Integration in Real PCF

Abbas Edalat   Martín Hötzel Escardó
Department of Computing, Imperial College, London SW7 2BZ
{ae,mhe}@doc.ic.ac.uk

1 Introduction

Traditionally, in computing science one represents real numbers by floating-point approximations. If we assume that these approximations are "exact" then we can prove correctness of numerical programs by analytical methods. Such an idealization is the idea behind the BSS model [3]. However, such "correct" programs do not produce correct results in practice, due to the presence of round-off errors. Moreover, they are inappropriate for problems whose solution is sensitive to small variations on the input.

As a consequence, "exact real number computation" has been advocated as an alternative solution (see e.g. [4, 5, 20] on the practical side and e.g. [2, 15, 16, 19, 21, 22, 23] on the foundational side). However, work on exact real number computation has focused on *representations* of real numbers and has neglected the issue of *data types* for real numbers. In particular, programming languages for exact real number computation with an explicit distinction between operational semantics, which is representation-dependent, and denotational semantics, which is representation-independent, have hardly been investigated. Two exceptions are [6] and [11]. Such programming languages do allow for correctness proofs based on analytical methods.

Real PCF [11] is an extension of the programming language PCF [18] with a data type for real numbers, with operational and denotational semantics. Of course, the operational semantics cannot evaluate a program denoting a real number in finitely many steps. However, it can compute an arbitrarily small rational interval containing the real number in a sufficiently large number of steps.

Based on previous work on domain theory and integration [8, 7], we show how to handle integration in Real PCF. In domain-theoretic integration, one obtains increasingly better approximations to the value of the integral of a real-valued function. In order to extend Real PCF with integration, we generalize Riemann integration of real-valued maps of a real variable to interval-valued maps of an interval variable. This also extends the results in the interval analysis approach to integration [17].

We propose two approaches to integration in Real PCF. One consists in adding integration as primitive. The other consists in adding a primitive for maximization of functions and then recursively defining integration from maximization.

The fact that we are able to handle integration in Real PCF shows the strength of such a denotational approach to exact real number computation and also makes explicit the effective content of domain-theoretic integration.

Most proofs have been omitted due to lack of space. Full proofs can be found in [9, 11].

2 Real PCF

In this section we summarize the results of [11] needed in this paper. We assume familiarity with PCF [18, 14]. We are deliberately informal concerning syntax. For simplicity and without essential loss of generality, we restrict ourselves to the unit interval $[0, 1]$.

2.1 Interval expansions

It is well-known that decimal expansions of real numbers are not appropriate for real number computation, if we read infinite expansions from left to right. For example, multiplication by 3 is not computable wrt decimal representation. In fact, any base has essentially the same problem [23].

Let us consider binary expansions of numbers in the unit interval. In this case a solution for the above problem is to allow the digit $\frac{1}{2}$ in addition to the digits 0 and 1. According to Martin-Löf [16], this kind of solution goes back to Brouwer.

For $a_n \in \{0, \frac{1}{2}, 1\}$, the sequence $a_1 a_2 \cdots a_n \cdots$ represents the number $\sum_{n=1}^{\infty} a_n 2^{-n}$. Therefore the operations

$$\begin{aligned} a_1 a_2 \cdots a_n \cdots &\mapsto 0\, a_1 a_2 \cdots a_n \cdots \\ a_1 a_2 \cdots a_n \cdots &\mapsto \tfrac{1}{2}\, a_1 a_2 \cdots a_n \cdots \\ a_1 a_2 \cdots a_n \cdots &\mapsto 1\, a_1 a_2 \cdots a_n \cdots \end{aligned}$$

correspond to the following maps of the unit interval into itself:

$$\begin{aligned} s_0(x) &= (x+0)/2 \\ s_{\frac{1}{2}}(x) &= (x+\tfrac{1}{2})/2 \\ s_1(x) &= (x+1)/2. \end{aligned}$$

Thus, a binary expansion represents an intersection of a shrinking chain of intervals:

$a_1 a_2 \cdots a_n \cdots$ represents $\bigcap_{n=1}^{\infty} s_{a_1} \circ \cdots \circ s_{a_n}([0, 1])$.

Example 2.1.1 Routine algebra shows that the average operation $x \oplus y = (x + y)/2$ satisfies the equations

$$\begin{aligned} s_0(x) \oplus s_0(y) &= s_0(x \oplus y) \\ s_0(x) \oplus s_1(y) &= s_{\frac{1}{2}}(x \oplus y) \\ s_1(x) \oplus s_0(y) &= s_{\frac{1}{2}}(x \oplus y) \\ s_1(x) \oplus s_1(y) &= s_1(x \oplus y), \end{aligned}$$

which can be considered as a *recursive definition* of the average map [10]. □

There is no reason to commit ourselves to the particular operations s_0, $s_{\frac{1}{2}}$ and s_1. These operations are uniquely determined by their images $[0, \frac{1}{2}]$, $[\frac{1}{4}, \frac{3}{4}]$, and $[\frac{1}{2}, 1]$ respectively, in the following sense. Given any interval $[a, b] \subseteq [0, 1]$, there is a unique increasing linear map $\mathrm{cons}_{[a,b]} : [0, 1] \to [0, 1]$ with range $[a, b]$, namely

$$\mathrm{cons}_{[a,b]}(x) = (b - a)x + a.$$

That is, $\mathrm{cons}_{[a,b]}$ *rescales* and *translates* the unit interval so that it becomes $[a, b]$. Therefore the maps s_0, $s_{\frac{1}{2}}$ and s_1 are equal to the maps $\mathrm{cons}_{[0, \frac{1}{2}]}$, $\mathrm{cons}_{[\frac{1}{4}, \frac{3}{4}]}$, and $\mathrm{cons}_{[\frac{1}{2}, 1]}$ respectively.

Definition 2.1 *A sequence of intervals*

$$[a_1, b_1], [a_2, b_2], \ldots, [a_n, b_n], \ldots$$

is said to be an ***interval expansion*** *of the interval*

$$\bigcap_{n=1}^{\infty} \mathrm{cons}_{[a_1,b_1]} \circ \mathrm{cons}_{[a_2,b_2]} \circ \cdots \circ \mathrm{cons}_{[a_n,b_n]}([0, 1]).$$

For example, interval expansions formed from the intervals $[0, 1/10], [1/10, 2/10], \ldots, [9/10, 1]$ are essentially *decimal expansions*.

Interval expansions denote iterated selections of subintervals. For example, the interval expansion

$$[0, \tfrac{1}{2}], [\tfrac{1}{4}, \tfrac{3}{4}], [\tfrac{1}{2}, 1], [0, \tfrac{1}{2}], \ldots,$$

which corresponds to the binary expansion $0\frac{1}{2}10\cdots$, can be interpreted as the following sequence of instructions: select the two middle quarter parts of the interval $[0, \frac{1}{2}]$, select the second half of the resulting interval, select the first half of the resulting interval, and so on. Thus, an interval expansion denotes an intersection of *shrinking* chain of intervals. Conversely, any shrinking chain of intervals gives rise to an interval expansion, as shown at the end of the next subsection.

2.2 The unit interval domain

We think of intervals as *approximations of real numbers*, the singleton intervals being *"exact" approximations*. We consider these approximations as ***generalized real numbers.*** Therefore, we sometimes notationally identify singleton intervals and real numbers.

We let $\mathcal{I}$ be the set of compact subintervals of $[0, 1]$ ordered by reverse inclusion, denoted by $\sqsubseteq$. The letters $\mathbf{x}, \mathbf{y}, \mathbf{z}$, $\mathbf{a}, \mathbf{b}, \mathbf{c}$ in **bold font** range over $\mathcal{I}$, and we put $\mathbf{x} = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$.

$(\mathcal{I}, \sqsubseteq)$ is a continuous Scott domain (bounded complete ω-continuous dcpo) [1] with bottom element $\perp = [0, 1]$. Its way-below relation is given by $\mathbf{x} \ll \mathbf{y}$ iff the interior of $\mathbf{x}$ contains $\mathbf{y}$. The set of maximal elements (singleton intervals) with the subspace topology of the Scott topology of $\mathcal{I}$ is homeomorphic to the Euclidean unit interval. More generally, we can consider the domain $\mathcal{I}[a, b]$ of all compact subintervals of $[a, b]$.

Definition 2.2 *Given intervals* $\mathbf{x}, \mathbf{y} \in \mathcal{I}$, *define* $\mathbf{x} \mathbin{\square} \mathbf{y} = \mathrm{cons}_{\mathbf{x}}(\mathbf{y})$.

Recall that a monoid is a set together with an associative binary operation and a neutral element.

Theorem 2.1 *1. The map* $\mathrm{cons}_{\mathbf{a}} : \mathcal{I} \to \mathcal{I}$ *is continuous for every* $\mathbf{a} \in \mathcal{I}$.

2. $(\mathcal{I}, \square, \perp)$ *is a monoid.*

3. The ***information order*** *of* $(\mathcal{I}, \sqsubseteq)$ *coincides with the* ***prefix preorder*** *of* $(\mathcal{I}, \square, \perp)$, *in the sense that* $\mathbf{x} \sqsubseteq \mathbf{z}$ *iff* $\mathbf{x} \mathbin{\square} \mathbf{y} = \mathbf{z}$ *for some* $\mathbf{y}$. *Moreover, such a y is unique iff* $\mathbf{x}$ *is non-maximal.*

Items 1 and 2 are the basis for the operational semantics of Real PCF and item 3 is the fundamental link between the denotational and the operational semantics.

If $\mathbf{x} \sqsubseteq \mathbf{z}$ and $\mathbf{x}$ is non-maximal then we denote the unique $\mathbf{y}$ such that $\mathbf{x} \mathbin{\square} \mathbf{y} = \mathbf{z}$ by $\mathbf{z} \mathbin{\boxslash} \mathbf{x}$. Now it is easy to see that a shrinking a chain of intervals can be represented by an interval expansion. In fact, if $\mathbf{a}_1 \sqsubseteq \mathbf{a}_2 \sqsubseteq \ldots \sqsubseteq \mathbf{a}_n \sqsubseteq \ldots$ is a chain of non-singleton intervals with lub $\mathbf{x}$ then the sequence $\mathbf{a}_1, (\mathbf{a}_2 \mathbin{\boxslash} \mathbf{a}_1), \ldots, (\mathbf{a}_{n+1} \mathbin{\boxslash} \mathbf{a}_n), \ldots$ is an interval expansion of $\mathbf{x}$.

2.3 Real PCF

The linear maps $\mathrm{cons}_{\mathbf{a}}$ for $\mathbf{a} \neq \perp$ *with distinct rational end-points* will play a rôle analogous to the rôle played by the successor map on natural numbers.

The predecessor map, undefined or arbitrarily defined at zero, is a left inverse of the successor map. Similarly, we now look for a continuous left inverse of $\mathrm{cons}_{\mathbf{a}}$; that is, a map $\mathrm{tail}_{\mathbf{a}}$ such that $\mathrm{tail}_{\mathbf{a}}(\mathrm{cons}_{\mathbf{a}}(\mathbf{x})) = \mathbf{x}$. Since this equation is equivalent to $\mathrm{tail}_{\mathbf{a}}(\mathbf{a} \mathbin{\square} \mathbf{x}) = \mathbf{x}$, we see that $\mathrm{tail}_{\mathbf{a}}$

removes the prefix $\mathbf{a}$ from its argument, if such a prefix exists.

In order to define $\mathrm{tail}_{\mathbf{a}}$, first consider $\mathrm{cons}_{[a,b]}$ as a map $[0,1] \to [0,1]$. The co-restriction of $\mathrm{cons}_{[a,b]}$ to its range $[a,b]$ is invertible. Hence the continuous map $\mathrm{tail}_{[a,b]} : [0,1] \to [0,1]$ defined by

$$\begin{aligned}\mathrm{tail}_{[a,b]}(x) &= \begin{cases} \mathrm{cons}_{[a,b]}^{-1}(a) & \text{if } x \le a \\ \mathrm{cons}_{[a,b]}^{-1}(x) & \text{if } x \in [a,b] \\ \mathrm{cons}_{[a,b]}^{-1}(b) & \text{if } x \ge b \end{cases} \\ &= \max\{0, \min\{(x-a)/(b-a), 1\}\}.\end{aligned}$$

is a left inverse of $\mathrm{cons}_{[a,b]} : [0,1] \to [0,1]$. We thus let $\mathrm{tail}_{[a,b]} : \mathcal{I} \to \mathcal{I}$ be defined by

$$\mathrm{tail}_{[a,b]}([x,y]) = [\mathrm{tail}_{[a,b]}(x), \mathrm{tail}_{[a,b]}(y)].$$

This map is clearly a left inverse of $\mathrm{cons}_{[a,b]} : \mathcal{I} \to \mathcal{I}$ and is continuous.

We need two more primitives, in addition to the linear maps $\mathrm{cons}_{\mathbf{a}}$ and their left inverses $\mathrm{tail}_{\mathbf{a}}$.

The first is a counterpart of the equality test for zero on natural numbers. For each $r \in (0,1)$ define a continuous map $\mathbf{x} \mapsto (\mathbf{x} <_\perp r) : \mathcal{I} \to \mathcal{T}$, where $\mathcal{T} = \{\mathrm{true}, \mathrm{false}\}_\perp$, by

$$\mathbf{x} <_\perp r = \begin{cases} \mathrm{true} & \text{if } \overline{\mathbf{x}} < r \\ \perp & \text{if } r \in \mathbf{x} \\ \mathrm{false} & \text{if } \underline{\mathbf{x}} > r. \end{cases}$$

We are interested in the case that r is rational.

The second primitive is a stronger version of the conditional, called the ***parallel conditional***:

$$\mathrm{pif}\ p \ \mathrm{then}\ \mathbf{x}\ \mathrm{else}\ \mathbf{y} = \begin{cases} \mathbf{x} & \text{if } p = \mathrm{true} \\ \mathbf{x} \sqcap \mathbf{y} & \text{if } p = \perp \\ \mathbf{y} & \text{if } p = \mathrm{false}. \end{cases}$$

This map is also continuous. The idea is that $\mathbf{x} \sqcap \mathbf{y}$ is the best information compatible with both $\mathbf{x}$ and $\mathbf{y}$. Therefore, if the condition is undefined then this information can be safely produced anyway.

Example 2.3.1 The recursive definition of average of real numbers given in Example 2.1.1 generalizes to a recursive definition of average of intervals:

$$\begin{aligned} \mathrm{cons}_L(\mathbf{x}) \oplus \mathrm{cons}_L(\mathbf{y}) &= \mathrm{cons}_L(\mathbf{x} \oplus \mathbf{y}) \\ \mathrm{cons}_L(\mathbf{x}) \oplus \mathrm{cons}_R(\mathbf{y}) &= \mathrm{cons}_C(\mathbf{x} \oplus \mathbf{y}) \\ \mathrm{cons}_R(\mathbf{x}) \oplus \mathrm{cons}_L(\mathbf{y}) &= \mathrm{cons}_C(\mathbf{x} \oplus \mathbf{y}) \\ \mathrm{cons}_R(\mathbf{x}) \oplus \mathrm{cons}_R(\mathbf{y}) &= \mathrm{cons}_R(\mathbf{x} \oplus \mathbf{y}), \end{aligned}$$

where $L = [0, \frac{1}{2}]$, $C = [\frac{1}{4}, \frac{3}{4}]$, and $R = [\frac{1}{2}, 1]$. By means of the primitives that we have introduced, this recursive definition can be rewritten as

$$\begin{aligned} & x \oplus y = \mathrm{pif}\ x <_\perp \tfrac{1}{2} \\ & \quad \mathrm{then}\ \mathrm{pif}\ y <_\perp \tfrac{1}{2}\ \mathrm{then}\ \mathrm{cons}_L(\mathrm{tail}_L(x) \oplus \mathrm{tail}_L(y)) \\ & \qquad\qquad\qquad\qquad \mathrm{else}\ \mathrm{cons}_C(\mathrm{tail}_L(x) \oplus \mathrm{tail}_R(y)) \\ & \quad \mathrm{else}\ \ \mathrm{pif}\ y <_\perp \tfrac{1}{2}\ \mathrm{then}\ \mathrm{cons}_C(\mathrm{tail}_R(x) \oplus \mathrm{tail}_L(y)) \\ & \qquad\qquad\qquad\qquad \mathrm{else}\ \mathrm{cons}_R(\mathrm{tail}_R(x) \oplus \mathrm{tail}_R(y)). \end{aligned}$$ □

Real PCF consists of PCF extended with a ground type for generalized real numbers and constants for the primitive operations introduced in this subsection.

2.4 Adequacy

The operational semantics of Real PCF is given by the following reduction rules:

1. $\mathrm{cons}_{\mathbf{a}}(\mathrm{cons}_{\mathbf{b}}(\mathbf{x})) \to \mathrm{cons}_{\mathrm{cons}_{\mathbf{a}}(\mathbf{b})}(\mathbf{x})$

2. $\mathrm{tail}_{\mathbf{a}}(\mathrm{cons}_{\mathbf{b}}(\mathbf{x})) \to \mathrm{fix}\ (\mathrm{cons}_L)$ if $\overline{\mathbf{a}} \le \underline{\mathbf{b}}$

3. $\mathrm{tail}_{\mathbf{a}}(\mathrm{cons}_{\mathbf{b}}(\mathbf{x})) \to \mathrm{fix}\ (\mathrm{cons}_R)$ if $\overline{\mathbf{b}} \le \underline{\mathbf{a}}$

4. $\mathrm{tail}_{\mathbf{a}}(\mathrm{cons}_{\mathbf{b}}(\mathbf{x})) \to \mathrm{cons}_{\mathrm{tail}_{\mathbf{a}}(\mathbf{b})}(\mathbf{x})$
   if $\mathbf{a} \sqsubseteq \mathbf{b}$ and $\mathbf{a} \ne \mathbf{b}$

5. $\mathrm{tail}_{\mathbf{a}}(\mathrm{cons}_{\mathbf{b}}(\mathbf{x})) \to \mathrm{cons}_{\mathrm{tail}_{\mathbf{a}}(\mathbf{b})}(\mathrm{tail}_{\mathrm{tail}_{\mathbf{b}}(\mathbf{a})}(\mathbf{x}))$
   if $\mathrm{tail}_{\mathbf{a}}(\mathbf{b})$ and $\mathrm{tail}_{\mathbf{b}}(\mathbf{a})$ are non-maximal

6. $\mathrm{cons}_{\mathbf{a}}(\mathbf{x}) <_\perp r \to \mathrm{true}$ if $\overline{\mathbf{a}} < r$

7. $\mathrm{cons}_{\mathbf{a}}(\mathbf{x}) <_\perp r \to \mathrm{false} \quad \text{if } \underline{\mathbf{a}} > r$

8. $\mathrm{pif\ true\ then}\ \mathbf{x}\ \mathrm{else}\ \mathbf{y} \to \mathbf{x}$

9. $\mathrm{pif\ false\ then}\ \mathbf{x}\ \mathrm{else}\ \mathbf{y} \to \mathbf{y}$

10. $\mathrm{pif}\ p\ \mathrm{then}\ \mathrm{cons}_{\mathbf{a}}(\mathbf{x})\ \mathrm{else}\ \mathrm{cons}_{\mathbf{b}}(\mathbf{y}) \to$

$$\begin{array}{ll} \mathrm{cons}_{\mathbf{a}\sqcap\mathbf{b}}(\mathrm{pif}\ p & \mathrm{then}\ \ \mathrm{cons}_{\mathrm{tail}_{\mathbf{a}\sqcap\mathbf{b}}(\mathbf{a})}(\mathbf{x}) \\ & \mathrm{else}\ \ \mathrm{cons}_{\mathrm{tail}_{\mathbf{a}\sqcap\mathbf{b}}(\mathbf{b})}(\mathbf{y})) \end{array}$$

if $\mathbf{a} \sqcap \mathbf{b} \neq \perp$.

Roughly, these rules

1. reduce computations on generalized real numbers to computations on intervals with rational end-points, namely the *subscripts* of cons and tail,

2. "factor out" as many cons primitives as possible.

The underlying idea is that if we have a program X of the form $\mathrm{cons}_{\mathbf{a}}(X')$ with X' *unevaluated*, we know that the result of X is contained in the interval $\mathbf{a}$, because by definition $\mathrm{cons}_{\mathbf{a}}$ is a map with range $\mathbf{a}$. We can thus say that such an X is a ***partially evaluated program***.

The above reduction rules allow us to partially evaluate any program, producing better and better *partial* results converging to its *actual* result, in the sense of Theorem 2.2 below.

Definition 2.3 *We extend the notion of computable PCF term to Real PCF by adding the following clause to the inductive definition given in [18]:*

*A program X denoting a real number $\mathbf{x}$ is **computable** if for every non-bottom interval $\mathbf{y} \ll \mathbf{x}$, as close to $\mathbf{x}$ as we please, there is an interval $\mathbf{a}$ such that $\mathbf{y} \sqsubseteq \mathbf{a} \sqsubseteq \mathbf{x}$ and X partially evaluates to a program of the form $\mathrm{cons}_{\mathbf{a}}(X')$.*

Theorem 2.2 (Adequacy) *Every Real PCF term is computable.*

It follows that a program has some partial evaluation iff it does not denote bottom.

3 Interval Riemann integrals

A generalization of the Riemann theory of integration based on domain theory was introduced in [7]. Essentially, a domain-theoretic framework for the integration of real-valued functions wrt any finite measure on a compact metric space was constructed using the probabilistic power domain of the upper space of the metric space. In this paper we will only be concerned with integration wrt to the Lebesgue measure (uniform distribution) in $\mathbb{R}^n$. Other measures in $\mathbb{R}^n$ will be dealt with in a future paper.

In order to extend Real PCF with integration, we embark on a novel approach compared to [7] for integration wrt the Lebesgue measure in $\mathbb{R}$. We work with the continuous dcpo $\mathcal{R}$ of compact intervals of the real line ordered by reverse inclusion, and we consider integration of maps of type $\mathcal{R}^n \to \mathcal{R}$ rather than $\mathbb{R}^n \to \mathbb{R}$, and we deduce various properties which are interesting in their own right as well.

Addition in the continuous dcpo $\mathcal{R}$ is defined by $\mathbf{x}+\mathbf{y} = [\underline{\mathbf{x}}+\underline{\mathbf{y}}, \overline{\mathbf{x}}+\overline{\mathbf{y}}]$. The map $(\mathbf{x},\mathbf{y}) \mapsto \mathbf{x}+\mathbf{y} : \mathcal{R}\times\mathcal{R} \to \mathcal{R}$ is continuous. Given a real number α and an interval $\mathbf{x}$, we put $\mathbf{x}\alpha = \alpha\mathbf{x} = [\underline{\mathbf{x}}\alpha, \overline{\mathbf{x}}\alpha]$. The map $\mathbf{x} \mapsto \mathbf{x}\alpha : \mathcal{R} \to \mathcal{R}$ is continuous too. We also put $\mathrm{d}\mathbf{x} = \overline{\mathbf{x}} - \underline{\mathbf{x}}$.

A ***partition*** of an interval $[a,b]$ is a finite set of the form

$$P = \{[a,x_1],[x_1,x_2],\ldots,[x_{n-1},x_n],[x_n,b]\}.$$

We denote by $\mathcal{P}[a,b]$ the set of all partitions of $[a,b]$. A partition Q ***refines*** a partition P if Q is obtained by partitioning some elements of P.

Lemma 3.1 *$\mathcal{P}[a,b]$ is directed by the refinement order. That is, for any two partitions of $[a,b]$ there is a third partition refining both.*

Definition 3.1 *Let $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ be a map and $[a,b]$ be an interval. An **interval Riemann sum** of $\mathbf{f}$ on $[a,b]$ is a sum of the form*

$$\sum_{\mathbf{x}\in P} \mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{x} \qquad \textit{for } P \in \mathcal{P}[a,b].$$

Lemma 3.2 *Let $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ be a* monotone *map* (wrt the information order)*. If a partition Q of an interval $[a,b]$ refines a partition P then*

$$\sum_{\mathbf{x}\in P} \mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{x} \sqsubseteq \sum_{\mathbf{x}\in Q} \mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{x}.$$

Therefore, the set of interval Riemann sums of $\mathbf{f}$ on $[a,b]$ is directed.

Definition 3.2 *The **interval Riemann integral** of a* monotone *map $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ on an interval $[a,b]$ is defined by*

$$\int_a^b \mathbf{f} = \bigsqcup_{P\in\mathcal{P}[a,b]} \sum_{\mathbf{x}\in P} \mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{x}.$$

We sometimes denote $\int_a^b f$ by $\int_a^b f(\mathbf{x})\mathrm{d}\mathbf{x}$.

Proposition 3.3 *For all monotone maps $\mathbf{f}, \mathbf{g} : \mathcal{R} \to \mathcal{R}$ and all real numbers α and β,*

$$\begin{aligned} \int_a^a \mathbf{f} &= 0, \\ \int_a^b \mathbf{f} + \int_b^c \mathbf{f} &= \int_a^c \mathbf{f}, \\ \int_a^b (\alpha\mathbf{f} + \beta\mathbf{g}) &= \alpha \int_a^b \mathbf{f} + \beta \int_a^b \mathbf{g}. \end{aligned}$$

Clearly $\int_a^b \mathbf{f}$ depends only on the values that $\mathbf{f}$ assumes on $\mathcal{I}[a,b]$.

Theorem 3.4 *For every interval $[a,b]$ the integration map*

$$\mathbf{f} \mapsto \int_a^b \mathbf{f} : [\mathcal{I}[a,b] \to \mathcal{R}] \to \mathcal{R}$$

is continuous.

Lemma 3.5 *If $\mathbf{f}$ is continuous then we can replace arbitrary partitions in Definition 3.2 by partitions consisting of basis elements, for any basis of the domain $\mathcal{I}[a,b]$.*

Remark 3.6 *Moore [17] handles integration by considering sums which are essentially interval Riemann sums for partitions consisting of n intervals of the same length, but he restricts his definition to rational functions. The integrand is assumed to be monotone wrt inclusion and continuous wrt the Hausdorff metric on intervals. Since the Hausdorff metric induces the Lawson topology on $\mathcal{R}$, the integrand is Scott continuous [12, 13]. Therefore Lemma 3.5 above and Theorem 3.10 below show that our definition generalizes that of Moore to all Scott continuous functions.*

Given any continuous map $f : \mathbb{R} \to \mathbb{R}$, the map $\mathcal{I}f : \mathcal{R}^n \to \mathcal{R}$ defined by

$$\mathcal{I}f(\mathbf{x}) = f(\mathbf{x})$$

is also continuous. Since continuous maps preserve connectedness and compactness,

$$\mathcal{I}f(\mathbf{x}) = [\inf f(\mathbf{x}), \sup f(\mathbf{x})].$$

Hence the end-points of an interval Riemann sum $\sum_{\mathbf{x}\in P} \mathcal{I}f(\mathbf{x})\mathrm{d}\mathbf{x}$ are the lower and upper Darboux sums $\sum_{\mathbf{x}\in P} \inf f(\mathbf{x})\mathrm{d}\mathbf{x}$ and $\sum_{\mathbf{x}\in P} \sup f(\mathbf{x})\mathrm{d}\mathbf{x}$ respectively. Therefore

$$\int_a^b \mathcal{I}f = \left[\underline{\int_a^b} f, \overline{\int_a^b} f\right] = \left\{\int_a^b f\right\}.$$

The map $\mathcal{I}f$ is an ***extension*** of the map f, in the sense that $\mathcal{I}f(\{x\}) = \{f(x)\}$. Any continuous map $f : \mathbb{R} \to \mathbb{R}$ has infinitely many distinct continuous extensions to $\mathcal{R} \to \mathcal{R}$. The extension $\mathcal{I}f$ is characterized as the greatest one. Theorem 3.10 below shows that $\mathcal{I}f$ can be replaced by *any* continuous extension $\mathbf{f}$ in the above equation.

Lemma 3.7 *For every continuous function $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ there is a greatest continuous function $\hat{\mathbf{f}} : \mathcal{R} \to \mathcal{R}$ such that $\mathbf{f}_{|\operatorname{Max}\mathcal{R}} = \hat{\mathbf{f}}_{|\operatorname{Max}\mathcal{R}}$, given by*

$$\begin{aligned}\hat{\mathbf{f}}(\mathbf{x}) &= \bigsqcap \mathbf{f}(\uparrow\mathbf{x} \cap \operatorname{Max}\mathcal{R}) \\ &= \left[\inf_{x\in\mathbf{x}} \underline{\mathbf{f}(\{x\})}, \sup_{x\in\mathbf{x}} \overline{\mathbf{f}(\{x\})}\right]\end{aligned}$$

Lemma 3.8 *For any continuous $\mathbf{f} : \mathcal{R} \to \mathcal{R}$,*

$$\int_a^b \mathbf{f} \quad = \quad \int_a^b \hat{\mathbf{f}}$$

Theorem 3.9 *The integral of a continuous function $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ depends only on the value that $\mathbf{f}$ assumes at maximal elements, in the sense that for any continuous function $\mathbf{g} : \mathcal{R} \to \mathcal{R}$,*

$$\mathbf{f}_{|\operatorname{Max}(\mathcal{R})} = \mathbf{g}_{|\operatorname{Max}(\mathcal{R})} \quad \textit{implies} \quad \int_a^b \mathbf{f} = \int_a^b \mathbf{g}.$$

Theorem 3.10 *If $f : \mathbb{R} \to \mathbb{R}$ is continuous and $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ is a continuous extension of f then*

$$\int_a^b \mathbf{f} \quad = \quad \left\{ \int_a^b f \right\}.$$

The significance of Theorems 3.9 and 3.10 is that sometimes it is easy to obtain a Real PCF program for an extension of a function f but it is difficult or undesirable to obtain a program for its greatest continuous extension. For instance, the distributive law does not hold for the greatest continuous extensions of addition and multiplication, so that two different definitions of the same function can give rise to two different extensions and two different programs [17].

4 Integration in Real PCF

Again, for simplicity and without essential loss of generality, we restrict ourselves to the unit interval. Clearly, the map $\int_0^1 : [\mathcal{I} \to \mathcal{R}] \to \mathcal{R}$ restricts to $[\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$. We denote the restriction by $\int$.

In this section we add $\int : [\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$ as a primitive operation to Real PCF.

Lemma 4.1 *For any continuous map $\mathbf{f} : \mathcal{I} \to \mathcal{I}$,*

$$\int \operatorname{cons}_{\mathbf{a}} \circ \mathbf{f} = \operatorname{cons}_{\mathbf{a}} \left(\int \mathbf{f} \right),$$

$$\int \mathbf{f} = \int \mathbf{f} \circ \operatorname{cons}_L \oplus \int \mathbf{f} \circ \operatorname{cons}_R.$$

Notation 4.1 *We let $\int Y \mathrm{d}\mathbf{x}$ stand for $\int \lambda\mathbf{x} Y$ and $\int F(\mathbf{x})\mathrm{d}\mathbf{x}$ stand for $\int F$ if F is a primitive operation of type $\mathcal{I} \to \mathcal{I}$.*

This notation is ambiguous. For instance, $\int \operatorname{cons}_{\mathbf{a}}(\mathbf{x})\mathrm{d}\mathbf{x}$ can stand for both $\int \operatorname{cons}_{\mathbf{a}}$ and $\int \lambda\mathbf{x} \operatorname{cons}_{\mathbf{a}}(\mathbf{x})$. However, his ambiguity does not introduce inconsistencies, because both terms have the same meaning.

Lemma 4.1 gives rise to the following reduction rules:

1. $\int \operatorname{cons}_{\mathbf{a}}(Y)\mathrm{d}\mathbf{x} \to \operatorname{cons}_{\mathbf{a}} \left(\int Y \mathrm{d}\mathbf{x}\right)$
2. $\int Y[\mathbf{x}]\mathrm{d}\mathbf{x} \to \int Y[\operatorname{cons}_L(\mathbf{x})]\mathrm{d}\mathbf{x} \oplus \int Y[\operatorname{cons}_R(\mathbf{x})]\mathrm{d}\mathbf{x}$
3. $\int Y \mathrm{d}\mathbf{x} \to \int Y' \mathrm{d}\mathbf{x} \quad \text{if } Y \to Y'$.

Here $Y[\mathbf{x}]$ is a term of type $\mathcal{I}$ with some free occurrences of $\mathbf{x}$, and $Y[X]$ denotes the term Y with all free occurrences of $\mathbf{x}$ replaced by the term X.

We call these rules the ***output***, ***input***, and ***production*** rules respectively. Intuitively, the output rule produces partial output, the input rule supplies partial input, and the production rule partially evaluates the integrand (with no input or with the partial input supplied by the input rule in previous reduction steps).

In order to establish adequacy we need some lemmas.

Lemma 4.2 *For every natural number n define a map $\int^{(n)} : [\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$ by*

$$\int^{(n)} \mathbf{f} = \sum_{k=1}^{2^n} \mathbf{f}\left(\left[\frac{k-1}{2^n}, \frac{k}{2^n}\right]\right) \frac{1}{2^n}.$$

Then $\int^{(n)}$ is continuous and

$$\int \mathbf{f} = \bigsqcup_{n=0}^{\infty} \int^{(n)} \mathbf{f},$$

where the lub is directed.

Lemma 4.3 *For every natural number n,*

$$\begin{aligned} \int^{(0)} \mathbf{f} &= \mathbf{f}(\perp), \\ \int^{(n+1)} \mathbf{f} &= \int^{(n)} \mathbf{f} \circ \mathrm{cons}_L \oplus \int^{(n)} \mathbf{f} \circ \mathrm{cons}_R. \end{aligned}$$

Corollary 4.4 *For every n there is a program in Real PCF (without the integration primitive) defining $\int^{(n)}$.*

Lemma 4.5 *If $F : \mathcal{I} \to \mathcal{I}$ is a closed computable term then every partial result produced by the program $\int^{(n)} F$ is also produced by the program $\int F$.*

Proof By induction on n. For the base case use the production rule, the output rule, and the fact that F is computable. For the inductive step use the input rule 2^n times, the fact that $\oplus$ and $\int^{(n)}$ are computable, and the fact that $Y[\mathrm{cons}_{\mathbf{a}}(\mathbf{x})]$ is computable whenever $Y[\mathbf{x}]$ is computable. □

Lemma 4.6 *$\int$ is a computable primitive operation.*

Proof Let F be any closed computable program denoting a function $\mathbf{f} : \mathcal{I} \to \mathcal{I}$, and let $\mathbf{y} \ll \int \mathbf{f}$. Since $\int \mathbf{f}$ is the lub of the chain $\{\int^{(n)} \mathbf{f}\}_{n=0}^{\infty}$, there is an n such that $\mathbf{y} \ll \int^{(n)} \mathbf{f}$, by a basic property of the way-below relation in *continuous* dcpos. Now, $\int^{(n)} F$ is computable because $\int^{(n)}$ and F are computable. Hence there is some $\mathbf{a}$ with $\mathbf{y} \sqsubseteq \mathbf{a} \sqsubseteq \int^{(n)} \mathbf{f}$ such that $\int^{(n)} F$ partially evaluates to a program of the form $\mathrm{cons}_{\mathbf{a}}(X)$, by definition of computability. But $\mathbf{y} \sqsubseteq \mathbf{a} \sqsubseteq \int^{(n)} \mathbf{f} \sqsubseteq \int \mathbf{f}$ and $\int F$ partially evaluates to a program of the form $\mathrm{cons}_{\mathbf{a}}(X')$, by virtue of Lemma 4.5. Therefore $\int$ is computable. □

Theorem 4.7 (Adequacy) *Every term in Real PCF extended with integration is computable.*

5 Recursive integration

It is natural to ask if the integration primitive discussed in the previous section is already recursively definable in Real PCF. Lemma 4.1 leads one to consider the map $G : D \to D$ defined by

$$G(F)(\mathbf{f}) = F(\mathbf{f} \circ \mathrm{cons}_L) \oplus F(\mathbf{f} \circ \mathrm{cons}_R),$$

where $D = [\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$. Then the integration operator $\int$ is a fixed point of G. However, the least fixed point is the bottom element of D.

Peter Freyd suggested that if we restrict ourselves to the subspace $D' \subseteq D$ of functions $F \in D$ such that $\inf \mathbf{f} \leq F(\mathbf{f}) \leq \sup \mathbf{f}$, for $\inf \mathbf{f}$ and $\sup \mathbf{f}$ suitably defined, then G restricts to $G' : D' \to D'$ and $\int$ is the least fixed point of G'. We use this idea in a modified form, obtaining $\int$ directly as the least fixed point of a function $H : D \to D$.

Recall that we defined a continuous extension $\mathcal{I}f : \mathcal{R} \to \mathcal{R}$ for every continuous map $f : \mathbb{R} \to \mathbb{R}$ in Section 3. This construction clearly generalizes to continuous functions $\mathbb{R}^n \to \mathbb{R}$. When there is no danger of ambiguity we write f instead of $\mathcal{I}f$.

Define a map $(x, y, z) \mapsto x(\!|y|\!)z : \mathbb{R}^3 \to \mathbb{R}$ by $x(\!|y|\!)z = \max(x, \min(y, z)) = \min(\max(x, y), z)$. Then, given $a \leq b$, the map $r(x) = a(\!|x|\!)b$ is idempotent, $a \leq r(x) \leq b$, and $r(x) = x$ iff $a \leq x \leq b$.

This observation motivates the following definition of $H : D \to D$:

$$H(F)(\mathbf{f}) = \inf \mathbf{f} (\!| F(\mathbf{f} \circ \mathrm{cons}_L) \oplus F(\mathbf{f} \circ \mathrm{cons}_R) |\!) \sup \mathbf{f},$$

where $\sup : [\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$ is defined by

$$\sup \mathbf{f} \quad = \quad \left[\sup_{\mathbf{x} \in \mathrm{Max}\,\mathcal{I}} \underline{\mathbf{f}(\mathbf{x})}, \sup_{\mathbf{x} \in \mathrm{Max}\,\mathcal{I}} \overline{\mathbf{f}(\mathbf{x})} \right]$$

and $\inf : [\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$ is defined similarly.

Lemma 5.1 $\inf, \sup : [\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$ *are continuous operators, and for every continuous function* $f : \mathcal{I} \to \mathcal{I}$,

$$H^n(\perp)(\mathbf{f}) = \int^{(n)} \hat{\mathbf{f}},$$

where $\hat{\mathbf{f}}$ *is defined as in Lemma 3.7.*

Proposition 5.2 $\int$ *is the least fixed point of* H.

Lemma 5.3 $\min$ *and* $\max$ *(in their curried form* $\mathcal{I} \to \mathcal{I} \to \mathcal{I}$*) are definable in Real PCF.*

Therefore, if inf and sup are definable in Real PCF then $\int$ is also definable. But it is improbable that inf or sup are definable, because this would imply that the existential quantifier defined in [18], which is not definable in PCF, is definable in Real PCF.

However, sup can be introduced as a primitive operation.

Lemma 5.4 *For any continuous map* $\mathbf{f} : \mathcal{I} \to \mathcal{I}$,

$$\sup \mathrm{cons}_{\mathbf{a}} \circ \mathbf{f} = \mathrm{cons}_{\mathbf{a}} (\sup \mathbf{f}),$$

$$\sup \mathbf{f} = \max (\sup \mathbf{f} \circ \mathrm{cons}_L, \sup \mathbf{f} \circ \mathrm{cons}_R).$$

Notation 5.1 *We let* $\sup_{\mathbf{x}} Y$ *stand for* $\sup \lambda \mathbf{x} Y$ *and* $\sup_{\mathbf{x}} F(\mathbf{x})$ *stand for* $\sup F$ *if* F *is a primitive operation of type* $\mathcal{I} \to \mathcal{I}$.

Lemma 5.4 gives rise to the following reduction rules:

1. $\sup_{\mathbf{x}} \mathrm{cons}_{\mathbf{a}}(Y) \to \mathrm{cons}_{\mathbf{a}} \left(\sup_{\mathbf{x}} Y \right)$
2. $\sup_{\mathbf{x}} Y[\mathbf{x}] \to \max \left(\sup_{\mathbf{x}} Y[\mathrm{cons}_L(\mathbf{x})] \right) \left(\sup_{\mathbf{x}} Y[\mathrm{cons}_R(\mathbf{x})] \right)$
3. $\sup_{\mathbf{x}} Y \to \sup_{\mathbf{x}} Y' \quad$ if $Y \to Y'$.

Notice that these are the reduction rules for $\int$ with $\int$ and $\oplus$ replaced by sup and max respectively. We obtain the following similar results.

Lemma 5.5 *For every natural number* n *define a map* $\sup^{(n)} : [\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$ *by*

$$\sup^{(n)} \mathbf{f} = \max_{k=1}^{2^n} \mathbf{f} \left(\left[\frac{k-1}{2^n}, \frac{k}{2^n} \right] \right).$$

Then $\sup^{(n)}$ *is continuous and*

$$\sup \mathbf{f} = \bigsqcup_{n=0}^{\infty} \sup^{(n)} \mathbf{f},$$

where the lub is directed.

Lemma 5.6 *For every natural number* n,

$$\begin{aligned} \sup^{(0)} \mathbf{f} &= \mathbf{f}(\perp), \\ \sup^{(n+1)} \mathbf{f} &= \max \left(\sup^{(n)} \mathbf{f} \circ \mathrm{cons}_L, \sup^{(n)} \mathbf{f} \circ \mathrm{cons}_R \right). \end{aligned}$$

Corollary 5.7 *For every* n *there is a Real PCF program defining* $\sup^{(n)}$.

Lemma 5.8 *If* $F : \mathcal{I} \to \mathcal{I}$ *is a closed computable term then every partial result produced by the program* $\sup^{(n)} F$ *is also produced by the program* $\sup F$.

Lemma 5.9 sup *is a computable primitive operation.*

Theorem 5.10 (Adequacy) *Every term in Real PCF extended with* sup *is computable.*

The operation inf is definable from sup by $\inf \mathbf{f} = 1 - \sup_{\mathbf{x}} 1 - \mathbf{f}(\mathbf{x})$.

Acknowledgements

We would like to thank Peter Freyd for his suggestion about how to obtain a recursive definition of the integral operator. The second author has been supervised by Mike Smyth and has benefited from endless discussions with him. The first author has been supported by two EPSRC projects: "Constructive Approximation Using Domain Theory" and "Foundational Structures in Computer Science". The second author has been supported by the Brazilian agency CNPq.

References

[1] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D.M. Gabbay, and T.S.E Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3, pages 1–168. Clarendon Press, Oxford, 1994.

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

[3] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers. *Bull. Amer. Math. Soc.*, 21:1–46, 1989.

[4] H.J. Boehm and R. Cartwright. Exact real arithmetic: Formulating real numbers as functions. In Turner. D., editor, *Research Topics in Functional Programming*, pages 43–64. Addison-Wesley, 1990.

[5] H.J. Boehm, R. Cartwright, M. Riggle, and M.J. O'Donnel. Exact real arithmetic: A case study in higher order programming. In *ACM Symposium on Lisp and Functional Programming*, 1986.

[6] P. di Gianantonio. *A Functional Approach to Computability on Real Numbers*. PhD thesis, Università Degli Studi di Pisa, Dipartamento di Informatica, 1993.

[7] A. Edalat. Domain theory and integration. *Theoretical Computer Science*, 151:163–193, 1995.

[8] A. Edalat. Dynamical systems, measures and fractals via domain theory. *Information and Computation*, 120(1):32–48, July 1995.

[9] A. Edalat and M.H. Escardó. Real PCF extended with integration. Imperial College, London (available by anonymous ftp at theory@doc.ic.ac.uk:papers/Escardo/papers), November 1995.

[10] M.H. Escardó. Induction and recursion on the real line. In C. Hankin, I. Mackie, and R. Nagarajan, editors, *Theory and Formal Methods 1994: Proceedings of the Second Imperial College Department of Computing Workshop on Theory and Formal Methods*, pages 259–282, Møller Centre, Cambridge, UK, 11–14 September 1994. IC Press. 1995.

[11] M.H. Escardó. PCF extended with real numbers. *Theoretical Computer Science*, to appear in 1996. (available by anonymous ftp at theory@doc.ic.ac.uk:papers/Escardo/papers).

[12] M.H. Escardó and D.M. Claudio. Scott domain theory as a foundation for interval analysis. Technical Report 218, UFRGS/II, Porto Alegre, Brazil, 1993.

[13] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, New York, 1980.

[14] C. A. Gunter. *Semantics of Programming Languages – Structures and Techniques*. The MIT Press, London, 1992.

[15] Ker-I Ko. *Complexitity Theory of Real Functions*. Birkhauser, Boston, 1991.

[16] P. Martin-Lof. *Notes on Constructive Mathematics*. Almqvist & Wiksell, Stockholm, 1970.

[17] R.E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, 1966.

[18] G. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5(1):223–255, 1977.

[19] M.B. Pour-el and I. Richards. Computability and non-computability in classical analysis. *Trans. Am. Math. Soc.*, pages 539–560, 1983.

[20] J. Vuillemin. Exact real arithmetic with continued fractions. In *Proc. ACM Conference on Lisp and Functional Programming*, pages 14–27, 1988.

[21] Klaus Weihrauch. *Computability*. Springer-Verlag, Berlin, 1987.

[22] Klaus Weihrauch. A simple introduction to computable analysis. Technical Report 171 – 7/1995, FernUniversitat, 1995.

[23] E. Wiedmer. Computing with infinite objects. *Theoretical Computer Science*, 10:133–155, 1980.