Integration in Real PCF

Abbas Edalat Martín Hötzel Escardó

Department of Computing
Imperial College
180 Queen's Gate, London SW7 2BZ, UK
{ae,mhe}@doc.ic.ac.uk

Abstract

Real PCF is an extension of the programming language PCF with a data type for real numbers. Although a Real PCF definable real number cannot be computed in finitely many steps, it is possible to compute an arbitrarily small rational interval containing the real number in a sufficiently large number of steps. Based on a domain-theoretic approach to integration, we show how to define integration in Real PCF. 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. In both cases we have an adequacy theorem for the corresponding extension of Real PCF. Moreover, based on previous work on Real PCF definability, we show that Real PCF extended with the maximization operator is universal, which implies that it is also fully abstract.

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 so-called 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, 25] on the practical side and e.g. [2, 20, 21, 24, 26, 27, 28] 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 [16]. Such programming languages do allow for correctness proofs based on analytical methods.

Real PCF [16] is an extension of the programming language PCF [23] 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 [11, 8], 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. This has led to exact computations of integrals in various fields such as statistical physics [7], neural nets [9], stochastic processes [10], and fractal geometry [12].

In order to handle integration in Real PCF, 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 [22].

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. In both cases we have an adequacy theorem for the corresponding extension of Real PCF. Moreover, based on previous work on Real PCF definability [14], we show that Real PCF extended with the maximization operator is universal, which implies that it is also fully abstract.

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 makes explicit the effective content of domain-theoretic integration.

Since numerical solution to differential equations is invariably based on integration of functions, Real PCF with integration also provides a framework for solving differential equations up to any precision.

Organization

In Section 2 we briefly introduce Real PCF. In Section 3 we define interval Riemann integrals. In Section 4 we extend Real PCF with a primitive for integration. In Section 5 we define a function maximization operator. In Section 6 we extend Real PCF with a primitive for maximization. In Section 7 we show how to recursively define integration from maximization. In Section 8 we define multiple integrals and show how to define them from interval Riemann integration. In Section 9 we show that Real PCF extended with maximization is universal.

Several proofs have been omitted due to lack of space. For a full version of this paper containing all proofs see [13].

2. Real PCF

In this section we summarize the results of [16, 14] needed in this paper. We assume familiarity with PCF [23, 19]. 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 w.r.t. decimal representation. In fact, any base has essentially the same problem [28].

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 [21], 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

$$\sum_{n \geq 1} a_n 2^{-n}.$$

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

$$\bigcap_{n \geq 1} s_{a_1} \circ \cdots \circ s_{a_n}([0, 1]).$$

Example 2.1 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 [15]. $\square$

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 2.2 *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

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

For example, interval expansions formed from the intervals

$$[0,\ 1/10], [1/10,\ 2/10], \ldots, [9/10,\ 1]$$

are essentially *decimal expansions* of real numbers contained in the unit interval.

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 closed 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 $\bot = [0, 1]$. Its way-below relation is given by

$\mathbf{x} \ll \mathbf{y}$ iff the interior of $\mathbf{x}$ contains $\mathbf{y}$.

The set $\mathrm{Max}(\mathcal{I})$ 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 $\mathbf{I}[a, b]$ of all closed subintervals of $[a, b]$.

Definition 2.3 *Given intervals $\mathbf{x}, \mathbf{y} \in \mathcal{I}$, define a binary operation $\square$ on $\mathcal{I}$ by*

$$\mathbf{x} \,\square\, \mathbf{y} = \mathrm{cons}_{\mathbf{x}}(\mathbf{y}).$$

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

Theorem 2.4 *1. The map $\mathrm{cons}_{\mathbf{a}} : \mathcal{I} \to \mathcal{I}$ is continuous for every $\mathbf{a} \in \mathcal{I}$.*

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

3. The information order of $(\mathcal{I}, \sqsubseteq)$ coincides with the prefix preorder of $(\mathcal{I}, \square, \bot)$, in the sense that

$$\mathbf{x} \sqsubseteq \mathbf{z} \text{ iff } \mathbf{x} \,\square\, \mathbf{y} = \mathbf{z} \text{ for some } \mathbf{y}.$$

Moreover, such a $\mathbf{y}$ 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 $\mathbf{x} \,\square\, \mathbf{y} = \mathbf{z}$ by $\mathbf{z} \boxslash \mathbf{x}$. Now it is easy to see that a shrinking a chain of intervals can be represented by an interval expansion. In fact, if

$$\mathbf{a}_1 \sqsubseteq \mathbf{a}_2 \sqsubseteq \cdots \sqsubseteq \mathbf{a}_n \sqsubseteq \ldots$$

is a chain of non-singleton intervals with join $\mathbf{x}$ then the sequence

$$\mathbf{a}_1, (\mathbf{a}_2 \boxslash \mathbf{a}_1), \ldots, (\mathbf{a}_{n+1} \boxslash \mathbf{a}_n), \ldots$$

is an interval expansion of $\mathbf{x}$.

2.3. Real PCF

The linear maps $\mathrm{cons}_{\mathbf{a}}$ for $\mathbf{a} \neq \bot$ *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

$$\mathrm{tail}_{\mathbf{a}}(\mathbf{a} \,\square\, \mathbf{x}) = \mathbf{x},$$

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

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

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. The left-inverses of s_0, $s_{\frac{1}{2}}$ and s_1 corresponding to $\mathrm{tail}_{[0,\,\frac{1}{2}]}$, $\mathrm{tail}_{[\frac{1}{4},\,\frac{3}{4}]}$, and $\mathrm{tail}_{[\frac{1}{2},\,1]}$ respectively are

$$\begin{aligned} p_0(x) &= \min(2x,\,1) \\ p_{\frac{1}{2}}(x) &= \max(0,\,\min(2x - \tfrac{1}{2},\,1)) \\ p_1(x) &= \max(0,\,2x-1). \end{aligned}$$

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

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

We are interested in the case that r is rational.

Remark 2.5 The function $\mathbf{x} \mapsto (\mathbf{x} <'_\perp r) : \mathcal{I} \to \mathcal{T}$, defined by

$$\mathbf{x} <'_\perp r \;=\; \begin{cases} \text{true} & \text{if } \overline{\mathbf{x}} < r \\ \text{false} & \text{if } \underline{\mathbf{x}} \geq r \\ \perp & \text{otherwise.} \end{cases}$$

is monotone but not continuous, and hence not computable. In fact, equality of real numbers is not decidable [21] (see Subsection 2.4 below). The map $\mathbf{x} \mapsto (\mathbf{x} <_\perp r)$ can be regarded as the best continuous approximation to the monotone function $\mathbf{x} \mapsto (\mathbf{x} <'_\perp r)$. □

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

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

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 (see Subsection 2.4 below).

Example 2.6 The recursive definition of average of real numbers given in Example 2.1 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

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

For recursive definitions of the complement map $x \mapsto 1-x$, multiplication, and logarithm see [16].

Definition 2.7 *Real PCF consists of PCF extended with a ground type $\mathcal{I}$ for the unit interval and constants for the primitive operations introduced in this subsection.* □

An extension of PCF with a further type for the real line is introduced in [16].

2.4. A note on the parallel conditional

Recall that the *sequential conditional* is defined by

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

Proposition 2.8 *Let R be a domain with $\mathrm{Max}(R)$ homeomorphic to the real line or the unit interval, let D be any domain, let $p : R \to \mathcal{T}$ be a continuous predicate, let $g, h : R \to D$ be continuous functions, and define a function $f : R \to D$ by*

$$f(x) = \text{if } p(x) \text{ then } g(x) \text{ else } h(x).$$

If p is non-trivial, *in the sense that there are maximal elements x and y such that $p(x) = \mathrm{true}$ and $p(y) = \mathrm{false}$, then f is* not total, *in the sense that $f(z) = \perp$ for some maximal element z.*

Proof The non-empty disjoint sets $U = p^{-1}(\mathrm{true}) \cap \mathrm{Max}(R)$ and $V = p^{-1}(\mathrm{false}) \cap \mathrm{Max}(R)$ are open in $\mathrm{Max}(R)$, because p is continuous, and $\{\mathrm{true}\}$ and $\{\mathrm{false}\}$ are open in $\mathcal{T}$. Hence $U \cup V \neq \mathrm{Max}(R)$, because $\mathrm{Max}(R)$ is connected. Therefore there is some maximal element z such that $p(z) = \perp$. □

Thus, the sequential conditional is not appropriate for definition by cases of total functions on R, because it produces non-total functions in non-trivial cases.

In most definitions by cases of the form

$$f(x) = \text{pif } p(x) \text{ then } g(x) \text{ else } h(x)$$

which occur in practice, one has that $f(x) = g(x)$ for all maximal x with $p(x) = \perp$. This is the case, for instance, in the recursive definition of average given in Example 2.6. Another example is given by the following definition of the absolute value function:

$$|x| = \mathrm{pif}\ x <_\perp 0\ \mathrm{then}\ -x\ \mathrm{else}\ x.$$

For the case $x = 0$ one has

$$|0| = \mathrm{pif}\ \perp\ \mathrm{then}\ -0\ \mathrm{else}\ 0 = 0 \sqcap 0 = 0.$$

Hence, the parallel conditional is also useful to overcome the fact that equality of real numbers is not decidable.

2.5. 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 $\overline{\mathbf{a}} \leq \underline{\mathbf{b}}$
3. $\mathrm{tail}_{\mathbf{a}}(\mathrm{cons}_{\mathbf{b}}(\mathbf{x})) \to \mathrm{fix}\ (\mathrm{cons}_R)$ if $\overline{\mathbf{b}} \leq \underline{\mathbf{a}}$
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 $\mathbf{a} \neq \mathbf{b}$
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. $\mathrm{cons}_{\mathbf{a}}(\mathbf{x}) <_\perp r \to \mathrm{false}$ 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. $\mathrm{pif}\ p\ \mathrm{then}\ \mathrm{cons}_{\mathbf{a}}(\mathbf{x})\ \mathrm{else}\ \mathrm{cons}_{\mathbf{b}}(\mathbf{y}) \to$

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

    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*,

then 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 $\mathbf{a}$.

Definition 2.9 *A Real PCF program of the form* $\mathrm{cons}_{\mathbf{a}}(X)$ *is said to be a partially evaluated program with partial result* $\mathbf{a}$.

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 2.11 below.

Definition 2.10 *We extend the notion of* computable PCF term *to Real PCF by adding the following clause to the inductive definition given in [23]:*

> *A Real PCF program X denoting a generalized 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, X produces a partial result $\mathbf{a}$ with*
>
> $$\mathbf{y} \sqsubseteq \mathbf{a} \sqsubseteq \mathbf{x}.$$

Theorem 2.11 (Adequacy) *Every Real PCF term is computable.*

It follows that a program has some partial evaluation iff it does not denote bottom; it is important here that $\mathbf{a}$ cannot be bottom in a primitive operation $\mathrm{cons}_{\mathbf{a}}$.

3. Interval Riemann integrals

A generalization of the Riemann theory of integration based on domain theory was introduced in [8]. Essentially, a domain-theoretic framework for the integration of real-valued functions w.r.t. 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 w.r.t. 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 [8] for integration w.r.t. the Lebesgue measure in $\mathbb{R}$. We work with the continuous dcpo $\mathcal{R} = \mathbf{I}\,\mathbb{R}$ 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 3.2 *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*

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

Lemma 3.3 *Let $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ be a* monotone *map* (w.r.t. 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 3.4 *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 $\int_a^b f(\mathbf{x})\mathrm{d}\mathbf{x}$. $\square$

Proposition 3.5 *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 $\mathbf{I}[a,b]$.

Theorem 3.6 *For every interval $[a,b]$, the integration map*

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

is continuous.

Proof Let $\mathcal{F}$ be a directed subset of $[\mathbf{I}[a,b] \to \mathcal{R}]$. Then

$$\begin{aligned} \int_a^b \bigsqcup \mathcal{F} &= \bigsqcup_P \sum_{\mathbf{x}\in P} \left(\bigsqcup \mathcal{F}\right)(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \bigsqcup_P \sum_{\mathbf{x}\in P} \left(\bigsqcup_{\mathbf{f}\in\mathcal{F}} \mathbf{f}(\mathbf{x})\right)\mathrm{d}\mathbf{x} \\ &= \bigsqcup_P \sum_{\mathbf{x}\in P} \bigsqcup_{\mathbf{f}\in\mathcal{F}} \mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \bigsqcup_P \bigsqcup_{\mathbf{f}\in\mathcal{F}} \sum_{\mathbf{x}\in P} \mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \bigsqcup_{\mathbf{f}\in\mathcal{F}} \bigsqcup_P \sum_{\mathbf{x}\in P} \mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \bigsqcup_{\mathbf{f}\in\mathcal{F}} \int_a^b \mathbf{f}. \qquad \square \end{aligned}$$

Lemma 3.7 *Let $[a,b]$ be an interval, let B be any basis of $\mathbf{I}[a,b]$, and denote by $\mathcal{P}_B[a,b]$ the partitions of $[a,b]$ consisting of basis elements. Then for any* continuous *function $\mathbf{f} : \mathbf{I}[a,b] \to \mathcal{R}$,*

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

Remark 3.8 Moore [22] 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 w.r.t. inclusion and continuous w.r.t. the Hausdorff metric on intervals. Since the Hausdorff metric induces the Lawson topology on $\mathcal{R}$, the integrand is Scott continuous [17, 18]. Therefore Lemma 3.7 above and Theorem 3.12 below show that our definition generalizes that of Moore to all Scott continuous functions. $\square$

Given any continuous function $f : \mathbb{R} \to \mathbb{R}$, the function $\mathbf{I}f : \mathcal{R} \to \mathcal{R}$ defined by

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

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

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

Hence the end-points of an interval Riemann sum are given by lower and upper Darboux sums respectively:

$$\sum_{\mathbf{x}\in P} \mathbf{I}f(\mathbf{x})\mathrm{d}\mathbf{x} = \left[\sum_{\mathbf{x}\in P} \inf f(\mathbf{x})\mathrm{d}\mathbf{x}, \sum_{\mathbf{x}\in P} \sup f(\mathbf{x})\mathrm{d}\mathbf{x}\right].$$

Therefore

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

The map $\mathbf{I}f$ is an *extension* of the map f, in the sense that

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

Any continuous map $f : \mathbb{R} \to \mathbb{R}$ has infinitely many distinct continuous extensions to $\mathcal{R} \to \mathcal{R}$. The extension $\mathbf{I}f$ is characterized as the greatest one. Theorem 3.12 below shows that $\mathbf{I}f$ can be replaced by *any* continuous extension $\mathbf{f}$ in the above equation.

Lemma 3.9 *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*

$$\mathbf{f}_{|\,\mathrm{Max}\,\mathcal{R}} = \hat{\mathbf{f}}_{|\,\mathrm{Max}\,\mathcal{R}},$$

given by

$$\hat{\mathbf{f}}(\mathbf{x}) \quad = \quad \left[\inf_{x\in\mathbf{x}} \underline{\mathbf{f}(\{x\})}, \sup_{x\in\mathbf{x}} \overline{\mathbf{f}(\{x\})}\right].$$

Lemma 3.10 *For any continuous $\mathbf{f} : \mathcal{R} \to \mathcal{R}$,*

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

Theorem 3.11 *The interval Riemann 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 $\mathbf{g} : \mathcal{R} \to \mathcal{R}$,*

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

Proof By Lemma 3.9, $\mathbf{f}_{|\,\mathrm{Max}(\mathcal{R})} = \mathbf{g}_{|\,\mathrm{Max}(\mathcal{R})}$ implies $\hat{\mathbf{f}} = \hat{\mathbf{g}}$. Therefore the result follows from Lemma 3.10. □

Theorem 3.12 *If $f : \mathbb{R} \to \mathbb{R}$ is continuous and $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ is a continuous extension of f then*

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

Proof We know that this is true for $\mathbf{f} = \mathbf{I}f$. If $\mathbf{f}$ is any extension then Lemma 3.9 implies that $\hat{\mathbf{f}} = \mathbf{I}f$. Therefore the result follows from Theorem 3.11. □

The significance of Theorems 3.11 and 3.12 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 [22].

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 $\mathbf{f} : \mathcal{I} \to \mathcal{I}$,*

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

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

Proof (Outline) The first equation is linearity. For the second equation we have

$$\begin{aligned}
\int \mathbf{f} &= \int_0^1 \mathbf{f} \\
&= \int_0^{\frac{1}{2}} \mathbf{f} + \int_{\frac{1}{2}}^1 \mathbf{f} \\
&= \int_0^1 \mathbf{f}\left(\tfrac{\mathbf{x}}{2}\right) \tfrac{1}{2}\mathrm{d}\mathbf{x} + \int_0^1 \mathbf{f}\left(\tfrac{\mathbf{x}+1}{2}\right) \tfrac{1}{2}\mathrm{d}\mathbf{x} \\
&= \int \mathbf{f} \circ \mathrm{cons}_L \oplus \int \mathbf{f} \circ \mathrm{cons}_R. \quad \square
\end{aligned}$$

Notation 4.2 *1. $\int Y \mathrm{d}\mathbf{x}$ stands for $\int \lambda\mathbf{x}Y$.*

2. $\int F(\mathbf{x})\mathrm{d}\mathbf{x}$ stands for $\int F$ if F is a primitive operation of type $\mathcal{I} \to \mathcal{I}$. □

This notation is ambiguous. For instance, $\int \mathrm{cons}_{\mathbf{a}}(\mathbf{x})\mathrm{d}\mathbf{x}$ can stand for both $\int \mathrm{cons}_{\mathbf{a}}$ and $\int \lambda\mathbf{x}\, \mathrm{cons}_{\mathbf{a}}(\mathbf{x})$. 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. $\int \mathrm{cons}_{\mathbf{a}}(Y)\mathrm{d}\mathbf{x} \to \mathrm{cons}_{\mathbf{a}}\left(\int Y \mathrm{d}\mathbf{x}\right)$
2. $\int Y[\mathbf{x}]\mathrm{d}\mathbf{x} \to \int Y[\mathrm{cons}_L(\mathbf{x})]\mathrm{d}\mathbf{x} \oplus \int Y[\mathrm{cons}_R(\mathbf{x})]\mathrm{d}\mathbf{x}$
3. $\int Y \mathrm{d}\mathbf{x} \to \int Y' \mathrm{d}\mathbf{x}$ 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 4.3 *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

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

where the join is directed.

Lemma 4.4 *For every natural number n,*

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

Corollary 4.5 *For every n there is a program in Real PCF (without the integration primitive) defining $\int^{(n)}$.*

Lemma 4.6 *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 if $Y[\mathbf{x}]$ is computable. □

Lemma 4.7 *$\int$ 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 join of the chain $\{\int^{(n)} \mathbf{f}\}_{n \geq 0}$, 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 4.6. Therefore $\int$ is computable. □

Theorem 4.8 (Adequacy) *Every term in Real PCF extended with integration is computable.*

Proof Extend the proof of adequacy of Real PCF given in [16] by including Lemma 4.7 as one of the inductive steps. □

5. A supremum operator

In this section we define a supremum operator. The presentation follows the same pattern as Section 3.

Recall that an extension $\mathbf{I}f : \mathcal{R} \to \mathcal{R}$ of any continuous function $f : \mathbb{R} \to \mathbb{R}$ was defined in Section 3. This construction clearly generalizes to real valued continuous of several real variables. For example, we have that

$$\mathbf{I}\max(\mathbf{x}, \mathbf{y}) = [\max(\underline{\mathbf{x}}, \underline{\mathbf{y}}), \max(\overline{\mathbf{x}}, \overline{\mathbf{y}})].$$

When there is not danger of ambiguity, we write f instead of $\mathbf{I}f$.

Lemma 5.1 *Let $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ be a* monotone *map* (w.r.t. the information order)*. If a partition Q of an interval $[a, b]$ refines a partition P then*

$$\max_{\mathbf{x} \in P} \mathbf{f}(\mathbf{x}) \sqsubseteq \max_{\mathbf{x} \in Q} \mathbf{f}(\mathbf{x}).$$

Definition 5.2 *For a function $f : \mathbb{R} \to \mathbb{R}$ we write*

$$\sup_{[a,b]} f = \sup_{x \in [a,b]} f(x).$$

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

$$\sup_{[a,b]} \mathbf{f} = \bigsqcup_{P \in \mathcal{P}[a,b]} \max_{\mathbf{x} \in P} \mathbf{f}(\mathbf{x}). \quad \square$$

Proposition 5.3 *For all monotone maps $\mathbf{f}, \mathbf{g} : \mathcal{R} \to \mathcal{R}$ and all real numbers α and β,*

$$\begin{aligned} \sup_{[a,a]} \mathbf{f} &= \mathbf{f}(a), \\ \max(\sup_{[a,b]} \mathbf{f}, \sup_{[b,c]} \mathbf{f}) &= \sup_{[a,c]} \mathbf{f}, \\ \sup_{[a,b]} \max(\alpha \mathbf{f}, \beta \mathbf{g}) &= \max(\alpha \sup_{[a,b]} \mathbf{f}, \beta \sup_{[a,b]} \mathbf{g}). \end{aligned}$$

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

Theorem 5.4 *For every interval $[a, b]$, the supremum map*

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

is continuous.

Lemma 5.5 *Let $[a, b]$ be an interval, and let B be any basis of $\mathbf{I}[a, b]$. Then for any* continuous *function $\mathbf{f} : \mathbf{I}[a, b] \to \mathcal{R}$,*

$$\sup_{[a,b]} \mathbf{f} = \bigsqcup_{Q \in \mathcal{P}_B[a,b]} \max_{\mathbf{x} \in Q} \mathbf{f}(\mathbf{x}).$$

Clearly, for $f : \mathbb{R} \to \mathbb{R}$ continuous we have that

$$\max_{\mathbf{x}\in P} \mathbf{I}f(\mathbf{x}) = \left[\max_{\mathbf{x}\in P} \inf f(\mathbf{x}), \max_{\mathbf{x}\in P} \sup f(\mathbf{x})\right].$$

Therefore

$$\sup_{[a,b]} \mathbf{I}f = \left\{\sup_{[a,b]} f\right\}.$$

Lemma 5.6 *For any continuous* $\mathbf{f} : \mathcal{R} \to \mathcal{R}$,

$$\sup_{[a,b]} \mathbf{f} \quad = \quad \sup_{[a,b]} \hat{\mathbf{f}}.$$

Theorem 5.7 *The supremum of a continuous function* $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ *depends only on the value that* $\mathbf{f}$ *assumes at maximal elements.*

Theorem 5.8 *If* $f : \mathbb{R} \to \mathbb{R}$ *is continuous and* $\mathbf{f} : \mathcal{R} \to \mathcal{R}$ *is a continuous extension of* f *then*

$$\sup_{[a,b]} \mathbf{f} \quad = \quad \left\{\sup_{[a,b]} f\right\}.$$

An infimum operator $\inf$ is defined similarly, by replacing $\max$ by $\min$.

6. Real PCF extended with supremum

This section follows the same pattern as Section 4. Again, for simplicity and without essential loss of generality, we restrict ourselves to the unit interval. Clearly, the map $\sup_{[0,1]} : [\mathcal{I} \to \mathcal{R}] \to \mathcal{R}$ restricts to $[\mathcal{I} \to \mathcal{I}] \to \mathcal{I}$. We denote the restriction by $\sup$.

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

Lemma 6.1 $\max$ *and* $\min$ *(in curried form* $\mathcal{I} \to \mathcal{I} \to \mathcal{I}$*) are definable in Real PCF.*

Proof $\max$ can be recursively defined by

$$\begin{array}{l} \max(\mathbf{x},\mathbf{y}) \ = \text{pif } \mathbf{x} <_\perp \tfrac{1}{2} \\ \text{then pif } \mathbf{y} <_\perp \tfrac{1}{2} \ \text{ then } \ \text{cons}_L(\max(\text{tail}_L(\mathbf{x}), \text{tail}_L(\mathbf{y}))) \\ \qquad\qquad\qquad\quad \text{else } \ \mathbf{y} \\ \text{else } \ \text{pif } \mathbf{y} <_\perp \tfrac{1}{2} \ \text{ then } \ \mathbf{x} \\ \qquad\qquad\qquad\quad \text{else } \ \text{cons}_R(\max(\text{tail}_R(\mathbf{x}), \text{tail}_R(\mathbf{y}))), \end{array}$$

and $\min$ can be defined by

$$\min(\mathbf{x},\mathbf{y}) = 1 - \max(1-\mathbf{x}, 1-\mathbf{y}). \quad \Box$$

Lemma 6.2 *For any continuous map* $\mathbf{f} : \mathcal{I} \to \mathcal{I}$,

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

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

Notation 6.3 *1.* $\sup_\mathbf{x} Y$ *stands for* $\sup \lambda\mathbf{x}Y$.

2. $\sup_\mathbf{x} F(\mathbf{x})$ *stands for* $\sup F$ *if* F *is a primitive operation of type* $\mathcal{I} \to \mathcal{I}$. $\Box$

Lemma 6.2 gives rise to the following reduction rules:

1. $\sup_\mathbf{x} \text{cons}_\mathbf{a}(Y) \to \text{cons}_\mathbf{a}\left(\sup_\mathbf{x} Y\right)$
2. $\sup_\mathbf{x} Y[\mathbf{x}] \to \max\left(\sup_\mathbf{x} Y[\text{cons}_L(\mathbf{x})]\right) \left(\sup_\mathbf{x} Y[\text{cons}_R(\mathbf{x})]\right)$
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 6.4 *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*

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

where the join is directed.

Lemma 6.5 *For every natural number* n,

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

Corollary 6.6 *For every* n *there is a Real PCF program defining* $\sup^{(n)}$.

Lemma 6.7 *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 6.8 $\sup$ *is a computable primitive operation.*

Theorem 6.9 (Adequacy) *Every term in Real PCF extended with* $\sup$ *is computable.*

The operation $\inf$ is definable from $\sup$ by

$$\inf \mathbf{f} = 1 - \sup_\mathbf{x}(1 - \mathbf{f}(\mathbf{x})),$$

so there is no need to include it as primitive too.

7. A Recursive definition of integration

It is natural to ask if the integration operator, added in Section 4 as primitive, is already recursively definable in Real PCF.

Let $D = [[\mathcal{I} \to \mathcal{I}] \to \mathcal{I}]$. Then the second equation of Lemma 4.1 leads one to consider the map $G : D \to D$ defined by

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

Thus 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

$$\inf \mathbf{f} \leq F(\mathbf{f}) \leq \sup \mathbf{f},$$

then G restricts to a map $G' : D' \to D'$, 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 $H : D \to D$.

Define a map $p : \mathbb{R}^3 \to \mathbb{R}$ by

$$p(x, y, z) = \max(x, \min(y, z)).$$

Then, given $a \leq b$, the map

$$r(x) = p(a, x, b)$$

is idempotent,

$$a \leq r(x) \leq b,$$

and

$$r(x) = x \text{ iff } a \leq x \leq b.$$

Define a function $H : D \to D$ by

$$H(F)(\mathbf{f}) = p\left(\inf \mathbf{f}, F(\mathbf{f} \circ \mathrm{cons}_L) \oplus F(\mathbf{f} \circ \mathrm{cons}_R), \sup \mathbf{f}\right).$$

Lemma 7.1 *For every continuous function $f : \mathcal{I} \to \mathcal{I}$,*

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

where $\hat{\mathbf{f}}$ is defined as in Lemma 3.9.

Proof By induction on n. For the base case use the fact that $\hat{\mathbf{f}}(\bot) = p(\inf \mathbf{f}, \bot, \sup \mathbf{f})$. □

Proposition 7.2 *$\int$ is the least fixed point of H.*

Proof Immediate consequence of Lemmas 4.3 and 7.1. □

Corollary 7.3 *The integration operator is definable in Real PCF extended with* sup*.*

Proof We clearly have that

$$\mathbf{I}p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \max(\mathbf{x}, \min(\mathbf{y}, \mathbf{z})).$$

Therefore H is definable. □

8. Multiple integrals

A *partition* of a *hyper-cube*

$$\vec{\mathbf{a}} = (\mathbf{a}_1, \ldots, \mathbf{a}_n) \in \mathcal{R}^n$$

is a cartesian product

$$\vec{P} = P_1 \times \cdots \times P_n$$

of partitions of $\mathbf{a}_1, \ldots, \mathbf{a}_n$ respectively. We denote the set of partitions of $\vec{\mathbf{a}}$ by $\mathcal{P}\vec{\mathbf{a}}$. Refinements are defined coordinatewise. The *volume* of an n-dimensional hyper-cube $\vec{\mathbf{x}}$ is

$$\mathrm{d}\vec{\mathbf{x}} = \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_n.$$

Definition 8.1 *Let $\mathbf{f} : \mathcal{R}^n \to \mathcal{R}$ be a map and $\vec{\mathbf{a}}$ be an n-dimensional hyper-cube. A multiple interval Riemann sum of $\mathbf{f}$ on $\vec{\mathbf{a}}$ is a sum of the form*

$$\sum_{\vec{\mathbf{x}} \in \vec{P}} \mathbf{f}(\vec{\mathbf{x}}) \mathrm{d}\vec{\mathbf{x}} \qquad \text{for } \vec{P} \in \mathcal{P}\vec{\mathbf{a}}. \quad \Box$$

Definition 8.2 *The multiple interval Riemann integral of a monotone map $\mathbf{f} : \mathcal{R}^n \to \mathcal{R}$ on a hyper-cube $\vec{\mathbf{a}}$ is defined by*

$$\int_{\vec{\mathbf{a}}} \mathbf{f} = \bigsqcup_{\vec{P} \in \mathcal{P}\vec{\mathbf{a}}} \sum_{\vec{\mathbf{x}} \in \vec{P}} \mathbf{f}(\vec{\mathbf{x}}) \mathrm{d}\vec{\mathbf{x}}. \quad \Box$$

For $n = 1$ this definition reduces to our previous definition:

$$\int_{(\mathbf{a})} \mathbf{f} = \int_{\underline{\mathbf{a}}}^{\overline{\mathbf{a}}} \mathbf{f}.$$

Theorem 8.3 (Fubini's Rule) *For every natural number $n > 1$, every continuous function $\mathbf{f} : \mathcal{R}^n \to \mathcal{R}$, and every n-dimensional hyper-cube $\vec{\mathbf{a}}$,*

$$\int_{\vec{\mathbf{a}}} \mathbf{f} = \int_{(\mathbf{a}_1)} \left(\int_{\vec{\mathbf{a}}'} \mathbf{f}(\mathbf{x}) \mathrm{d}\vec{\mathbf{x}}' \right) \mathrm{d}\mathbf{x}_1,$$

where $\vec{\mathbf{a}}' = (\mathbf{a}_2, \ldots, \mathbf{a}_n)$ and $\vec{\mathbf{x}}' = (\mathbf{x}_2, \ldots, \mathbf{x}_n)$.

Proof For notational simplicity and without essential loss of generality, we prove the claim for $n = 2$:

$$\begin{aligned}
\int_{(\mathbf{a},\mathbf{b})} \mathbf{f} &= \bigsqcup_{P \times Q \in \mathcal{P}(\mathbf{a},\mathbf{b})} \sum_{(\mathbf{x},\mathbf{y}) \in P \times Q} \mathbf{f}(\mathbf{x}, \mathbf{y}) \mathrm{d}(\mathbf{x}, \mathbf{y}) \\
&= \bigsqcup_{P \in \mathcal{P}\mathbf{a}} \bigsqcup_{Q \in \mathcal{P}\mathbf{b}} \sum_{\mathbf{x} \in P} \sum_{\mathbf{y} \in Q} \mathbf{f}(\mathbf{x}, \mathbf{y}) \mathrm{d}\mathbf{x} \mathrm{d}\mathbf{y} \\
&= \bigsqcup_{P \in \mathcal{P}\mathbf{a}} \sum_{\mathbf{x} \in P} \left(\bigsqcup_{Q \in \mathcal{P}\mathbf{b}} \sum_{\mathbf{y} \in Q} \mathbf{f}(\mathbf{x}, \mathbf{y}) \mathrm{d}\mathbf{y} \right) \mathrm{d}\mathbf{x} \\
&= \int_{(\mathbf{a})} \lambda \mathbf{x} \int_{(\mathbf{b})} \lambda \mathbf{y} \mathbf{f}(\mathbf{x}, \mathbf{y}). \quad \Box
\end{aligned}$$

Corollary 8.4 *For every n there is a program in Real PCF extended with either integration or* $\sup$ *which computes the multiple integration operator* $\int : [\mathcal{I}^n \to \mathcal{I}] \to \mathcal{I}$ *of order* n*.*

Proof Since PCF does not have cartesian products, we have to use curried maps. Our primitive or program for integration takes care of the case $n = 1$. The equation of the above theorem can be read as a definition of a program for the case $n+1$ from a program for the case n. By the adequacy theorems, these programs indeed compute multiple integrals of order n. □

This application of the adequacy theorems shows that adequacy is a powerful result. In fact, it allows us to derive correct programs from analytical results, in a representation-independent fashion. This is precisely the idea behind denotational semantics.

9. Universality of Real PCF extended with the supremum operator

Definition 9.1 *A programming language $\mathcal{L}$ is universal if every computable element in the universe of discourse of $\mathcal{L}$ is $\mathcal{L}$-definable.* □

PCF is not universal; however, PCF extended with the parallel conditional and the existential quantification operator $\exists : (\mathcal{N} \to \mathcal{T}) \to \mathcal{T}$ defined by

$$\exists(p) = \begin{cases} \text{true} & \text{if } p(n) = \text{true for for some } n \\ \text{false} & \text{if } p(\bot) = \text{false} \\ \bot & \text{otherwise.} \end{cases}$$

is universal [23]. Real PCF with no extensions is not universal, because $\exists$ is not definable [14]. If we extend Real PCF with $\exists$ and the computation rules given in *loc. cit.*, then the adequacy property remains true. The following universality result is proved in [14]:

Theorem 9.2 *Real PCF extended with $\exists$ is universal.*

Corollary 9.3 *Real PCF extended with $\exists$ is fully abstract.*

The reader is referred to [23] for the definition of full abstraction.

Although Theorem 9.2 implies that $\sup$ is definable in Real PCF extended with $\exists$, we do not know a neat fixed-point definition of $\sup$.

Proposition 9.4 . *The existential quantification operator $\exists$ is definable in Real PCF extended with* $\sup$*.*

Proof For $D \in \{\mathcal{N}, \mathcal{T}\}$, define continuous maps

$$D \underset{s_D}{\overset{r_D}{\leftrightarrows}} \mathcal{I}$$

by

$$\begin{aligned} s_{\mathcal{N}}(n) &= \text{if } n = 0 \text{ then } 0 \text{ else } \text{cons}_R(s_{\mathcal{N}}(n-1)) \\ r_{\mathcal{N}}(x) &= \text{if } x <_\bot 1/4 \text{ then } 0 \text{ else } r_{\mathcal{N}}(\text{tail}_R(x)) + 1, \end{aligned}$$

$$\begin{aligned} s_{\mathcal{T}}(t) &= \text{if } t \text{ then } 1 \text{ else } 0 \\ r_{\mathcal{T}}(x) &= \text{if } x <_\bot 1/2 \text{ then false else true} \end{aligned}$$

Then (s_D, r_D) is a section-retraction pair with D as a retract, in the sense that

$$r_D \circ s_D = \text{id}_D$$

and $s_D \circ r_D$ is an idempotent on $\mathcal{I}$ with image isomorphic to D. This is immediate for $D = \mathcal{T}$. For $D = \mathcal{N}$, we prove by induction on n that $r_{\mathcal{N}} \circ s_{\mathcal{N}}(n) = n$. If $n = \bot$ or $n = 0$ this is immediate. For the inductive step we have that

$$\begin{aligned} r_{\mathcal{N}} \circ s_{\mathcal{N}}(n+1) &= r_{\mathcal{N}}(\text{cons}_R(s_{\mathcal{N}}(n))) \\ &= r_{\mathcal{N}}(\text{tail}_R \circ \text{cons}_R(s_{\mathcal{N}}(n))) + 1 \\ &= r_{\mathcal{N}}(s_{\mathcal{N}}(n)) + 1 \\ &= n + 1 \quad \text{by the induction hypothesis.} \end{aligned}$$

It follows that the diagram below commutes:

$$\begin{array}{ccc} [\mathcal{I} \to \mathcal{I}] & \xrightarrow{\sup} & \mathcal{I} \\ {\scriptstyle r_{\mathcal{N}} \to s_{\mathcal{T}}} \uparrow & & \downarrow {\scriptstyle r_{\mathcal{T}}} \\ [\mathcal{N} \to \mathcal{T}] & \xrightarrow[\exists]{} & \mathcal{T} \end{array}$$

In fact, let $p \in [\mathcal{N} \to \mathcal{T}]$ and define $\mathbf{f} : \mathcal{I} \to \mathcal{I}$ by

$$\mathbf{f} = (r_{\mathcal{N}} \to s_{\mathcal{T}})(p) = s_{\mathcal{T}} \circ p \circ r_{\mathcal{N}}.$$

If there is some n such that $p(n) = \text{true}$, then there is some $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) = 1$, namely $\mathbf{x} = s_{\mathcal{N}}(n)$, and in this case we have that $\sup \mathbf{f} = 1$. If $p(\bot) = \text{false}$, then $\mathbf{f}(\bot) = 0$, and in this case we have that $\sup \mathbf{f} = 0$. Therefore $\exists$ is definable. □

Corollary 9.5 *Real PCF extended with* $\sup$ *is universal and hence fully abstract.*

We do not know whether Real PCF extended with integration is universal. Moreover, we do not know whether integration is definable in Real PCF with no extensions.

For applications of Real PCF to real analysis, it seems more natural to include the supremum operator as a primitive operation than to include the existential quantification operator.

Remark 9.6 Notice that the section-retraction pairs defined in the proof of Proposition 9.4 are expressed in terms of sequential primitives only, and that the maximum operation on $\mathcal{I}$ represents the "parallel or" operation $\vee$ on $\mathcal{T}$ defined in [23], in the sense that the following diagram commutes:

$$\begin{array}{ccc} \mathcal{I} \times \mathcal{I} & \xrightarrow{\max} & \mathcal{I} \\ {\scriptstyle s_{\mathcal{T}} \times s_{\mathcal{T}}} \uparrow & & \downarrow {\scriptstyle r_{\mathcal{T}}} \\ \mathcal{T} \times \mathcal{T} & \xrightarrow[\vee]{} & \mathcal{T} \end{array}$$

Acknowledgements

We would like to thank Peter Freyd for his suggestion about the recursive definition of the integration operator. The second author has been supervised by Mike Smyth and has benefited from endless discussions with him.

This work has been supported by EPSRC projects "Constructive Approximation Using Domain Theory" and "Foundational Structures in Computer Science", and by an ARC project "A Computational Approach to Measure and Integration Theory". The second author has also been supported by the Brazilian agency CNPq.

References

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

[5] H. Boehm, R. Cartwright, M. Riggle, and M. 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 of computation of a random field in statistical physics. 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*, Moler Centre, Cambridge, 11-14 September 1995. IC Press.

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

[9] A. Edalat. Domain theory in learning processes. In S. Brookes, M. Main, A. Melton, and M. Mislove, editors, *Proceedings of MFPS*, volume one of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1995.

[10] A. Edalat. Domain theory in stochastic processes. In *Proceedings of Logic in Computer Science (LICS), Tenth Annual IEEE Symposium, 26-19 June, 1995, San Diego*. IEEE Computer Society Press, 1995.

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

[12] A. Edalat. Power domains and iterated function systems. *Information and Computation*, 124:182–197, 1996.

[13] A. Edalat and M. Escardó. Integration in Real PCF. Available by ftp (theory.doc.ic.ac.uk:papers/Escardo/papers) and WWW (http://theory.doc.ic.ac.uk/∼mhe/), December 1995.

[14] M. Escardó. Real PCF extended with $\exists$ is universal. Available by anonymous ftp at theory. doc. ic. ac. uk: papers/ Escardo/ papers and also by WWW at http:// theory. doc. ic. ac. uk/ ∼mhe/ Imperial College, London.

[15] M. 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.

[16] M. Escardó. PCF extended with real numbers. *Theoretical Computer Science*, 162, July 1996.

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

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

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

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

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

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

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

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

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

[26] K. Weihrauch. *Computability*. Springer-Verlag, Berlin, 1987.

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

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