:

Consider $\mathsf{fst}(P)$. The function $x \mapsto \mathsf{fst}\,(x)$ is clearly continuous. So, we only need to show that

$$\forall n \in \mathbb{N} \cdot \exists m \in \mathbb{N} \cdot \mathsf{fst}\,(\mathsf{step}\,(P, n)) \sqsubseteq \mathsf{step}\,(\mathsf{fst}\,(P)\,, m)$$

for any closed computable term P. Let P be a closed computable term and $n \in \mathbb{N}$. Let

$$i = \mu n \cdot (\mathsf{step}\,(P, n) = (Q, R))\,.$$

- Consider $i > n$.

$$\begin{aligned}
& \mathsf{fst}\,(\mathsf{step}\,(P, n)) \\
=\ & \bot \\
=\ & \mathsf{step}\,(\mathsf{fst}\,(P)\,, 0)\,.
\end{aligned}$$

- Consider $i \leq n$. Clearly $i \neq \infty$.

$$\begin{aligned}
& \mathsf{fst}\,(\mathsf{step}\,(P, n)) \\
=\ & \mathsf{fst}\,(\mathsf{step}\,(\mathsf{reduce}\,(P, i)\,, n - i)) \\
=\ & \mathsf{fst}\,(\mathsf{step}\,((Q, R)\,, n - i)) \\
=\ & \mathsf{step}\,(Q, n - i) \\
=\ & \mathsf{step}\,(\mathsf{fst}\,((Q, R))\,, n + 1 - i) \\
=\ & \mathsf{step}\,(\mathsf{fst}\,(\mathsf{reduce}\,(P, i))\,, n + 1 - i) \\
=\ & \mathsf{step}\,(\mathsf{fst}\,(P)\,, n + 1)\,.
\end{aligned}$$

Similarly for $\mathsf{snd}\,(P)$. ■

Proof of Lemma 95:

The function $x \mapsto y \mapsto x \lhd y$ is clearly continuous. So, we only need to show that

$$\forall n \in \mathbb{N} \cdot \exists m \in \mathbb{N} \cdot [\![P\lhd]\!]\,(\mathsf{step}\,(Q, n)) \sqsubseteq \mathsf{step}\,(P \lhd Q, m)$$

for all closed computable terms P and Q. Let P and Q be closed computable terms and consider $n \in \mathbb{N}$. Let

$$i = \mu n \cdot \mathsf{step}\,(Q, n) \subseteq (0, \infty)\,.$$

- Consider $i > n$.

$$\begin{array}{rcl} [\![P\lhd]\!]\,(\mathsf{step}\,(Q, n)) & = & \bot \\ & = & \mathsf{step}\,(P \lhd Q, 0)\,. \end{array}$$

- Consider $i \leq n$. Clearly $i \neq \infty$.

  – Consider $\mathsf{reduce}\,(Q, i) = [V\rangle$. Using the computability of P, there exists $m \in \mathbb{N}$ such that

$$[\![P]\!]\,(\mathsf{step}\,(\langle|V|\rangle\,, n - i)) \sqsubseteq \mathsf{step}\,(P\,(\langle|V|\rangle)\,, m - i - 1)\,.$$

  Therefore

$$\begin{array}{rcl} [\![P\lhd]\!]\,(\mathsf{step}\,(Q, n)) & = & [\![P\lhd]\!]\,(\mathsf{step}\,(\mathsf{reduce}\,(Q, i)\,, n - i)) \\ & = & [\![P\lhd]\!]\,(\mathsf{step}\,([V\rangle\,, n - i)) \\ & = & [\![P]\!]\,(\mathsf{step}\,(\langle|V|\rangle\,, n - i)) \\ & \sqsubseteq & \mathsf{step}\,(P\,(\langle|V|\rangle)\,, m - i - 1) \\ & = & \mathsf{step}\,(P \lhd [V\rangle\,, m - i) \\ & = & \mathsf{step}\,(P \lhd \mathsf{reduce}\,(Q, i)\,, m - i) \\ & = & \mathsf{step}\,(P \lhd Q, m)\,. \end{array}$$

  – Consider $\mathsf{reduce}\,(Q, i) = [M\rangle\,R$. Using the computability of P, there exists $m \in \mathbb{N}$ such that

$$[\![P]\!]\,(\mathsf{step}\,(\langle|M|\rangle\,R, n - i)) \sqsubseteq \mathsf{step}\,(P\,(\langle|M|\rangle\,R)\,, m - i - 1)\,.$$

Therefore

$$\begin{aligned}
[\![P\lhd]\!]\,(\mathsf{step}\,(Q,n)) &= [\![P\lhd]\!]\,(\mathsf{step}\,(\mathsf{reduce}\,(Q,i)\,,n-i)) \\
&= [\![P\lhd]\!]\,(\mathsf{step}\,([M\rangle\,R,n-i)) \\
&= [\![P]\!]\,(\mathsf{step}\,(\langle|M|\rangle\,R,n-i)) \\
&\sqsubseteq \mathsf{step}\,(P\,(\langle|M|\rangle\,R)\,,m-i-1) \\
&= \mathsf{step}\,(P\lhd[M\rangle\,R,m-i) \\
&= \mathsf{step}\,(P\lhd\mathsf{reduce}\,(Q,i)\,,m-i) \\
&= \mathsf{step}\,(P\lhd Q,m)\,.\ \blacksquare
\end{aligned}$$

Proof of Lemma 96:
Consider $[M]$.

- Consider $P = A^\circ \to Q$.

$$\begin{aligned}
[M]\,P &= [M]\,A^\circ \\
&\to [M]\,Q.
\end{aligned}$$

- Consider $P = [V\rangle$.

$$\begin{aligned}
[M]\,P &= [M]\,[V\rangle \\
&\to [M\bullet V\rangle\,.
\end{aligned}$$

- Consider $P = [N\rangle\,\langle V^+\rangle \to [N\bullet V^+\rangle$.

$$\begin{aligned}
[M]\,P &= [M]\,\big([N\rangle\,\langle V^+\rangle\big) \\
&\to [M\bullet N\rangle\,\langle V^+\rangle \\
&\to \big[M\bullet N\bullet V^+\big\rangle\,.
\end{aligned}$$

- Consider $P = [N\rangle\,(\langle O^+\rangle\,Q) \to [N\bullet O^+\rangle\,Q$.

$$\begin{aligned}
[M]\,P &= [M]\,\big([N\rangle\,(\langle O^+\rangle\,Q)\big) \\
&\to [M\bullet N\rangle\,(\langle O^+\rangle\,Q) \\
&\to \big[M\bullet N\bullet O^+\big\rangle\,Q.
\end{aligned}$$

- Consider $P = [N\rangle\, A^\circ \to [N\rangle\, Q$.

$$\begin{array}{rcl} [M]\,P & = & [M]\,([N\rangle\, A^\circ) \\ & \to & [M \bullet N\rangle\, A^\circ \\ & \to & [M \bullet N\rangle\, Q. \end{array}$$

Similarly for $[M\rangle$, $\langle M\rangle$. ■

Proof of Lemma 97:
Consider $[M]$. The proof is by induction on $n \in \mathbb{N}$.
Basis:

$$\mathsf{reduce}\,([M]\,(\mathsf{reduce}\,(P, 0))\,, 1) = \mathsf{reduce}\,([M]\,P, 1)\,.$$

Inductive Step: Assume

$$\mathsf{reduce}\,([M]\,(\mathsf{reduce}\,(P, n))\,, 1) = \mathsf{reduce}\,([M]\,P, n+1)\,.$$

Using lemma (96)

$$\mathsf{reduce}\,([M]\,(\mathsf{reduce}\,(P, 1))\,, 1) = \mathsf{reduce}\,([M]\,P, 2)$$

$$\begin{array}{rcl} \mathsf{reduce}\,([M]\,(\mathsf{reduce}\,(P, n+1))\,, 1) & = & \mathsf{reduce}\,([M]\,(\mathsf{reduce}\,(\mathsf{reduce}\,(P, n)\,, 1))\,, 1) \\ & = & \mathsf{reduce}\,([M]\,(\mathsf{reduce}\,(P, n))\,, 2) \\ & = & \mathsf{reduce}\,(\mathsf{reduce}\,([M]\,(\mathsf{reduce}\,(P, n))\,, 1)\,, 1) \\ & = & \mathsf{reduce}\,(\mathsf{reduce}\,([M]\,P, n+1)\,, 1) \\ & = & \mathsf{reduce}\,([M]\,P, n+2)\,. \end{array}$$

Similarly for $[M\rangle$, $\langle M\rangle$. ■

Proof of Lemma 98:
Consider $[M]$.

- Consider $P = A^\circ \to Q$.

$$\begin{array}{rcl} [\![[M]]\!]\,(\mathsf{value}\,(P)) & = & [\![[M]]\!]\,(\mathsf{value}\,(A^\circ)) \\ & = & [\![[M]]\!]\,(\mathbb{R}^\infty) \\ & = & \mathbb{R}^\infty \\ & = & \mathsf{value}\,([M]\,Q) \\ & = & \mathsf{step}\,([M]\,P, 1)\,. \end{array}$$

- Consider $P = [V\rangle$.

$$\begin{aligned}
[\![[M]]\!] (\mathsf{value}(P)) &= [\![[M]]\!] (\mathsf{value}([V\rangle)) \\
&= [\![M]\!] (\mathsf{info}(V)) \\
&= \mathsf{info}(M \bullet V) \\
&= \mathsf{value}([M \bullet V\rangle) \\
&= \mathsf{step}([M]\,[V\rangle, 1) \\
&= \mathsf{step}([M]\,P, 1)\,.
\end{aligned}$$

- Consider $P = [N\rangle\, Q$.

$$\begin{aligned}
[\![[M]]\!] (\mathsf{value}(P)) &= [\![[M]]\!] (\mathsf{value}([N\rangle\, Q)) \\
&= [\![M]\!] (\mathsf{info}(N)) \\
&= \mathsf{info}(M \bullet N) \\
&= \mathsf{value}([M \bullet N\rangle\, Q) \\
&= \mathsf{step}([M]\,([N\rangle\, Q), 1) \\
&= \mathsf{step}([M]\,P, 1)\,.
\end{aligned}$$

Similarly for $[M\rangle$, $\langle M\rangle$. ■

Proof of Lemma 99:
Consider $[M]$ and $n \in \mathbb{N}$. Using Lemma 98 and lemma (97)

$$\begin{aligned}
[\![[M]]\!] (\mathsf{step}(P, n)) &= [\![[M]]\!] (\mathsf{value}(\mathsf{reduce}(P, n))) \\
&= \mathsf{step}([M]\,(\mathsf{reduce}(P, n)), 1) \\
&= \mathsf{value}(\mathsf{reduce}([M]\,(\mathsf{reduce}(P, n)), 1)) \\
&= \mathsf{value}(\mathsf{reduce}([M]\,P, n+1)) \\
&= \mathsf{step}([M]\,P, n+1)\,.
\end{aligned}$$

Similarly for $[M\rangle$, $\langle M\rangle$ by symmetry in the matrix rules. ■

Proof of Lemma 100:
Consider $[T]$.
Consider $P = B^\circ$.

$$\begin{aligned}
&\mathsf{step}([T]\,(\mathsf{reduce}(P, 1)), n) \\
= \;&\mathsf{step}([T]\,P, n+1)\,.
\end{aligned}$$

Consider $P = ([V\rangle, [W\rangle)$.

$$\begin{aligned}
& \mathsf{step}\left([T\rangle\,(\mathsf{reduce}\,(P,1)), n\right) \\
= {} & \mathsf{step}\left([T]\,P, n\right).
\end{aligned}$$

Consider $P = ([M\rangle\,\langle V^+\rangle, [W\rangle)$.

$$\begin{aligned}
& \mathsf{step}\left([T]\,(\mathsf{reduce}\,(P,1)), n\right) \\
= {} & \mathsf{step}\left([T]\left(\left[M \bullet V^+\right\rangle, [W\rangle\right), n\right) \\
\sqsubseteq {} & \mathsf{step}\left(\left[T \bullet_1 \left(M \bullet V^+\right) \bullet W\right\rangle, n\right) \\
= {} & \mathsf{step}\left([T \bullet_1 M \bullet_2 W\rangle \left\langle V^+\right\rangle, n+1\right) \\
= {} & \mathsf{step}\left([T]\,P, n+2\right).
\end{aligned}$$

Consider $P = \left([M\rangle \left\langle M_0^+\right\rangle R, [W\rangle\right)$.

$$\begin{aligned}
& \mathsf{step}\left([T]\,(\mathsf{reduce}\,(P,1)), n\right) \\
= {} & \mathsf{step}\left([T]\left(\left[M \bullet M_0^+\right\rangle R, [W\rangle\right), n\right) \\
\sqsubseteq {} & \mathsf{step}\left(\left[T \bullet_1 \left(M \bullet M_0^+\right) \bullet_2 W\right\rangle R, n\right) \\
= {} & \mathsf{step}\left([T \bullet_1 M \bullet_2 W\rangle \left\langle M_0^+\right\rangle R, n+1\right) \\
= {} & \mathsf{step}\left([T]\,P, n+2\right).
\end{aligned}$$

Consider $P = ([M\rangle\,A_1^\circ, [W\rangle)$.

$$\begin{aligned}
& \mathsf{step}\left([T]\,(\mathsf{reduce}\,(P,1)), n\right) \\
= {} & \mathsf{step}\left([T]\,(([M\rangle\,(\mathsf{reduce}\,(A_1^\circ,1)), [W\rangle)), n\right) \\
\sqsubseteq {} & \mathsf{step}\left([T \bullet_1 M \bullet_2 W\rangle\,(\mathsf{reduce}\,(A_1^\circ,1)), n\right) \\
= {} & \mathsf{step}\left([T \bullet_1 M \bullet_2 W\rangle\,A_1^\circ, n+1\right) \\
= {} & \mathsf{step}\left([T]\,P, n+2\right).
\end{aligned}$$

Consider $P = ([M\rangle\,\langle V^+\rangle, [N\rangle\,\langle W\rangle)$.

$$\begin{aligned}
& \mathsf{step}\left([T]\,(\mathsf{reduce}\,(P,1)), n\right) \\
= {} & \mathsf{step}\left([T]\left(\left[M \bullet V^+\right\rangle, [N \bullet W\rangle\right), n\right) \\
\sqsubseteq {} & \mathsf{step}\left(\left[T \bullet_1 \left(M \bullet V^+\right) \bullet (N \bullet W)\right\rangle, n\right) \\
= {} & \mathsf{step}\left(\left[X^{\mathrm{head}}\right\rangle \left\langle X^{\mathrm{tail}} \bullet_1 V^+ \bullet W\right\rangle, n+1\right) \\
= {} & \mathsf{step}\left(\left[X^{\mathrm{head}}\right\rangle \left\langle X^{\mathrm{tail}}\right\rangle' \left(\left\langle V^+\right\rangle, \langle W\rangle\right), n+2\right) \\
= {} & \mathsf{step}\left([X\rangle \left(\left\langle V^+\right\rangle, \langle W\rangle\right), n+3\right) \\
= {} & \mathsf{step}\left([T]\,P, n+4\right).
\end{aligned}$$

where

$$X = T \bullet_1 M \bullet_2 N.$$

Consider $P = \left([M\rangle \left\langle M_0^+\right\rangle R, [N\rangle \langle W\rangle\right)$.

$$\begin{aligned}
& \mathsf{step}\,([T]\,(\mathsf{reduce}\,(P,1))\,,n) \\
= \; & \mathsf{step}\,\left([T]\,\left(\left[M \bullet M_0^+\right\rangle R, [N \bullet W\rangle\right)\,,n\right) \\
\sqsubseteq \; & \mathsf{step}\,\left(\left[T \bullet_1 \left(M \bullet M_0^+\right) \bullet_2 (N \bullet W)\right\rangle R, n\right) \\
= \; & \mathsf{step}\,\left(\left[X^{\mathrm{head}}\right\rangle \left\langle X^{\mathrm{tail}} \bullet_1 M_0^+ \bullet_2 W\right\rangle R, n+1\right) \\
= \; & \mathsf{step}\,\left(\left[X^{\mathrm{head}}\right\rangle \left\langle X^{\mathrm{tail}}\right\rangle' \left(\left\langle M_0^+\right\rangle R, \langle W\rangle\right)\,,n+2\right) \\
= \; & \mathsf{step}\,\left([X\rangle \left(\left\langle M_0^+\right\rangle R, \langle W\rangle\right)\,,n+3\right) \\
= \; & \mathsf{step}\,([T]\,P, n+4)
\end{aligned}$$

where

$$X = T \bullet_1 M \bullet_2 N.$$

Consider $P = ([M\rangle A_1^\circ, [N\rangle \langle W^+\rangle)$.

$$\begin{aligned}
& \mathsf{step}\,([T]\,(\mathsf{reduce}\,(P,1))\,,n) \\
= \; & \mathsf{step}\,\left([T]\,\left([M\rangle\,(\mathsf{reduce}\,(A_1^\circ,1))\,, \left[N \bullet W^+\right\rangle\right)\,,n\right) \\
\sqsubseteq \; & \mathsf{step}\,\left(\left[T \bullet_1 M \bullet_2 \left(N \bullet W^+\right)\right\rangle (\mathsf{reduce}\,(A_1^\circ,1))\,,n\right) \\
= \; & \mathsf{step}\,\left(\left[T \bullet_1 M \bullet_2 \left(N \bullet W^+\right)\right\rangle A_1^\circ, n+1\right) \\
= \; & \mathsf{step}\,\left(\left[X^{\mathrm{head}}\right\rangle \left\langle X^{\mathrm{tail}} \bullet_2 W^+\right\rangle A_1^\circ, n+2\right) \\
= \; & \mathsf{step}\,\left(\left[X^{\mathrm{head}}\right\rangle \left\langle X^{\mathrm{tail}}\right\rangle' \left(A_1^\circ, \left\langle W^+\right\rangle\right)\,,n+3\right) \\
= \; & \mathsf{step}\,\left([X\rangle \left(A_1^\circ, \left\langle W^+\right\rangle\right)\,,n+4\right) \\
= \; & \mathsf{step}\,([T]\,P, n+5)
\end{aligned}$$

where

$$X = T \bullet_1 M \bullet_2 N.$$

Consider $P = \left([M\rangle \left\langle M_0^+\right\rangle R, [N\rangle \left\langle N_0^+\right\rangle S\right)$.

$$\begin{aligned}
& \mathsf{step}\,([T]\,(\mathsf{reduce}\,(P,1))\,,n) \\
= \; & \mathsf{step}\,\left([T]\,\left(\left[M \bullet M_0^+\right\rangle R, \left[N \bullet N_0^+\right\rangle S\right)\,,n\right) \\
\sqsubseteq \; & \mathsf{step}\,\left(\left[Z^{\mathrm{head}}\right\rangle \left\langle Z^{\mathrm{tail}}\right\rangle' (R, S)\,,n\right) \\
= \; & \mathsf{step}\,\left(\left[X^{\mathrm{head}}\right\rangle \left\langle Y^{\mathrm{head}}\right\rangle \left\langle Y^{\mathrm{tail}}\right\rangle' (R, S)\,,n+1\right) \\
= \; & \mathsf{step}\,\left(\left[X^{\mathrm{head}}\right\rangle \langle Y\rangle\,(R, S)\,,n+2\right)
\end{aligned}$$

$$\begin{aligned}
&= \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle X^{\text{tail}}\right\rangle' \left(\left\langle M_0^+\right\rangle R, \left\langle N_0^+\right\rangle S\right), n+3\right)\\
&= \mathsf{step}\left([X\rangle \left(\left\langle M_0^+\right\rangle R, \left\langle N_0^+\right\rangle S\right), n+4\right)\\
&= \mathsf{step}\left([T]\, P, n+5\right)
\end{aligned}$$

where

$$\begin{aligned}
X &= T \bullet_1 M \bullet_2 N\\
Y &= X^{\text{tail}} \bullet_1 M_0^+ \bullet_2 N_0^+\\
Z &= T \bullet_1 \left(M \bullet M_0^+\right) \bullet_2 \left(N \bullet N_0^+\right).
\end{aligned}$$

Consider

$$P = \left([M\rangle\, A_1^\circ, [N\rangle \left\langle N_0^+\right\rangle \left\langle N_1^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\right).$$

$$\begin{aligned}
&\ \mathsf{step}\left([T]\left(\mathsf{reduce}\,(P,1)\right), n\right)\\
&= \mathsf{step}\left([T]\begin{pmatrix} [M\rangle\left(\mathsf{reduce}\,(A_1^\circ,1)\right)\\ \left[N \bullet N_0^+\right\rangle \left\langle N_1^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\end{pmatrix}, n\right)\\
&\sqsubseteq \mathsf{step}\left(\left[Z_s \bullet Y_s^{\text{tail}} \bullet_2 W^+\right\rangle \left(\mathsf{reduce}\,(A_1^\circ,1)\right), n\right)\\
&= \mathsf{step}\left(\left[Z_s \bullet Y_s^{\text{tail}} \bullet_2 W^+\right\rangle A_1^\circ, n+1\right)\\
&= \mathsf{step}\left([Z_s\rangle \left\langle Y_s^{\text{tail}} \bullet_2 W^+\right\rangle A_1^\circ, n+2\right)\\
&= \mathsf{step}\left([Z_s\rangle \left\langle Y_s^{\text{tail}}\right\rangle' \left(A_1^\circ, \left\langle W^+\right\rangle\right), n+3\right)\\
&\ \vdots\\
&= \mathsf{step}\left([Z_i\rangle \left\langle Y_{i+1}^{\text{head}}\right\rangle \left\langle Y_{i+1}^{\text{tail}}\right\rangle' \begin{pmatrix} A_1^\circ\\ \left\langle N_{i+2}^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\end{pmatrix}, n+3s+1-3i\right)\\
&= \mathsf{step}\left([Z_i\rangle \left\langle Y_{i+1}\right\rangle \left(A_1^\circ, \left\langle N_{i+2}^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\right), n+3s+2-3i\right)\\
&= \mathsf{step}\left([Z_i\rangle \left\langle Y_i^{\text{tail}}\right\rangle' \left(A_1^\circ, \left\langle N_{i+1}^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\right), n+3s+3-3i\right)\\
&\ \vdots\\
&= \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle Y_0^{\text{head}}\right\rangle \left\langle Y_0^{\text{tail}}\right\rangle' \left(A_1^\circ, \left\langle N_1^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\right), n+3s+4\right)\\
&= \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle Y_0\right\rangle \left(A_1^\circ, \left\langle N_1^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\right), n+3s+5\right)\\
&= \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle X^{\text{tail}}\right\rangle' \left(A_1^\circ, \left\langle N_0^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\right), n+3s+6\right)\\
&= \mathsf{step}\left([X\rangle \left(A_1^\circ, \left\langle N_0^+\right\rangle \cdots \left\langle N_s^+\right\rangle \left\langle W^+\right\rangle\right), n+3s+7\right)\\
&= \mathsf{step}\left([T]\, P, n+3s+8\right)
\end{aligned}$$

where

$$X = T \bullet_1 M \bullet_2 N$$

$$\begin{array}{rcl} Y_0 &=& X^{\text{tail}} \bullet_2 N_0^+ \\ Z_0 &=& X^{\text{head}} \bullet Y_0^{\text{head}} \\ Y_{n+1} &=& Y_n^{\text{tail}} \bullet_2 N_{n+1}^+ \\ Z_{n+1} &=& Z_n \bullet Y_{n+1}^{\text{head}}. \end{array}$$

Consider

$$P = \left([M\rangle\, A_1^\circ, [N\rangle \left\langle N_0^+\right\rangle \left\langle N_1^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ\right).$$

$$\begin{array}{rl} & \mathsf{step}\left([T]\left(\mathsf{reduce}\left(P,1\right)\right), n\right) \\ = & \mathsf{step}\left([T]\left(\begin{array}{l} [M\rangle\left(\mathsf{reduce}\left(A_1^\circ,1\right)\right) \\ \left[N \bullet N_0^+\right\rangle \left\langle N_1^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ \end{array}\right), n\right) \\ \sqsubseteq & \mathsf{step}\left([Z_s\rangle \left\langle Y_s^{\text{tail}}\right\rangle' \left(\mathsf{reduce}\left(A_1^\circ,1\right), \mathsf{reduce}\left(A_2^\circ,1\right)\right), n\right) \\ = & \mathsf{step}\left([Z_s\rangle \left\langle Y_s^{\text{tail}}\right\rangle' \left(A_1^\circ, A_2^\circ\right), n+1\right) \\ \vdots & \\ = & \mathsf{step}\left([Z_i\rangle \left\langle Y_{i+1}^{\text{head}}\right\rangle \left\langle Y_{i+1}^{\text{tail}}\right\rangle' \left(\begin{array}{l} A_1^\circ \\ \left\langle N_{i+2}^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ \end{array}\right), n+3s-1-3i\right) \\ = & \mathsf{step}\left([Z_i\rangle \left\langle Y_{i+1}\right\rangle \left(A_1^\circ, \left\langle N_{i+2}^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ\right), n+3s-3i\right) \\ = & \mathsf{step}\left([Z_i\rangle \left\langle Y_i^{\text{tail}}\right\rangle' \left(A_1^\circ, \left\langle N_{i+1}^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ\right), n+3s+1-3i\right) \\ \vdots & \\ = & \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle Y_0^{\text{head}}\right\rangle \left\langle Y_0^{\text{tail}}\right\rangle' \left(A_1^\circ, \left\langle N_1^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ\right), n+3s+2\right) \\ = & \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle Y_0\right\rangle \left(A_1^\circ, \left\langle N_1^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ\right), n+3s+3\right) \\ = & \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle X^{\text{tail}}\right\rangle' \left(A_1^\circ, \left\langle N_0^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ\right), n+3s+4\right) \\ = & \mathsf{step}\left([X\rangle \left(A_1^\circ, \left\langle N_0^+\right\rangle \cdots \left\langle N_s^+\right\rangle A_2^\circ\right), n+3s+5\right) \\ = & \mathsf{step}\left([T]\, P, n+3s+6\right) \end{array}$$

where

$$\begin{array}{rcl} X &=& T \bullet_1 M \bullet_2 N \\ Y_0 &=& X^{\text{tail}} \bullet_2 N_0^+ \\ Z_0 &=& X^{\text{head}} \bullet Y_0^{\text{head}} \\ Y_{n+1} &=& Y_n^{\text{tail}} \bullet_2 N_{n+1}^+ \\ Z_{n+1} &=& Z_n \bullet Y_{n+1}^{\text{head}}. \end{array}$$

Consider $P = \left([M\rangle\, A_1^\circ, [N\rangle\, A_2^\circ\right)$.

$$\mathsf{step}\left([T]\left(\mathsf{reduce}\left(P,1\right)\right), n\right)$$

$$\begin{aligned}
&= \mathsf{step}\left([T]\left([M\rangle\left(\mathsf{reduce}\left(A_1^\circ, 1\right)\right), [N\rangle\left(\mathsf{reduce}\left(A_2^\circ, 1\right)\right)\right), n\right) \\
&\sqsubseteq \mathsf{step}\left(\left[X^{\text{head}}\right\rangle\left\langle X^{\text{head}}\right\rangle'\left(\mathsf{reduce}\left(A_1^\circ, 1\right), \mathsf{reduce}\left(A_2^\circ, 1\right)\right), n\right) \\
&= \mathsf{step}\left(\left[X^{\text{head}}\right\rangle\left\langle X^{\text{head}}\right\rangle'\left(A_1^\circ, A_2^\circ\right), n+1\right) \\
&= \mathsf{step}\left([X\rangle\left(A_1^\circ, A_2^\circ\right), n+2\right) \\
&= \mathsf{step}\left([T]\left([M\rangle A_1^\circ, [N\rangle A_2^\circ\right), n+3\right) \\
&= \mathsf{step}\left([T] P, n+4\right)
\end{aligned}$$

where

$$X = T \bullet_1 M \bullet_2 N.$$

Consider $[T\rangle'$.
Consider $P = C^\circ$.

$$\begin{aligned}
&\mathsf{step}\left([T\rangle'\left(\mathsf{reduce}\left(P, 1\right)\right), n\right) \\
=\ &\mathsf{step}\left([T\rangle' P, n+1\right).
\end{aligned}$$

Consider $P = B$. Similar to $[T]$ by symmetry in the tensor rules.
Consider $P = (\langle V\rangle, A_2^\circ)$.

$$\begin{aligned}
&\mathsf{step}\left([T\rangle'\left(\mathsf{reduce}\left(P, 1\right)\right), n\right) \\
=\ &\mathsf{step}\left([T\rangle'\left(\langle V\rangle, \mathsf{reduce}\left(A_2^\circ, 1\right)\right), n\right) \\
\sqsubseteq\ &\mathsf{step}\left([T \bullet_1 V\rangle\left(\mathsf{reduce}\left(A_2^\circ, 1\right)\right), n\right) \\
=\ &\mathsf{step}\left([T \bullet_1 V\rangle A_2^\circ, n+1\right) \\
=\ &\mathsf{step}\left([T\rangle' P, n+2\right).
\end{aligned}$$

Consider $P = (\langle M\rangle\langle V^+\rangle, A_2^\circ)$.

$$\begin{aligned}
&\mathsf{step}\left([T\rangle'\left(\mathsf{reduce}\left(P, 1\right)\right), n\right) \\
=\ &\mathsf{step}\left([T\rangle'\left(\left\langle M \bullet V^+\right\rangle, \mathsf{reduce}\left(A_2^\circ, 1\right)\right), n\right) \\
\sqsubseteq\ &\mathsf{step}\left(\left[T \bullet_1 \left(M \bullet V^+\right)\right\rangle\left(\mathsf{reduce}\left(A_2^\circ, 1\right)\right), n\right) \\
=\ &\mathsf{step}\left(\left[T \bullet_1 \left(M \bullet V^+\right)\right\rangle A_2^\circ, n+1\right) \\
=\ &\mathsf{step}\left(\left[X^{\text{head}}\right\rangle\left\langle X^{\text{tail}} \bullet V^+\right\rangle A_2^\circ, n+2\right) \\
=\ &\mathsf{step}\left(\left[X^{\text{head}}\right\rangle\left\langle X^{\text{tail}}\right\rangle'\left(\left\langle V^+\right\rangle, A_2^\circ\right), n+3\right) \\
=\ &\mathsf{step}\left([X\rangle\left(\left\langle V^+\right\rangle, A_2^\circ\right), n+4\right) \\
=\ &\mathsf{step}\left([T\rangle' P, n+5\right)
\end{aligned}$$

where

$$X = T \bullet_1 M.$$

Consider

$$P = \left(\langle M\rangle \left\langle M_0^+\right\rangle \left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle, A_2^\circ\right).$$

$$\begin{array}{rl}
 & \mathsf{step}\left(\left[T\right\rangle' \left(\mathsf{reduce}\left(P,1\right)\right), n\right) \\
= & \mathsf{step}\left(\left[T\right\rangle' \left(\left\langle M \bullet M_0^+\right\rangle \left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle, \mathsf{reduce}\left(A_2^\circ, 1\right)\right), n\right) \\
\sqsubseteq & \mathsf{step}\left(\left[Z_s \bullet Y_s^{\mathrm{tail}} \bullet_1 V^+\right\rangle \left(\mathsf{reduce}\left(A_2^\circ, 1\right)\right), n\right) \\
= & \mathsf{step}\left(\left[Z_s \bullet Y_s^{\mathrm{tail}} \bullet_1 V^+\right\rangle A_2^\circ, n+1\right) \\
= & \mathsf{step}\left(\left[Z_s\right\rangle \left\langle Y_s^{\mathrm{tail}} \bullet_1 V^+\right\rangle A_2^\circ, n+2\right) \\
= & \mathsf{step}\left(\left[Z_r\right\rangle \left\langle Y_r^{\mathrm{tail}}\right\rangle' \left(\left\langle V^+\right\rangle, A_2^\circ\right), n+3\right) \\
\vdots & \\
= & \mathsf{step}\left(\left[Z_i\right\rangle \left\langle Y_{i+1}^{\mathrm{head}}\right\rangle \left\langle Y_{i+1}^{\mathrm{tail}}\right\rangle' \left(\begin{array}{l} \left\langle M_{i+2}^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle \\ A_2^\circ \end{array}\right), n+3r+1-3i\right) \\
= & \mathsf{step}\left(\left[Z_i\right\rangle \left\langle Y_{i+1}\right\rangle \left(\left\langle M_{i+2}^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle, A_2^\circ\right), n+3r+2-3i\right) \\
= & \mathsf{step}\left(\left[Z_i\right\rangle \left\langle Y_i^{\mathrm{tail}}\right\rangle' \left(\left\langle M_{i+1}^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle, A_2^\circ\right), n+3r+3-3i\right) \\
\vdots & \\
= & \mathsf{step}\left(\left[X^{\mathrm{head}}\right\rangle \left\langle Y_0^{\mathrm{head}}\right\rangle \left\langle Y_0^{\mathrm{tail}}\right\rangle' \left(\left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle, A_2^\circ\right), n+3r+4\right) \\
= & \mathsf{step}\left(\left[X^{\mathrm{head}}\right\rangle \left\langle Y_0\right\rangle \left(\left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle, A_2^\circ\right), n+3r+5\right) \\
= & \mathsf{step}\left(\left[X^{\mathrm{head}}\right\rangle \left\langle X^{\mathrm{tail}}\right\rangle' \left(\left\langle M_0^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle, A_2^\circ\right), n+3r+6\right) \\
= & \mathsf{step}\left(\left[X\right\rangle \left(\left\langle M_0^+\right\rangle \cdots \left\langle M_r^+\right\rangle \left\langle V^+\right\rangle, A_2^\circ\right), n+3r+7\right) \\
= & \mathsf{step}\left(\left[T\right\rangle' P, n+3r+8\right)
\end{array}$$

where

$$\begin{array}{rcl}
X & = & T \bullet_1 M \\
Y_0 & = & X^{\mathrm{tail}} \bullet_1 M_0^+ \\
Z_0 & = & X^{\mathrm{head}} \bullet Y_0^{\mathrm{head}} \\
Y_{n+1} & = & Y_n^{\mathrm{tail}} \bullet_1 M_{n+1}^+ \\
Z_{n+1} & = & Z_n \bullet Y_{n+1}^{\mathrm{head}}.
\end{array}$$

Consider

$$P = \left(\langle M\rangle \left\langle M_0^+\right\rangle \left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ, A_2^\circ\right).$$

$$\begin{array}{rl}
 & \mathsf{step}\left(\left[T\right\rangle' \left(\mathsf{reduce}\left(P,1\right)\right), n\right) \\
= & \mathsf{step}\left(\left[T\right\rangle' \left(\left\langle M \bullet M_0^+\right\rangle \left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ, \mathsf{reduce}\left(A_2^\circ, 1\right)\right), n\right)
\end{array}$$

$$\begin{array}{cl}
\sqsubseteq & \mathsf{step}\left([Z_r\rangle \left\langle Y_r^{\text{tail}}\right\rangle' (\mathsf{reduce}(A_1^\circ, 1), \mathsf{reduce}(A_2^\circ, 1)), n\right) \\
= & \mathsf{step}\left([Z_r\rangle \left\langle Y_r^{\text{tail}}\right\rangle' (A_1^\circ, A_2^\circ), n+1\right) \\
\vdots & \\
= & \mathsf{step}\left([Z_i\rangle \left\langle Y_{i+1}^{\text{head}}\right\rangle \left\langle Y_{i+1}^{\text{tail}}\right\rangle' \begin{pmatrix} \left\langle M_{i+2}^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ \\ A_2^\circ \end{pmatrix}, n+3r-1-3i\right) \\
= & \mathsf{step}\left([Z_i\rangle \langle Y_{i+1}\rangle \left(\left\langle M_{i+2}^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ, A_2^\circ\right), n+3r-3i\right) \\
= & \mathsf{step}\left([Z_i\rangle \left\langle Y_i^{\text{tail}}\right\rangle' \left(\left\langle M_{i+1}^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ, A_2^\circ\right), n+3r+1-3i\right) \\
\vdots & \\
= & \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle Y_0^{\text{head}}\right\rangle \left\langle Y_0^{\text{tail}}\right\rangle' \left(\left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ, A_2^\circ\right), n+3r+2\right) \\
= & \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \langle Y_0\rangle \left(\left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ, A_2^\circ\right), n+3r+3\right) \\
= & \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle X^{\text{tail}}\right\rangle' \left(\left\langle M_0^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ, A_2^\circ\right), n+3r+4\right) \\
= & \mathsf{step}\left([X\rangle \left(\left\langle M_0^+\right\rangle \left\langle M_1^+\right\rangle \cdots \left\langle M_r^+\right\rangle A_1^\circ, A_2^\circ\right), n+3r+5\right) \\
= & \mathsf{step}\left([T\rangle' P, n+3r+6\right)
\end{array}$$

where

$$\begin{array}{rcl}
X & = & T \bullet_1 M \\
Y_0 & = & X^{\text{tail}} \bullet_1 M_0^+ \\
Z_0 & = & X^{\text{head}} \bullet Y_0^{\text{head}} \\
Y_{n+1} & = & Y_n^{\text{tail}} \bullet_1 M_{n+1}^+ \\
Z_{n+1} & = & Z_n \bullet Y_{n+1}^{\text{head}}.
\end{array}$$

Consider $P = (\langle M\rangle A_1^\circ, A_2^\circ)$.

$$\begin{array}{cl}
 & \mathsf{step}\left([T\rangle' (\mathsf{reduce}(P, 1)), n\right) \\
= & \mathsf{step}\left([T\rangle' (\langle M\rangle (\mathsf{reduce}(A_1^\circ, 1)), \mathsf{reduce}(A_2^\circ, 1)), n\right) \\
\sqsubseteq & \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle X^{\text{tail}}\right\rangle' (\mathsf{reduce}(A_1^\circ, 1), \mathsf{reduce}(A_2^\circ, 1)), n\right) \\
= & \mathsf{step}\left(\left[X^{\text{head}}\right\rangle \left\langle X^{\text{tail}}\right\rangle' (A_1^\circ, A_2^\circ), n+1\right) \\
= & \mathsf{step}\left([X\rangle (A_1^\circ, A_2^\circ), n+2\right) \\
= & \mathsf{step}\left([T\rangle' P, n+3\right)
\end{array}$$

where

$$X = T \bullet_1 M.$$

Consider $\langle T\rangle'$. Similar to $[T\rangle'$. ■

Proof of Lemma 101:
Consider $[T]$. Consider $m \in \mathbb{N}$. The proof is by induction on $n \in \mathbb{N}$.
Basis:

$$\mathsf{step}\left([T]\left(\mathsf{reduce}\left(P,0\right)\right),m\right) = \mathsf{step}\left([T]\,P,m\right).$$

Inductive Step: Assume there exists $i \in \mathbb{N}$ such that

$$\mathsf{step}\left([T]\left(\mathsf{reduce}\left(\mathsf{reduce}\left(P,1\right),n\right)\right),m\right) \sqsubseteq \mathsf{step}\left([T]\left(\mathsf{reduce}\left(P,1\right)\right),i\right).$$

Using Lemma 100, there exists $r \in \mathbb{N}$

$$\mathsf{step}\left([T]\left(\mathsf{reduce}\left(P,1\right)\right),i\right) \sqsubseteq \mathsf{step}\left([T]\,P,r\right).$$

Therefore

$$\begin{aligned}\mathsf{step}\left([T]\left(\mathsf{reduce}\left(P,n+1\right)\right),m\right) &= \mathsf{step}\left([T]\left(\mathsf{reduce}\left(\mathsf{reduce}\left(P,1\right),n\right)\right),m\right)\\ &\sqsubseteq \mathsf{step}\left([T]\left(\mathsf{reduce}\left(P,1\right)\right),i\right)\\ &\sqsubseteq \mathsf{step}\left([T]\,P,r\right).\end{aligned}$$

Similarly for $[T\rangle'$, $\langle T\rangle'$. ■

Proof of Lemma 102:
Consider $[T]$.

- Consider $P = B^{\circ}$.

$$\begin{aligned} &= [\![[T]]\!]\left(\mathsf{value}\left(P\right)\right)\\ &= \mathbb{R}^{\infty}\\ &= \mathsf{step}\left([T]\,P,0\right).\end{aligned}$$

- Consider $P = \left([V\rangle, [W\rangle\right)$.

$$\begin{aligned} &[\![[T]]\!]\left(\mathsf{value}\left(P\right)\right)\\ =\;& [\![[T]]\!]\left(\mathsf{value}\left([V\rangle, [W\rangle\right)\right)\\ =\;& [\![T]\!]\left(\mathsf{info}\left(V\right),\mathsf{info}\left(W\right)\right)\\ =\;& \mathsf{info}\left(T \bullet_1 V \bullet W\right)\\ =\;& \mathsf{value}\left([T \bullet_1 V \bullet W\rangle\right)\\ =\;& \mathsf{step}\left([T]\left([V\rangle, [W\rangle\right),1\right)\\ =\;& \mathsf{step}\left([T]\,P,1\right).\end{aligned}$$

- Consider $P = ([V\rangle, [N\rangle R)$.

$$\begin{aligned} & [\![[T]]\!] (\mathsf{value}(P)) \\ = \; & [\![[T]]\!] (\mathsf{value}([V\rangle, [N\rangle R)) \\ = \; & [\![T]\!] (\mathsf{info}(V), \mathsf{info}(N)) \\ = \; & \mathsf{info}(T \bullet_1 V \bullet N) \\ = \; & \mathsf{value}([T \bullet_1 V \bullet N\rangle R) \\ = \; & \mathsf{step}([T]([V\rangle, [N\rangle R), 1) \\ = \; & \mathsf{step}([T] P, 1). \end{aligned}$$

- Consider $P = ([M\rangle Q, [W\rangle)$. Similar to $P = ([V\rangle, [N\rangle R)$.
- Consider $P = ([M\rangle Q, [N\rangle R)$.
  - Consider $\mathsf{info}(T \bullet_1 M \bullet_2 N) = \mathbb{R}^\infty$.

$$\begin{aligned} & [\![[T]]\!] (\mathsf{value}(P)) \\ = \; & [\![[T]]\!] (\mathsf{value}([M\rangle Q, [N\rangle R)) \\ = \; & [\![T]\!] (\mathsf{info}(M), \mathsf{info}(N)) \\ = \; & \mathsf{info}(T \bullet_1 M \bullet_2 N) \\ = \; & \mathbb{R}^\infty \\ = \; & \mathsf{step}([T] P, 0). \end{aligned}$$

  - Consider $\mathsf{info}(T \bullet_1 M \bullet_2 N) \neq \mathbb{R}^\infty$.

$$\begin{aligned} & [\![[T]]\!] (\mathsf{value}(P)) \\ = \; & [\![[T]]\!] (\mathsf{value}([M\rangle Q, [N\rangle R)) \\ = \; & [\![T]\!] (\mathsf{info}(M), \mathsf{info}(N)) \\ = \; & \mathsf{info}(T \bullet_1 M \bullet_2 N) \\ = \; & \mathsf{value}\left(\left[(T \bullet_1 M \bullet_2 N)^{\mathrm{head}}\right\rangle \left(\left[(T \bullet_1 M \bullet_2 N)^{\mathrm{tail}}\right\rangle (Q, R)\right)\right) \\ = \; & \mathsf{step}([T \bullet_1 M \bullet_2 N\rangle (Q, R), 1) \\ = \; & \mathsf{step}([T]([M\rangle Q, [N\rangle R), 2) \\ = \; & \mathsf{step}([T] P, 2). \end{aligned}$$

Similarly for $[T\rangle'$, $\langle T\rangle'$. ■

Proof of Lemma 103:

Consider $[T]$ and $n \in \mathbb{N}$. Using Lemma 102, there exists $i \in \mathbb{N}$ such that

$$[\![[T]]\!] \left(\mathsf{step}\left(P, n\right)\right) = \mathsf{step}\left(\left[T\right]\left(\mathsf{reduce}\left(P, n\right)\right), i\right).$$

Using Lemma 101, there exists $m \in \mathbb{N}$ such that

$$\mathsf{step}\left(\left[T\right], \left(\mathsf{reduce}\left(P, n\right)\right), i\right) \sqsubseteq \mathsf{step}\left(\left[T\right] P, m\right).$$

Therefore

$$\begin{array}{rcl} [\![[T]]\!] \left(\mathsf{step}\left(P, n\right)\right) & = & \mathsf{step}\left(\left[T\right]\left(\mathsf{reduce}\left(P, n\right)\right), i\right) \\ & \sqsubseteq & \mathsf{step}\left(\left[T\right] P, m\right). \end{array}$$

Similarly for $[T\rangle'$, $\langle T\rangle'$. ■

Proof of Lemma 104:

- Consider $[P\rangle : \mathsf{real}^{\infty}$. The function $[\![\lambda x \cdot [x\rangle]\!]$ is continuous. So, we only need to show that

  $$\forall n \in \mathbb{N} \cdot \exists m \in \mathbb{N} \cdot [\![\lambda x \cdot [x\rangle]\!] \left(\mathsf{step}\left(P, n\right)\right) \sqsubseteq \mathsf{step}\left(\left(\lambda x \cdot [x\rangle\right) P, m\right)$$

  for every closed computable term P. So, let P be a closed computable term and consider $n \in \mathbb{N}$. Let

  $$i = \mu n \cdot \mathsf{reduce}\left(P, n\right) = X.$$

  - Consider $i > n$.

    $$\begin{array}{rcl} [\![\lambda x \cdot [x\rangle]\!] \left(\mathsf{step}\left(P, n\right)\right) & = & \mathbb{R}^{\infty} \\ & = & \mathsf{step}\left(\left(\lambda x \cdot [x\rangle\right) P, 0\right). \end{array}$$

  - Consider $i \leq n$.

    $$\begin{array}{rcl} [\![\lambda x \cdot [x\rangle]\!] \left(\mathsf{step}\left(P, n\right)\right) & = & [\![\lambda x \cdot [x\rangle]\!] \left(\mathsf{value}\left(X\right)\right) \\ & = & \mathsf{value}\left([X\rangle\right) \\ & = & \mathsf{step}\left([P\rangle, i\right) \\ & = & \mathsf{step}\left(\left(\lambda x \cdot [x\rangle\right) P, i+1\right). \end{array}$$

- Consider $\langle P\rangle : \mathsf{real}^{+}$. Similar to $[P\rangle : \mathsf{real}^{\infty}$.

- Consider $[P]\,Q : \mathsf{real}^\infty$. The function $[\![\lambda x \cdot [x]]\!]$ is continuous because Möbius transformations are continuous on the extended real line. So, we only need to show that
$$\forall n, m \in \mathbb{N} \cdot \exists r \in \mathbb{N} \cdot$$
$$[\![\lambda x \cdot [x\rangle]\!]\,(\mathsf{step}\,(P,n))\,(\mathsf{step}\,(Q,m)) \sqsubseteq \mathsf{step}\,((\lambda x \cdot [x\rangle)\,PQ, r)$$
for all closed computable terms P and Q. So, let P and Q be closed computable terms and consider $n, m \in \mathbb{N}$. Let
$$i = \mu n \cdot \mathsf{reduce}\,(P,n) = X.$$
  - Consider $i > n$.
$$\begin{aligned} & [\![\lambda x \cdot [x\rangle]\!]\,(\mathsf{step}\,(P,n))\,(\mathsf{step}\,(Q,m)) \\ = \; & [\![\lambda x \cdot [x\rangle]\!]\,(\bot)\,(\mathsf{step}\,(Q,m)) \\ = \; & \mathbb{R}^\infty \\ = \; & \mathsf{step}\,((\lambda x \cdot [x\rangle)\,PQ, 0)\,. \end{aligned}$$
  - Consider $i \leq n$. Using Lemma 99 and Lemma 103, there exists $r \in \mathbb{N}$ such that
$$[\![[X]]\!]\,(\mathsf{step}\,(Q,m)) \sqsubseteq \mathsf{step}\,([X]\,Q, r-1-i)\,.$$
Therefore
$$\begin{aligned} & [\![\lambda x \cdot [x]]\!]\,(\mathsf{step}\,(P,n))\,(\mathsf{step}\,(Q,m)) \\ = \; & [\![\lambda x \cdot [x]]\!]\,(\mathsf{value}\,(X))\,(\mathsf{step}\,(Q,m)) \\ = \; & [\![[X]]\!]\,(\mathsf{step}\,(Q,m)) \\ \sqsubseteq \; & \mathsf{step}\,([X]\,Q, r-1-i) \\ = \; & \mathsf{step}\,([\mathsf{reduce}\,(P,i)\rangle\,Q, r-1-i) \\ = \; & \mathsf{step}\,([P\rangle\,Q, r-1) \\ = \; & \mathsf{step}\,((\lambda x \cdot [x\rangle)\,PQ, r)\,. \end{aligned}$$
- Consider $[P\rangle\,Q : \mathsf{real}^+$. Similar to $[P]\,Q : \mathsf{real}^\infty$.
- Consider $\langle P\rangle\,Q : \mathsf{real}^+$. Similar to $[P]\,Q : \mathsf{real}^\infty$. ■

Proof of Lemma 105:

The proof is by structural induction on P. Let σ be a substitution of closed computable terms for the free variables in P. Thus, we must show that σP is computable.

- For $P = x$, 0, true, false, $\mathsf{succ}(Q)$, $\mathsf{pred}(Q)$, $\mathsf{zero}(Q)$, $\mathsf{if}\,Q\,\mathsf{then}\,R\,\mathsf{else}\,S$ or QR, P is computable because it is in PCF.

- For $P = \lambda x.Q$, we must show that $R = (\lambda x.\sigma Q)P_1P_2\ldots P_n$ is computable if $P_1, P_2, \ldots, P_n$ are closed computable terms and R has ground type t.

  - For $t \equiv \mathsf{num}$ or bool, R is computable because it is in PCF.
  - For $t \equiv \mathsf{real}^{\infty}$, real^{+} or $r \times s$, observe that

    $$R \to ([P_1/x](\sigma Q))P_2P_3\ldots P_n = ((\sigma[P_1/x])Q)P_2P_3\ldots P_n,$$

    which we will call S. But, S is computable because Q is computable. Therefore, $[\![R]\!] \sqsubseteq \mathsf{eval}\,(R)$ because $[\![R]\!] = [\![S]\!]$, $\mathsf{eval}\,(R) = \mathsf{eval}\,(S)$ and $[\![S]\!] \sqsubseteq \mathsf{eval}\,(S)$.

- For $P \equiv \mu x.Q$, we must show that $R \equiv SP_1P_2\ldots P_n$ where $S \equiv \mu x.\sigma Q$ is computable if $P_1, P_2, \ldots, P_n$ are closed computable terms and R has ground type t.

  - For $t \equiv \mathsf{num}$ or bool, R is computable because it is in PCF.
  - For $t \equiv \mathsf{real}^{\infty}$, real^{+} or $r \times s$ define S^n by

    $$\begin{array}{rcl} S^0 & \equiv & \mu x.x \\ S^{n+1} & \equiv & (\lambda x.\sigma Q)\,S^n. \end{array}$$

    It is easy to show by induction on n that $[\![S]\!] = \bigsqcup_{n=0}^{\infty}[\![S^n]\!]$. Clearly, S^n is computable for all $n \in \mathbb{N}$. Let $a \ll [\![R]\!]$.

    $$\begin{array}{rcl} [\![R]\!] & = & [\![S]\!][\![P_1]\!][\![P_2]\!]\ldots[\![P_n]\!] \\ & = & \left(\bigsqcup_{n=0}^{\infty}[\![S^n]\!]\right)[\![P_1]\!][\![P_2]\!]\ldots[\![P_n]\!] \\ & = & \bigsqcup_{n=0}^{\infty}([\![S^n]\!][\![P_1]\!][\![P_2]\!]\ldots[\![P_n]\!]) \\ & = & \bigsqcup_{n=0}^{\infty}[\![S^nP_1P_2\ldots P_n]\!]. \end{array}$$

    However, $\exists n \in \mathbb{N}$ such that $a \ll [\![S^nP_1P_2\ldots P_n]\!]$, therefore $\exists s_n \in \mathbb{N}$ such that $a \sqsubseteq \mathsf{step}\,(S^nP_1P_2\ldots P_n, s_n)$ because S^n is computable. Therefore $\exists r_n \in \mathbb{N}$ such that $\mathsf{step}\,(R, r_n) = \mathsf{step}\,(S^nP_1P_2\ldots P_n, s_n)$ by a straightforward extension of the Unwinding Theorem [29].

– For $P \equiv (Q, R)$. Proved in Lemma 93.
– For $P \equiv \mathsf{fst}(Q)$ or $\mathsf{snd}(Q)$. Proved in Lemma 94.
– For $P \equiv Q \lhd R$. Proved in Lemma 95.
– For $P \equiv [Q]$, $[Q\rangle$ or $\langle Q\rangle$. Proved in Lemma 104. ■

Proof of Lemma 107:
The proof is by induction on the definition of $\rightarrow$. The rules $P \rightarrow Q$ with $\mathsf{pvalue}\,(P) = \{\bot\}$ are trivial. Only the parallel rules need to be considered because we have already shown that the other rules are sound.
Basis:

- Consider $P\,\{Q, R\} \rightarrow \{PQ, PR\}$.

$$\begin{aligned} & \mathsf{pvalue}\,(P\,\{Q,R\} : \mathsf{real}^{\sigma}) \\ = & \begin{cases} \downarrow \{\mathsf{info}\,(M)\} & \text{if } P = [M\rangle \\ \downarrow \{\mathsf{info}\,(M)\} & \text{if } P = \langle M\rangle \\ \{\bot\} & \text{otherwise} \end{cases} \\ = & \ \mathsf{pvalue}\,(PQ) \sqcup \mathsf{pvalue}\,(PR) \\ = & \ \mathsf{pvalue}\,(\{PQ, PR\} : \mathsf{real}^{\sigma})\,. \end{aligned}$$

- Consider $(\{P, Q\}, R) \rightarrow \{(P, R), (Q, R)\}$.

$$\begin{aligned} & \mathsf{pvalue}\,((\{P,Q\}, R)) \\ = & \ \mathsf{pvalue}\,(\{P,Q\}) \times \mathsf{pvalue}\,(R) \\ = & \ (\mathsf{pvalue}\,(P) \sqcup \mathsf{pvalue}\,(Q)\,, \mathsf{pvalue}\,(R)) \\ = & \ (\mathsf{pvalue}\,(P)\,, \mathsf{pvalue}\,(R)) \sqcup (\mathsf{pvalue}\,(Q)\,, \mathsf{pvalue}\,(R)) \\ = & \ \mathsf{pvalue}\,((P,R)) \sqcup \mathsf{pvalue}\,((Q,R)) \\ = & \ \mathsf{pvalue}\,(\{(P,R)\,, (Q,R)\})\,. \end{aligned}$$

- Consider $(P, \{Q, R\}) \rightarrow \{(P, Q), (P, R)\}$. Similar to $(\{P, Q\}, R) \rightarrow \{(P, R), (Q, R)\}$.

Inductive Step:

- Consider $\dfrac{P \rightarrow R \quad Q \rightarrow S}{\{P, Q\} \rightarrow \{R, S\}}$.

Assume $\mathsf{pvalue}(P) \sqsubseteq \mathsf{pvalue}(R)$ and $\mathsf{pvalue}(Q) \sqsubseteq \mathsf{pvalue}(S)$.

$$\begin{aligned}
& \mathsf{pvalue}(\{P, Q\}) \\
= \; & \mathsf{pvalue}(P) \sqcup \mathsf{pvalue}(Q) \\
\sqsubseteq \; & \mathsf{pvalue}(R) \sqcup \mathsf{pvalue}(S) \\
= \; & \mathsf{pvalue}(\{R, S\}) \,. \; \blacksquare
\end{aligned}$$

Proof of Lemma 112:

Assume P and Q are computable and consider $\{P, Q\}$. The function $(\!|\lambda x \cdot \lambda y \cdot \{x, y\}\,|\!)$ is clearly continuous. So, we only need to show that

$$\forall n, m \in \mathbb{N} \cdot \exists r \in \mathbb{N} \cdot$$

$$(\!|\lambda x \cdot \lambda y \cdot \{x, y\}\,|\!)\,(\mathsf{pstep}\,(P, n))\,(\mathsf{pstep}\,(Q, m)) \sqsubseteq \mathsf{pstep}\,((P, Q)\,, r)$$

for any closed computable terms P and Q. So, let P and Q be closed computable terms and consider $n, m \in \mathbb{N}$.

$$\begin{aligned}
& (\!|\lambda x \cdot \lambda y \cdot \{x, y\}\,|\!)\,(\mathsf{pstep}\,(P, n))\,(\mathsf{pstep}\,(Q, m)) \\
\sqsubseteq \; & (\!|\lambda x \cdot \lambda y \cdot \{x, y\}\,|\!)\,(\mathsf{pstep}\,(P, \mathsf{max}\,(n, m)))\,(\mathsf{pstep}\,(Q, \mathsf{max}\,(n, m))) \\
= \; & (\mathsf{pstep}\,(P, \mathsf{max}\,(n, m))) \sqcup (\mathsf{pstep}\,(Q, \mathsf{max}\,(n, m))) \\
= \; & \mathsf{pvalue}\,(\{\mathsf{reduce}\,(P, \mathsf{max}\,(n, m))\,, \mathsf{reduce}\,(Q, \mathsf{max}\,(n, m))\}) \\
= \; & \mathsf{pstep}\,(\{P, Q\}\,, \mathsf{max}\,(n, m))\,. \; \blacksquare
\end{aligned}$$

Appendix B

List of Notation

$f : X \to Y$	total function
$f : X \rightsquigarrow Y$	partial function
$\perp$	bottom
$\mathcal{P}(X)$	powerset
$\mathcal{P}_f(X)$	set of finite subsets
$\mathcal{P}^*(X)$	set of non-empty subsets
$\mapsto$	barred arrow notation
$\mathbb{N}(n)$	$\mathbb{N}(n) = \{0, 1, 2, \ldots, n-1\}$
$\mathbb{Z}(n)$	$\mathbb{Z}(n) = \{1-n, 2-n, \ldots, 0, 1, 2, \ldots, n-2, n-1\}$
$\langle x_n \rangle_{n=0}^{\infty}$	infinite sequence $x_0, x_1, x_2, \ldots$
$\langle x_n \rangle_{n=0}^{m}$	finite sequence $x_0, x_1, x_2, \ldots, x_m$
$\|G\|$	order of a group
Ha	right coset
aH	left coset
$N \trianglelefteq G$	normal subgroup N of G
G/N	quotient group of G by the normal subgroup N
$\mathsf{image}(\theta)$	image of homomorphism θ
$\mathsf{kernel}(\theta)$	kernel of homomorphism θ
$x^{-1}gx$	conjugate of g by x
$GL(n, F)$	n-dimensional general linear group over F
F^*	multiplicative group of non-zero elements
$[a, b]$	closed bounded interval in $\mathbb{R}$
$\mathbb{I}X$	set of all intervals of X
$\mathbb{I}^{\mathrm{O}}X$	set of open intervals of X
$\mathbb{I}^{\mathrm{C}}X$	set of closed intervals of X
$\mathbb{I}^{\mathrm{F}}X$	set of compact intervals of X
$\mathsf{width}([a, b])$	$\mathsf{width}([a, b]) = b - a$
$\underline{[a, b]}$	$\underline{[a, b]} = a$

$\overline{[a,b]}$	$\overline{[a,b]} = b$
$\mathsf{ob}(\mathcal{C})$	objects of category $\mathcal{C}$
$\mathsf{mor}(\mathcal{C})$	morphisms of category $\mathcal{C}$
$\mathsf{src}\,(f)$	source of morphism f
$\mathsf{tar}\,(f)$	target of morphism f
$f \circ g$	function composition
$\sqsubseteq$	information ordering
$\sqcup A$	least upper bound
$\sqcap A$	greatest lower bound
$[D \to E]$	function space
$x \ll y$	x approximates y
$K\,(D)$	set of compact elements of dcpo D
$\uparrow x$	$\uparrow x = \{y \in D \mid x \sqsubseteq y\}$
$\downarrow x$	$\downarrow x = \{y \in D \mid y \sqsubseteq x\}$
$\twoheaduparrow x$	$\twoheaduparrow x = \{y \in D \mid x \ll y\}$
$\twoheaddownarrow x$	$\twoheaddownarrow x = \{y \in D \mid y \ll x\}$
$d \searrow e$	step function
$\mathrm{UPPER}\,(D)$	upper powerdomain
$\mathrm{LOWER}\,(D)$	lower powerdomain
$\mathrm{CONVEX}\,(D)$	convex powerdomain
$\mathbb{N}$	set of natural numbers
$\mathbb{Z}$	set of integers
$\mathbb{Q}$	set of rational numbers
$\mathbb{R}$	set of real numbers
$d_{\mathrm{E}}\,(x,y)$	Euclidean distance
$d_{\mathrm{C}}\,(x,y)$	chordal distance
$d_{\mathrm{P}}\,(x,y)$	alternative distance
$\lfloor x \rfloor$	floor of x
$\lceil x \rceil$	ceiling of x
$\lfloor x \rceil$	round of x
$\lfloor\lfloor x \rceil\rceil$	Euclidean part of x
$\lfloor\lfloor x \rfloor\rfloor$	basement of x
$\mathbb{F}^{\infty}$	$\mathbb{F}^{\infty} = \mathbb{F} \cup \{\infty\}$
$\mathbb{R}^{\infty}$	extended real line
$\mathbb{C}^{\infty}$	extended complex plane
$\mathbb{I}\,[\mathbb{F}]$	all intervals in $\mathbb{R}^{\infty}$ with end-points in $\mathbb{F}$
$\mathbb{I}^{\mathrm{O}}\,[\mathbb{F}]$	open intervals in $\mathbb{R}^{\infty}$ with end-points in $\mathbb{F}$
$\mathbb{I}^{\mathrm{C}}\,[\mathbb{F}]$	closed intervals in $\mathbb{R}^{\infty}$ with end-points in $\mathbb{F}$
$\mathbb{I}^{\mathrm{F}}\,[\mathbb{F}]$	compact intervals in $\mathbb{R}^{\infty}$ with end-points in $\mathbb{F}$
$H\,(O)$	set of functions holomorphic in O

$\overline{(B,\prec)}$	ideal completion of abstract basis
$\emptyset$	empty set
$D(c;r)$	open disc with centre c and radius r
$A(c;r,s)$	open annulus centre c and radii r and s
$(d_{m-1}\cdots d_0.d_{-1}d_{-2}\cdots d_n)_r$	radix r positional notation
$\mathbb{A}$	set of algebraic numbers
$\mathbb{B}(b)$	set of b-adic numbers
$\mathbb{Q}\left(\sqrt{d}\right)$	quadratic field for d
$\eta(I)$	$\eta(I)=\bigcap_{n=0}^{\infty} I_n$
$\mathbb{C}(I)$	continuous real domain
$\mathbb{A}(\mathbb{F})$	algebraic real domain
$\underline{x}$	$\underline{x}=\mathsf{inf}(x)$
$\overline{x}$	$\overline{x}=\mathsf{sup}(x)$
$\langle x\rangle$	$\langle x\rangle=\{y\in\mathbb{I}^{\mathrm{C}}[\mathbb{F}] \mid \underline{y}\le\underline{x}$ and $\overline{x}\le\overline{y}\}$
$\langle x\rangle\rangle$	$\langle x\rangle\rangle=\{y\in\mathbb{I}^{\mathrm{C}}[\mathbb{F}] \mid \underline{y}\le\underline{x}$ and $\overline{x}<\overline{y}\}$
$\langle\langle x\rangle$	$\langle\langle x\rangle=\{y\in\mathbb{I}^{\mathrm{C}}[\mathbb{F}] \mid \underline{y}<\underline{x}$ and $\overline{x}\le\overline{y}\}$
$\langle\langle x\rangle\rangle$	$\langle\langle x\rangle\rangle=\{y\in\mathbb{I}^{\mathrm{C}}[\mathbb{F}] \mid \underline{y}<\underline{x}$ and $\overline{x}<\overline{y}\}$
$[a_0.b_0,a_1.b_1,a_2.b_2,\ldots]$	continued fraction abbreviation
$[a_0,a_1,a_2,\ldots]$	simple continued fraction abbreviation
ϕ	golden ratio
e	natural number
π	pi
$\xi_{\mathbb{S}}$	continuation function
$[L/M]$	L, M Padé Approximant
$C(L/M)$	C table entry
$(a)_n$	Pochhammer symbol
${}_2F_1(a,b;c;z)$	ordinary hypergeometric function
${}_1F_1(b;c;x)$	Kummer confluent hypergeometric function
${}_nF_m$	n-m confluent hypergeometric function
$\mathsf{erf}(x)$	error function
$\mathsf{erfc}(x)$	complementary error function
$\gamma(a,x)$	incomplete gamma function
$J_a(x)$	Bessel function of the first kind
div	integer division
mod	integer remainder
rif	redundant if operator
pif	parallel if operator
$<_\epsilon$	quasi-relational comparison
$\mathsf{interior}(I)$	interior of I
$\mathsf{det}(M)$	determinant of matrix M

$\mathbb{V}$	set of vectors with integer coefficients
$\mathbb{M}$	set of matrices with integer coefficients
$\mathbb{T}$	set of tensors with integer coefficients
$\mathbb{V}^*$	set of vectors with natural coefficients
$\mathbb{M}^*$	set of matrices with natural coefficients
$\mathbb{T}^*$	set of tensors with natural coefficients
$\mathbb{V}^+$	set of unsigned vectors
$\mathbb{M}^+$	set of unsigned matrices
$\mathbb{T}^+$	set of unsigned tensors
V	V denotes vector
W	W denotes vector
M	M denotes matrix
N	N denotes matrix
O	O denotes matrix
T	T denotes tensor
U	U denotes tensor
$(X,Y)_0$	first projection
$(X,Y)_1$	second projection
L^{T}	transpose
$\bullet$	dot product
$\bullet_n$	general dot product
$\bullet_1$	left product
$\bullet_2$	right product
$\Phi(V)$	interpretation morphism of vector V
$\Psi(M)$	interpretation morphism of matrix M
$\Upsilon(T)$	interpretation morphism of tensor T
$\sigma(V)$	sign of vector V
$\mathcal{M}(\mathbb{F})$	group of Möbius transformations
$M^\dagger$	tame inverse of matrix M
$\mathsf{trace}(M)$	trace of matrix M
$\chi(\lambda, M)$	characteristic polynomial of matrix M
$\mathsf{invariance}(M)$	invariance of matrix M
$\mathsf{order}(M)$	order of matrix M
$\mathsf{gcd}(m,n)$	greatest common divisor of m and n
S_σ	sign matrix with $\sigma \in \{\infty, +, 0, -\}$
${}^bD_d$	radix b digit matrix with $d \in \mathbb{Z}(b)$
D_d	radix 2 digit matrix with $d \in \{-1, 0, 1\}$
$\mathbb{L}$	$\mathbb{L} = \mathbb{V} \cup \mathbb{M} \cup \mathbb{T}$
$\mathbb{L}^*$	$\mathbb{L}^* = \mathbb{V}^* \cup \mathbb{M}^* \cup \mathbb{T}^*$
$\mathbb{L}^+$	$\mathbb{L}^+ = \mathbb{V}^+ \cup \mathbb{M}^+ \cup \mathbb{T}^+$

$\mathsf{info}\,(L)$	information in lft L
T_i	basic arithmetic tensor $i \in \{+, -, \times, \div\}$
T^{head}	head of tensor T
T^{tail}	tail of tensor T
${}^b\mathfrak{D}_c^n$	${}^b\mathfrak{D}_c^n = {}^bD_{d_1}{}^bD_{d_2}\dots{}^bD_{d_n}$
$\mathfrak{D}_c^n$	$\mathfrak{D}_c^n = {}^2\mathfrak{D}_c^n$
$L\,\{E_1, \dots, E_N\}$	expression tree with root node L
$M\,[E_1]$	matrix square bracket application
$T\,[E_1, E_2]$	tensor square bracket application
$\overline{\Xi}\,(M)$	contractivity of matrix M
$\underline{\Xi}(M)$	expansivity of matrix M
$\underline{\delta}\,(M, \epsilon)$	underestimate
$\overline{\delta}\,(M, \epsilon)$	overestimate
$\mathsf{strategy}_{\mathrm{f}}\,(T, n)$	fair strategy
$\mathsf{strategy}_{\mathrm{o}}\,(T, n)$	information overlap strategy
$\mathsf{strategy}_{\mathrm{m}}\,(T, n)$	outcome minimization strategy
Δ_d	decision function
sem	sign emission function
dem	digit emission function
ab	absorption function
edem	efficient digit emission function
eab	efficient absorption function
$\#\,(a)$	number of storage bits required
$\Delta_1^-\,(L, \epsilon)$	conservative approximation
$\lambda x \cdot P$	abstraction
$\mu x \cdot P$	recursion
$[Q/x]P$	substitution
$x_1 : t_1, \dots, x_n : t_n$	type assignment
$H \vdash P : t$	typing judgement
$\longrightarrow$	one-step reduction relation
$\longrightarrow^*$	reduction relation
$[\![H \rhd P : t]\!]$	standard model interpretation
$(\!|H \rhd P : t|\!)$	parallel model interpretation
$[P]$	transformation construct
$[P\rangle$	transformation construct
$\langle P\rangle$	transformation construct
$\{P, P\}$	parallel construct
$P \lhd P$	squeeze construct
$\widehat{d}$	standard model to parallel model mapping

Bibliography

[1] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3, chapter 1, pages 1–168. Clarendon Press, 1994.

[2] A. Avizienis. Signed-digit number representations for fast parallel arithmetic. *IRE Transactions on electronic computers*, (10):389–400, 1961.

[3] A. Avizienis. Binary-compatible signed-digit arithmetic. In *AFIPS Conference Proceedings*, pages 663–672, 1964.

[4] G. A. Baker. *Essentials of Padé approximants.* Academic Press, 1975.

[5] H. J. Boehm and R. Cartwright. Exact real arithmetic: formulating real numbers as functions. In D. A. Turner, editor, *Research topics in functional programming.* Almqvist and Wiksell, 1990. The University of Texas Year of Programming Series.

[6] H. J. Boehm, R. Cartwright, M. J. O'Donnel, and M. Riggle. Exact real arithmetic: A case study in higher order programming. In *Proceedings of the 1986 ACM conference on Lisp and Functional Programming.* ACM, 1986.

[7] C. Brezinski. *History of continued fractions and Padé approximants*, volume 12 of *Springer series in Computational mathematics.* Springer Verlag, 1991.

[8] A. Church. An unsolvable problem in elementary number theory. *American Journal of Mathematics*, 58:345–363, 1936.

[9] P. M. Cohn. *Algebra*, volume 1. John Wiley & Sons, second edition, 1984.

[10] P. Di Gianantonio. *A functional approach to computability on real numbers.* PhD thesis No. TD-6/93, University of Pisa-Genova-Udine, 1993.

[11] P. Di Gianantonio. Real number computability and domain theory. In *Proceedings of the* 18th *International Symposium on Mathematical Foundations of Computer Science*, pages 413–422, Gdansk, Poland, September 1993. LNCS 711.

[12] P. Di Gianantonio. A golden ratio notation for the real numbers. Technical report, CWI Amsterdam, 1996.

[13] P. Di Gianantonio. Real number computability and domain theory. *Information and Computation*, 127(1):12–25, May 1996.

[14] A. Edalat. Domain theory and integration. *Theoretical Computer Science*, 151:163–193, 1995.

[15] A. Edalat. Dynamical systems, measures and fractals via domain theory. *Information and Computation*, 120(1):32–48, July 1995.

[16] A. Edalat. Power domains and iterated function systems. *Information and Computation*, 124:182–197, 1996.

[17] A. Edalat. Domains for computation in mathemetics, physics and exact real arithmetic. *Bulletin of Symbolic Logic*, 3(4):401–452, 1997.

[18] A. Edalat and M. H. Escardó. Integration in real PCF. In *Logic in Computer Science*. IEEE Computer Society Press, 1996.

[19] A. Edalat and R. Heckmann. A computational model for metric spaces. *Theoretical Computer Science*, (193):53–73, 1998.

[20] A. Edalat and P. J. Potts. A new representation for exact real numbers. In *Proceedings of Mathematical Foundations of Programming Semantics 13*, volume 6 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science B. V., 1997. Available from http://www.elsevier.nl/locate/entcs/volume6.html.

[21] A. Edalat, P. J. Potts, and P. Sünderhauf. Lazy computation with exact real numbers. In *ACM SIGPLAN International Conference on Functional Programming*, 1998. To appear.

[22] A. Edalat and P. Sünderhauf. A domain theoretic approach to computability on the real line. *Theoretical Computer Science*, 1997. To appear, available from `http://theory.doc.ic.ac.uk/people/Edalat/reals.ps.gz`.

[23] M. H. Escardó. PCF extended with real numbers. *Theoretical Computer Science*, 162(1):79–115, August 1996.

[24] L. Euler. Commentatio in fractionem continuam qua illustris La Grange potestates binomiales expressit. *Mémoires Acad. impér. Sci. Petersb.*, 6:3–11, 1813-1814.

[25] L. Euler. An essay on continued fractions. *Mathematical Systems Theory*, 18:295–328, 1985.

[26] G. Frobenius. Über Relationen zwischen den Näherungsbrüchen von Potenzreihen. *J. für Math. (Crette)*, 90:1–17, 1881.

[27] J. L. Gersting. *Mathematical structures for computer science.* Computer Science Press, third edition, 1993.

[28] R. W. Gosper. Continued fraction arithmetic. Technical Report HAKMEM Item 101B, MIT AI MEMO 239, MIT, February 1972. Available from `ftp://ftp.netcom.com/pub/hb/hbaker/hakmem`.

[29] C. A. Gunter. *Semantics of Programming Languages: Structures and Techniques.* MIT Press, Cambridge, MA, 1992.

[30] G. Hardy. *Ramanujan's collected papers.* Chelsea Publishing Company, 1962.

[31] R. Heckmann. Some results in exact real arithmetic. Department of Computing, Imperial College, 1997.

[32] R. Heckmann. The appearance of big integers in exact real arithmetic based on linear fractional transformations. In *Proceedings of Foundations of Software Science and Computation Structures (FoSSaCS '98)*, volume 1378 of *LNCS*, pages 172–188. Springer Verlag, 1998.

[33] R. Heckmann. Big integers and complexity issues in exact real arithmetic. Presented at the Third Comprox Workshop (September 1997 in Birmingham). Accepted for publication in ENTCS (Electronic Notes in Theoretical Computer Science), Volume 13, 1998.

[34] R. Heckmann. Contractivity of linear fractional transformations. In J. M. Chesneaux, F. Jézéquel, J. L. Lamotte, and J. Vignes, editors, *Third Real Numbers and Computers Conference (RNC3)*, pages 45–59, 1998.

[35] J.-C. Hervé, F. Morain, D. Salesin, B. Serpette, J. Vuillemin, and P. Zimmermann. BigNum: a portable and efficient package for arbitrary-precision arithmetic. Technical report.

[36] I. Holyer. *Functional programming with Miranda.* University College London Press, 1991.

[37] IEEE. IEEE Standard 754 for Binary Floating-Point Arithmetic. *SIGPLAN*, 22(2):9–25, 1985.

[38] D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Almqvist and Wiksell, 1981.

[39] P. Kornerup and D. W. Matula. An on-line arithmetic unit for bit-pipelined rational arithmetic. *Journal of Parallel and Distributed Computing*, 5(3):310–330, 1988.

[40] P. Kornerup and D. W. Matula. Exploiting redundancy in bit-pipelined rational arithmetic. In *Proceedings of the* 9^{th} *IEEE Symposium on Computer Arithmetic*, pages 119–126, Santa Monica, 1989. IEEE Computer Society Press.

[41] P. Kornerup and D. W. Matula. An algorithm for redundant binary bit-pipelined rational arithmetic. *IEEE Transactions on Computers*, C-39(8):1106–1115, 1990.

[42] P. Kornerup and D. W. Matula. Finite precision lexicographic continued fraction number systems. In *Proceedings of the* 7^{th} *IEEE Symposium on Computer Arithmetic*, pages 207–214, Urbana, 1995. IEEE Computer Society Press.

[43] E. E. Kummer. Uber die hypergeometrische Reihe $F(\alpha, \beta, x)$. *J. für Math.*, 15:39–83, 1836.

[44] J. H. Lambert. *Beiträge zum Gebrauch der Mathematik und deren Anwendung*, volume 1 of *Zweiter Teil*. Berlin, 1770.

[45] D. R. Lester. Vuillemin's exact real arithmetic. In R. Heldal, C. K. Holst, and P. L. Wadler, editors, *Functional Programming, Glasgow 1991: Proceedings of the 1991 Workshop, Portree, UK*, pages 225–238, Berlin, 1992. Springer Verlag.

[46] C. Mazenc. On the redundancy of real number representation systems. Technical Report Research Report 93-16, LIP, Ecole Normale Supérieure de Lyon, 1993. Available from `ftp://ftp.lip.ens-Lyon.fr/pub/LIP/Rapports/RR/RR93/RR93-16.ps.Z`.

[47] V. Ménissier-Morain. Arbitrary precision real arithmetic: design and algorithms. Submitted to the Journal of Symbolic Computation, September 1996. Available from `ftp://ftp.inria.fr/INRIA/ Projects/ cristal/ Valerie.Menissier/ submission_JSC.ps.gz`.

[48] R. E. Moore. *Interval Analysis.* Prentice Hall, Englewood Cliffs, 1966.

[49] J. Myhill. What is a real number? *American Mathematical Monthly*, 79(7):748–754, 1972.

[50] P. M. Neumann, G. A. Stoy, and E. C. Thompson. *Groups and geometry.* Oxford Science Publications, 1995.

[51] A. M. Nielsen and P. Kornerup. MSB-first digit serial arithmetic. *Journal of Universal Computer Science*, 1(7):527–547, July 1995.

[52] H. Padé. Sur la représentation approchée d'une fonction pour des fractions rationnelles. *Ann. Sci. École Norm. Sup. Suppl.*, 9:1–93, 1892.

[53] O. Perron. *Die Lehre von den Kettenbrüchen*, volume 1. B. G. Teubner Verlagsgesellschaft, Stuttgart, 1954.

[54] S. L. Peyton-Jones. Arbitrary precision arithmetic using continued fractions, 1984. INDRA Note 1530, University College London.

[55] G. D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[56] G. D. Plotkin. Post-graduate Lecture Notes in Advanced Domain Theory (incorporating the "Pisa Notes"). Dept. of Computer Science, Univ. of Edinburgh., 1981.

[57] G. D. Plotkin. A powerdomain for countable non-determinism. In M. Nielsen and E. M. Schmidt, editors, *Automata, Languages and programming*, pages 412–428, Berlin, 1982. EATCS, Springer Verlag. Lecture Notes in Computer Science Vol. 140.

[58] P. J. Potts. The storage capacity of forgetful neural networks. Master's thesis, Department of Computing, Imperial College, September 1995.

[59] P. J. Potts. Computable real arithmetic using linear fractional transformations, June 1996. Early draft PhD Thesis, Imperial College, available from `http://www-tfm.doc.ic.ac.uk/~pjp`.

[60] P. J. Potts. A smooth approximation on the edge of chaos. In A. Edalat, G. McCusker, and S. Jourdan, editors, *Proceeding of the Third Imperial College Workshop*, Advances in Theory and Formal Methods of Computing. Imperial College Press, April 1996.

[61] P. J. Potts and A. Edalat. Exact real arithmetic based on linear fractional transformations, December 1996. Imperial College, available from `http://www-tfm.doc.ic.ac.uk/~pjp`.

[62] P. J. Potts and A. Edalat. Exact real computer arithmetic. Technical Report DOC 97/9, Imperial College, March 1997. `http://www-tfm.doc.ic.ac.uk/~pjp`.

[63] P. J. Potts, A. Edalat, and M. H. Escardó. Semantics of exact computer arithmetic. In *Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 248–257, Warsaw, Poland, 1997. IEEE Computer Society Press.

[64] H. A. Priestley. *Introduction to complex analysis.* Oxford Science Publications, revised edition, 1995.

[65] A. Pringsheim. Über ein Convergenz-Kriterium für die Kettenbrüche mit positiven Gliedern. *Sitzungsber. der math.-phys.*, 29:261–268, 1899.

[66] H. Rogers. *Theory of recursive functions and effective computability.* McGraw Hill, 1967.

[67] D. S. Scott. Outline of a mathematical theory of computation. In *4th Annual Princeton Conference on Information Sciences and Systems*, pages 169–176, 1970.

[68] W. A. Sutherland. *Introduction to Metric and Topological Spaces.* Oxford University Press, 1993.

[69] A. M. Turing. On computable numbers with an application to the entscheidungs problem. *Proceedings of the London Mathematical Society*, (42):230–265, 1936.

[70] J. Vuillemin. Exact real computer arithmetic with continued fractions. In *Proceedings ACM conference on Lisp and Functional Programming.* ACM, 1988. Extended version as INRIA research report 760, 1987.

[71] J. Vuillemin. Exact real computer arithmetic with continued fractions. *IEEE Transactions on computers*, 39(8):1087–1105, August 1990.

[72] H. S. Wall. *Analytic Theory of Continued Fractions.* Chelsea Publishing Company, 1948.

[73] H. S. Wall. *Analytic theory of continued fractions.* Chelsea Publishing Company, 1973.

[74] O. Watanuki and M. D. Ercegovac. Error analysis of certain floating-point on-line algorithms. *IEEE Transactions on Computers*, C-32(4):352–358, April 1983.

[75] K. Weihrauch. *Computability.* EATCS monographs on theoretical computer science. Springer Verlag, 1987.

[76] P. Weis, M. Aponte, A. Laville, M. Mauny, and A. Suárez. The CAML reference manual. Technical Report 121, INRIA, Domaine de Voluceau, 78153 Rocquencourt, FRANCE, September 1990.

[77] E. Wiedmer. Computing with infinite objects . *Theoretical Computer Science*, 10:133–155, 1980.

[78] S. Wolfram. *Mathematica: a system for doing mathematics by computer.* Almqvist and Wiksell, 2 edition, 1991.

Index