

An Abstract Data Type for Real Numbers. ^{*}

Pietro Di Gianantonio

Dipartimento di Matematica e Informatica, Università di Udine
via delle Scienze 206 I-33100 Udine Italy
E-mail: digianantonio@dimi.uniud.it

Abstract. We present a PCF-like calculus having real numbers as a basic data type. The calculus is defined by its denotational semantics. We prove the universality of the calculus (i.e. every computable element is definable). We address the general problem of providing an operational semantics to calculi for the real numbers. We present a possible solution based on a new representation for the real numbers.

keywords: real number computability, domain theory, denotational and operational semantics, abstract data types.

1 Introduction

The aim of this work is to relate two different approaches to computability on real numbers: a practical approach based on programming languages, and a more theoretical one based on domain theory. Several implementations of exact computations on real numbers have been proposed so far ([BC90], [MM], [Vui88]). In these works, real numbers are represented by programs generating sequences of discrete elements, e.g. digits. On the other hand, different theoretical works on computability on real numbers are based on domain theory: [Lac59,ML70], [EE96], [DG96]. In all these works domains of approximations for the real numbers are considered. A point in these domains represents either a real number or the approximation of a real number. Approximated reals are normally described by intervals of the real line.

The relation existing between the two approaches is described in several steps. First we present a domain of approximations which is directly derived from a representation for the real number used in some implementations of the exact real number computation ([BC90,MM]). From this domain of approximations we derive a calculus for the real numbers. The calculus we present is an extension of PCF having the real numbers as ground type. We call it \mathcal{L}_r . We define \mathcal{L}_r giving its denotational semantics.

The next natural step consists in giving an operational semantics to the calculus, possibly using the representation for the real numbers we start with. If this would be possible, we will have established a close connection between the

^{*} Work partially supported by an EPSRC grant: “Techniques of Real Number Computation” at Imperial College of Science, Technology and Medicine, London and by EEC/HCM Network “Lambda Calcul Typé”.

domain of approximations for the real numbers and the implementations. We will have a calculus that is for many aspect similar to the calculi used in the implementations and whose terms can be directly interpreted in the approximations domain. Unfortunately we prove that it is impossible to define the operational semantics in this way. We prove this negative result in a general manner, the impossibility holds not only if we consider the particular representation for the real numbers we chose, the domain of approximations obtained from it and the calculus \mathcal{L}_r . The negative result holds for a large class of representations, domains, and calculi.

Finally we define an operational semantics for \mathcal{L}_r . In order to do this however we need to introduce a new representation for the real numbers. This new representation is quite different from the classical ones, in it real numbers can be represented also by sequences of digits undefined on some elements. In order to compute with this new representation is absolutely necessary to use parallel operators. The use of parallel operators is the price we need to pay to have a faithful calculus for the real numbers.

Acknowledgements: I would like to thank Abbas Edalat, Martin Escardo, Peter Potts and Michael Smith for several discussions on the subject.

2 Real Number Computation in PCF

We consider the following representation for real numbers:

Definition 1. *A real number x is represented by a computable sequence of integers $\langle s_0, \dots, s_i, \dots \rangle$ such that:*

- (i) $\forall n. 2s_n - 1 \leq s_{n+1} \leq 2s_n + 1$
- (ii) $x = \bigcap_{n \in \mathbb{N}} \left[\frac{s_n - 1}{2^n}, \frac{s_n + 1}{2^n} \right]$

In this representation a sequence of integers is used to describe a sequence of rational intervals. The intervals in the sequence are contained one into the other. For practical purposes this representation is quite convenient. It allows to reduce exact real number computation to computation on integers. In this way it is possible to exploit the implementation of integer arithmetic already available on computers. In [BCRO86] and [MM] a similar representation has been used to develop quite efficient algorithms for the arithmetic operations.

We refer to [Plo77] for a definition of PCF. In order to represent real numbers in PCF it is sufficient to translate in PCF the representation of Definition 1. In the following, given a type σ , $\mathcal{L}_{PA+\exists}^\sigma$ indicates the set of closed terms in $\mathcal{L}_{PA+\exists}$ having type σ .

Definition 2. *A partial representation function $\text{Eval}_{\mathbb{R}} : \mathcal{L}_{PA+\exists}^{\iota \rightarrow \iota} \rightarrow \mathbb{R}$ is defined by: $\text{Eval}_{\mathbb{R}}(M_{\iota \rightarrow \iota}) = x$ if there exists a sequence of integers s such that:*

- (i) $\forall n \in \mathbb{N}. \text{Eval}(M_{\iota \rightarrow \iota} n) = s_n$;
- (ii) $\forall n. 2s_n - 1 \leq s_{n+1} \leq 2s_n + 1$
- (iii) $x = \bigcap_{n \in \mathbb{N}} \left[\frac{s_n - 1}{2^n}, \frac{s_n + 1}{2^n} \right]$.

A real number x is said \mathcal{L} -computable, if belongs to the image of the $\text{Eval}_{\mathbb{R}}$.

We indicate with \mathbb{R}_l the set of the \mathcal{L} -computable real numbers. The definition of computability can be extended to functions on real numbers.

Definition 3. *The function $\text{Eval}_{\mathbb{R}}^1 : \mathcal{L}_{PA+\exists}^{(\iota \rightarrow \iota) \rightarrow (\iota \rightarrow \iota)} \rightarrow (\mathbb{R}_l \rightarrow \mathbb{R}_l)$ is defined by:*
 $\text{Eval}_{\mathbb{R}}(M) = f$ iff
 $\forall x \in \mathbb{R}_l. \forall N \in \mathcal{L}_{PA+\exists}^{\iota \rightarrow \iota}. \text{Eval}_{\mathbb{R}}(N) = x \Rightarrow \text{Eval}_{\mathbb{R}}(MN) = f(x).$
A function $f : \mathbb{R}_l \rightarrow \mathbb{R}_l$ is said \mathcal{L} -computable if belongs to the image of $\text{Eval}_{\mathbb{R}}^1$.

It is worthwhile to observe that the sequential operators are sufficient to define every computable function. That is every \mathcal{L} -computable function on reals can be defined by a term not containing the parallel test or the existential quantifier. The form of computation presented in this section, is very similar to the one used in implementations of exact real number computation and described in [BC90] and in [MM].

3 A Domain of Approximations for Real Numbers

In the literature there are several approaches to computability on real numbers which use of domain theory. Early works in this ambit are [Lac59], [ML70], and [Sco70]. In all these approaches the real line is embedded in a space of approximations where a notion of computability can be defined in a natural way. Many results concerning the computability theory on real numbers are given in these contexts. Here we are going to present a space of approximations that is similar in many respects to the ones mentioned above but has two important differences. First, we base our construction on the representation of Definition 1. As result our space has less approximation points and is more closely related to the computation describe in [BC90] and [MM]. A second important difference is the following: our space of approximations turns out to be a Scott-domain. The other approaches use spaces of approximations that are continuous but not algebraic cpos. The space of approximations presented here has been extensively studied in [DG96]. Here we resume the main results without giving the proofs.

The domain of approximations defined next is called Reals Domain (RD). We present a construction of RD starting with the integer sequence representation for real numbers. Let $\langle s_i \rangle_{i \in \mathbb{N}}$ be a sequence of integers defining a real number x according to Definition 1 and let $\langle s_i \rangle_{i < n}$ be an initial subsequence. $\langle s_i \rangle_{i < n}$ gives partial information about the value x . Examining $\langle s_i \rangle_{i < n}$ we can deduce that the value x is contained in an interval of real numbers.

Definition 4. *Let S be the subset of sequences of integers defined by:*

$$S = \{ \langle s_i \rangle_{i < n} \mid \forall i < n - 1. 2s_i - 1 \leq s_{i+1} \leq 2s_i + 1 \}.$$

The function ϕ from S to the set of rational intervals is defined by:

$$\phi(\langle s_0, s_1, \dots, s_n \rangle) = \left[\frac{s_n - 1}{2^n}, \frac{s_n + 1}{2^n} \right]$$

The set S contains the “valid” sequences of integers. The function ϕ associates to any finite sequence $\langle s_i \rangle_{i < n}$ the interval $[a, b]$ containing the real numbers that can be represented by sequences having as initial subsequence $\langle s_i \rangle_{i < n}$. The interval $[a, b]$ represents the information contained in the sequence $\langle s_i \rangle_{i < n}$.

Let (DI, \sqsubseteq) denote the partial order formed by the set of rational intervals in the image of the function ϕ . The order relation \sqsubseteq on DI is the superset relation, that is $[a, b] \sqsubseteq [a', b']$ if $[a', b'] \subseteq [a, b]$ (if $[a', b']$ is a more precise approximation of a real number than $[a, b]$). The set DI forms the base of the domain RD .

Definition 5. Let RD be the cpo obtained by the ideal completion of (DI, \sqsubseteq) .

Proposition 1. RD is a consistently complete ω -algebraic cpo (Scott-domain). RD is an effective Scott-domain when we consider the following enumeration of finite elements:

$$e^r(0) = \perp \quad e^r(\langle \langle n_1, n_2 \rangle, n_3 \rangle + 1) = \downarrow [(n_1 - n_2 - 1)/2^{n_3}, (n_1 - n_2 + 1)/2^{n_3}].$$

Where $\langle \rangle$ is an effective coding function for pairs of natural numbers.

The elements of RD can be thought as equivalence classes of (partial) sequences of integers. Each equivalence class is composed by sequences containing identical information about the real value they approximate. The relationship existing between the real line and the infinite elements of RD can be clarified by means of following functions:

Definition 6. A function $q_{\mathcal{P}} : RD \rightarrow \mathcal{P}(\mathbb{R})$ is defined by:

$$q_{\mathcal{P}}(d) = \bigcap_{[a,b] \in d} [a, b]$$

Conversely, three functions $e, e^-, e^+ : \mathbb{R} \rightarrow RD$ are defined by:

$$e(x) = \{[a, b] \in DI \mid x \in (a, b)\}$$

$$e^-(x) = \{[a, b] \in DI \mid x \in (a, b]\} \quad e^+(x) = \{[a, b] \in DI \mid x \in [a, b)\}$$

where (a, b) indicates the open interval from a to b and $(a, b]$ and $[a, b)$ indicate the obvious part open, part closed intervals.

Proposition 2. The following statements hold:

i) for every infinite element $d \in RD$ there exists a real number x such that $q_{\mathcal{P}}(d) = \{x\}$

ii) for every real number x , $\{x\} = q_{\mathcal{P}} \circ e(x) = q_{\mathcal{P}} \circ e^-(x) = q_{\mathcal{P}} \circ e^+(x)$,

iii) for every non-dyadic number $x \in \mathbb{R}/D$, $e(x) = e^-(x) = e^+(x)$,

iv) for every dyadic number $x \in D$, $e(x) \sqsubset e^-(x)$, $e(x) \sqsubset e^+(x)$ and $e^-(x)$ is not consistent with $e^+(x)$,

v) $e(\mathbb{R}) \cup e^-(\mathbb{R}) \cup e^+(\mathbb{R})$ is equal to the set of infinite elements of RD .

We can say that the infinite elements of RD are a close representation of the real line, the set of infinite elements in RD looks like the real line except that each dyadic number is triplicated.

In [DG96] it is shown how to solve the problem of multiple representations by means of a retract construction.

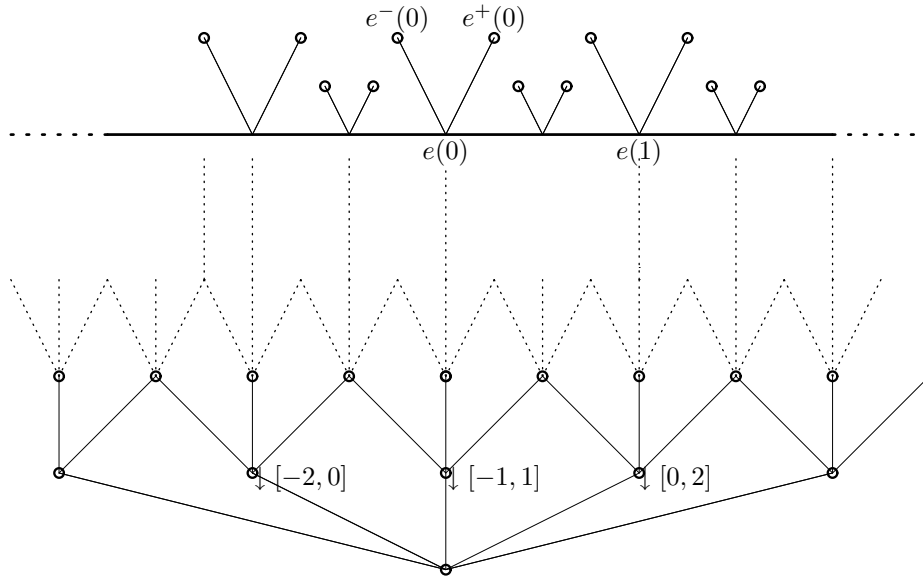


Fig. 1. The diagram representing RD .

4 PCF Extended with Real Numbers

In this section we use the domain RD introduced above, to define an extension of the language PCF having a ground type for the real numbers. We call \mathcal{L}_r this extension. We will prove that any computable function on RD is definable by a suitable expression in \mathcal{L}_r . A programming language similar to \mathcal{L}_r has been introduced in [DG93]. An extension of PCF based on a different domain of approximation for the real numbers has been presented in [Esc96].

Compared with the real computation described in Section 2, the real computation in \mathcal{L}_r has several advantages. Given a closed term $M \in \mathcal{L}^{(\iota \rightarrow \iota) \rightarrow (\iota \rightarrow \iota)}$ the value $\text{Eval}_{\mathbb{R}}(M)^1$ can be undefined for several reasons. For example:

- (i) there can be a term N representing a real number such that the sequence of $((MN)0), \dots, ((MN)n), \dots$ does not define a real number.
- (ii) there can be two terms N_1 and N_2 defining the same real number and such that (MN_1) and (MN_2) define different real numbers.

The language \mathcal{L}_r is free from these inadequacies. Terms of type r in \mathcal{L}_r can always be interpreted as an (approximated) real and more importantly terms of type $r \rightarrow r$ preserve the equivalence between different representations of the same real number. We can say that \mathcal{L}_r defines an abstract data type for real numbers. It defines a collection of primitive functions on reals which generate any other computable function.

The types of \mathcal{L}_r are the PCF types extended with a new ground r . The set T of type expressions is defined by the grammar:

$$\sigma := \iota \mid o \mid r \mid \sigma \rightarrow \tau$$

The terms of \mathcal{L}_r are the terms of $\mathcal{L}_{PA+\exists}$ extended with the new constants:

$$\begin{array}{ll} (-1), (+1), (\times 2), (\div 2), \text{PR} : r \rightarrow r, & \\ (\leq 0) : r \rightarrow o & \text{pif}_r : o \rightarrow r \rightarrow r \rightarrow r, \end{array}$$

We define \mathcal{L}_r giving its denotational semantics. To this end we use the set of Scott-domains, $UD = \{D_\sigma \mid \sigma \in T\}$, where $D_\iota = \mathbb{Z}_\perp$, $D_o = \{\text{tt}, \text{ff}\}_\perp$, $D_r = RD$ and $D_{\sigma \rightarrow \tau} = [D_\sigma \rightarrow D_\tau]$.

The denotation of the new constants is the following:
the constants $(+1), (-1), (\times 2), (\div 2)$ realize the corresponding functions on reals.

$$\begin{aligned} \llbracket (+1) \rrbracket_\rho(d) &= \{[a+1, b+1] \mid [a, b] \in d\} \\ \llbracket (-1) \rrbracket_\rho(d) &= \{[a-1, b-1] \mid [a, b] \in d\} \\ \llbracket (\times 2) \rrbracket_\rho(d) &= \{[a \times 2, b \times 2] \mid [a, b] \in d \wedge [a \times 2, b \times 2] \in RI\} \\ \llbracket (\div 2) \rrbracket_\rho(d) &= \bigcup_{[a, b] \in d} \downarrow [a \div 2, b \div 2] \end{aligned}$$

The constant (≤ 0) tests if a number is smaller or larger than 0.

$$\llbracket (\leq 0) \rrbracket_\rho(d) = \begin{cases} \text{tt} & \text{if it exists } [a, b] \in d, b \leq 0 \\ \text{ff} & \text{if it exists } [a, b] \in d, 0 \leq a \\ \perp & \text{otherwise} \end{cases}$$

The constant PR defines a kind of projection on the interval $[-1, 1]$.

$$\llbracket \text{PR} \rrbracket_\rho(d) = \begin{cases} d \sqcup \downarrow [-1, 1] & \text{if } d \text{ is consistent with } \downarrow [-1, 1] \\ e^+(-1) & \text{if } \exists [a, b] \in d. b \leq -1 \\ e^-(-1) & \text{if } \exists [a, b] \in d. a \geq 1 \end{cases}$$

The constant pif_r defines a parallel test.

$$\llbracket \text{pif}_r \rrbracket_\rho(e)(d)(d') := \begin{cases} d & \text{if } e = \text{tt} \\ d' & \text{if } e = \text{ff} \\ d \sqcap d' & \text{if } e = \perp \end{cases}$$

If the boolean argument is undefined the function $\llbracket \text{pif}_r \rrbracket_\rho$ gives as output the most precise approximation of the second and third argument.

It is not difficult to prove that for every closed expression M^σ and environment ρ , $\llbracket M^\sigma \rrbracket_\rho$ is a computable element of D_σ . Next we prove the universality of \mathcal{L}_r , that is, we prove that every computable functions on RD is definable by a suitable term in \mathcal{L}_r . In order to do this we present a generalisation of the universality theorem for PCF [Plo77, Theorem 5.1]. The generalisation applies to any extension of PCF where ground types are denoted by coherent domains. The proof in [Plo77] works only for flats domains. An equivalent generalisation

has already been given in [Str94]. In that work the proof is based on categorical arguments and uses as a lemma the original result in [Plo77]. Our proof follows the line of the original proof and it is more direct. Some definitions and lemmata are necessary here.

Definition 7. A subset A of a partial order P is coherent if any pair of elements has an upper bound. A coherent domain is a Scott-domain for which any coherent subset has an upper bound.

Coherent domains are closed for many semantics functors. In particular if D_1 and D_2 are coherent domains then $[D_1 \rightarrow D_2]$ is a coherent domain. Moreover the domain RD is coherent.

A fundamental step in the proof of universality consists in showing that for every type σ it is possible to define three functions, namely, c_σ , p_σ and $\#_\sigma$. Where c_σ and p_σ are respectively a test and a projection function for the types σ , while $\#_\sigma(n)(d)$ checks if the element d is inconsistent with the finite element $e^\sigma(n)$ (where e^σ is the effective enumeration of the finite elements of the domain D_σ ([Plo77, page 249])). Formally:

Definition 8. A partial function $f : D_{\sigma_1} \rightarrow \dots D_{\sigma_n} \rightarrow D_\sigma$ is definable in \mathcal{L}_r if there exists a closed term M such that for all $d_1 \in D_{\sigma_1} \dots d_n \in D_{\sigma_n}$ if $f(d_1) \dots (d_n)$ is defined then $\llbracket M \rrbracket_\rho(d_1) \dots (d_n) = f(d_1) \dots (d_n)$.

Definition 9. Given a coherent-domain D_σ the function $c_\sigma : B_\perp \rightarrow D_\sigma \rightarrow D_\sigma \rightarrow D_\sigma$, and the partial functions $\#_\sigma : \mathbb{Z}_\perp \rightarrow D_\sigma \rightarrow B_\perp$, $p_\sigma : \mathbb{Z}_\perp \rightarrow D_\sigma \rightarrow D_\sigma$ are defined by:

$$c_\sigma(b)(d_1)(d_2) = \begin{cases} d_1 & \text{if } b = \mathbf{tt} \\ d_2 & \text{if } b = \mathbf{ff} \\ d_1 \sqcap d_2 & \text{if } b = \perp \end{cases}$$

$$\#_\sigma(n)(d) = \begin{cases} \mathbf{ff} & \text{if } n \in \mathbb{N}, e^\sigma(n) \sqsubseteq d \\ \mathbf{tt} & \text{if } n \in \mathbb{N}, e^\sigma(n) \text{ and } d \text{ are inconsistent} \\ \text{undefined} & \text{if } n \text{ is a negative number} \\ \perp & \text{otherwise} \end{cases}$$

$$p_\sigma(n)(d) = \begin{cases} d \sqcup e^\sigma(n) & \text{if } n \in \mathbb{N}, d, e^\sigma(n) \text{ are consistent} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Lemma 1. If, in a language extending $\mathcal{L}_{PA+\exists}$ with new ground types, for every ground type τ the function $c_\tau, p_\tau, \#_\tau$ are definable by some terms $\text{pif}_\tau, P_\tau, T_\tau$ then for any other type σ the functions $c_\sigma, p_\sigma, \#_\sigma$ are definable by some suitable terms $\text{pif}_\sigma, P_\sigma, T_\sigma$.

Lemma 2. If in an extension of the language \mathcal{L} for a type σ the function p_σ is definable then every computable element in D_σ is definable.

Theorem 1. For every computable element d in D_σ there exists a closed expression M in \mathcal{L}_r such that: $\llbracket M \rrbracket_\rho = d$.

5 Operational Semantics, a First Attempt

In this section we discuss the problem of defining an operational semantics for \mathcal{L}_r . In Section 3 the elements of RD are constructed as equivalence classes of partial sequences of integers. One can use functions in $[\mathbb{Z}_\perp \rightarrow \mathbb{Z}_\perp]$ to represent sequences of integers and hence elements in RD . Following this approach one can use higher order function of $[\mathbb{Z}_\perp \rightarrow \mathbb{Z}_\perp]$ to represent functions on RD . The construction is the following. Let S' be the subset of $[\mathbb{Z}_\perp \rightarrow \mathbb{Z}_\perp]$ defined by,

$S' = \{s \mid \forall i \in \mathbb{N}. (s(i+1) \neq \perp \Rightarrow (s(i) \neq \perp \wedge 2s(i) - 1 \leq s(i+1) \leq 2(i+1)))\}$

the elements of S' define the partial sequences of digits representing elements in RD . Let $\phi' : S' \rightarrow RD$ be the function,

$$\phi'(s) = \bigcap \left\{ \left[\frac{s(i)-1}{2^i}, \frac{s(i)+1}{2^i} \right] \mid i \in \mathbb{N}, s(i) \neq \perp \right\}.$$

Given a function g on RD , for example, $g : RD \rightarrow RD \rightarrow RD$, we say that g is represented by a function $f : [\mathbb{Z}_\perp \rightarrow \mathbb{Z}_\perp] \rightarrow [\mathbb{Z}_\perp \rightarrow \mathbb{Z}_\perp] \rightarrow [\mathbb{Z}_\perp \rightarrow \mathbb{Z}_\perp]$ if for all $s_1, s_2 \in S'$, $g(\phi'(s_1))(\phi'(s_2)) = \phi'(f(s_1)(s_2))$.

The above representation for functions on RD suggests the following approach to operational semantics: for each new constant c in \mathcal{L}_r one try to find a computable function f_c on $[\mathbb{Z}_\perp \rightarrow \mathbb{Z}_\perp]$ representing the function $\llbracket c \rrbracket$. If the functions f_c would exist then a set of closed $\mathcal{L}_{PA+\exists}$ -terms M_c such that $\mathcal{E}\llbracket M_c \rrbracket_\rho = f_c$, would define an operational semantics for \mathcal{L}_r . The operational semantics would be given by the reductions rules $c \rightarrow M_c$. In fact the operational behaviour of M_c is in accordance with the denotational semantics of c . Unfortunately this natural approach is doomed to failure. In fact the function $\llbracket \text{pif}_r \rrbracket_\rho$ cannot be represented by any functional on integers. We state this negative result in a more general setting, considering not only the real number representation of Definition 1 and the corresponding domain RD but a large class of real number representations and domains of approximations.

In almost all the representations considered in the literature a real number is represented by a sequence of elements of a countable set C . For example C can be a set of digits, the set of integers, the set of p-adic rational numbers, the set of rational numbers, the set of rational intervals.

Definition 10. *A sequence representation for the real numbers is given by a countable set C , a subset S of $\mathbb{N} \rightarrow C$ and a representation function $v : S \rightarrow \mathbb{R}$. The set S is the subset of sequences defining real numbers.*

Repeating the construction of Section 3 we map finite sequences to subsets of reals.

Definition 11. *Given a sequence representation $v : S \rightarrow \mathbb{R}$, its extension to partial sequences $\bar{v} : [\mathbb{N} \rightarrow C_\perp] \rightarrow \mathcal{P}(\mathbb{R})$, is defined by,*

$$\bar{v}(s) = \{v(t) \mid t \in S, s \sqsubseteq t\}.$$

Given a sequence s and a natural number n we indicate with $s|_n$ the partial sequence containing the first n elements of s : $s|_n(m) = s(m)$ if $m \leq n$, $s|_n(m) = \perp$ otherwise. In [Wei87, pages 479–482] it has been introduced the

notion of admissible representation for real numbers. That definition can be reformulated as follows.

Definition 12. A sequence representation $\langle S, v \rangle$ is admissible if it satisfies the following conditions,

- (i) $\forall s \in S. \forall \epsilon \in \mathbb{R}. \exists n \in \mathbb{N}. \bar{v}(s|_n)$ is contained in an interval having width ϵ ,
- (ii) For each real number x there exists a sequence s such that for each n , x is contained in the interior of $\bar{v}(s|_n)$.

Condition (i) states that the function $v : S \rightarrow \mathbb{R}$ is continuous, w.r.t. the Cantor topology on S and the Euclidean topology on \mathbb{R} . Almost all the representation functions used in computable analysis are admissible.

Any sequence representation induces an information order on partial sequences: s is below t in the information order if $\bar{v}(s) \supseteq \bar{v}(t)$. We have the following negative result.

Theorem 2. For any admissible representation v , and there is no continuous functional $g : [\mathbb{N} \rightarrow C_{\perp}] \rightarrow [\mathbb{N} \rightarrow C_{\perp}] \rightarrow [\mathbb{N} \rightarrow C_{\perp}]$ such that:

- (i) g implements addition, that is: for all s, t in S , $v(g(s)(t)) = v(s) + v(t)$
- (ii) g respects the induced order relation on partial functions that is: for all s, s', t, t' in $[\mathbb{N} \rightarrow C_{\perp}]$, $\bar{v}(s) \supseteq \bar{v}(s')$ and $\bar{v}(t) \supseteq \bar{v}(t')$ implies $\bar{v}(g(s)(t)) \supseteq \bar{v}(g(s')(t'))$.

The previous theorem implies that, if we use an admissible then the operational semantics of \mathcal{L}_r cannot be given in terms of computations on sequences. This result generalises to any domain derived from an admissible representation and to any calculus define on the derived domain. There are two possible solutions to this problem. The first one consists in introducing non deterministic or intensional operators in the language. The second one consists in using representations that are not admissible, but that are suitable for real number computations. The first approach has been followed in [Esc96], there the operational semantics of a language similar to \mathcal{L}_r is given using a non deterministic operator. Here we will follow the second approach.

6 An Operational Semantics

The notations considered so far in the literature represent real numbers using sequences that are completely defined. It is possible to represent real numbers using sequences that are undefined on some elements. An example is the following.

Definition 13. A real number x in the interval $[-1, 1]$ is represented by a sequence s of digits $-1, 1$ such that: $x = \sum_{i \in \mathbb{N}} \prod_{0 \leq j \leq i} s_j / 2$

This notation is similar to the binary digit notation. The main differences consist in the use of the digit -1 instead of the digit 0 and in the fact that in this notation the value of a digit affects the weights of all the consecutive digits. In this notation the real number 0 has two representations: the sequence

$\langle -1, -1, 1, 1, 1 \dots \rangle$ and the sequence $\langle 1, -1, 1, 1, 1 \dots \rangle$. The two representations differ just for the first digit. Hence 0 can also be represented by the sequence $\langle \perp, -1, 1, 1, 1 \dots \rangle$ undefined on the first element. Moreover examining the finite initial parts of the incomplete sequence it is possible to determine the number represented by it with an arbitrary precision. Similar considerations hold for any other dyadic rational number. Every real number that is not rational dyadic has exactly one representation. If we allow as possible representations for the dyadic rational numbers also the sequences undefined on one element we obtain a representation suitable for the real number computation.

In order to represent the whole real line we consider the following notation.

Definition 14. A representation function $v : (\mathbb{N} \rightarrow \{-1, 1\}) \rightarrow \mathbb{R}$ is defined by:

$$v(s) = s(0) \times (k + \sum_{i \geq k} \prod_{k \leq j \leq i} s(j)/2)$$

where $k = \min\{i \mid i > 0, s(i) = -1\}$

This is a sort “sign, integer part, mantissa” notation for the real numbers. The first digit gives the sign, the next consecutive positive digits determine the integer part, the remaining part of the sequence is the mantissa. Also in this case every dyadic rational number is represented by two functions that differ just for one element and every real number that is not rational dyadic has exactly one representation.

Definition 15. The extension of v to partial functions is the function $\bar{v} : (\mathbb{N} \rightarrow \{-1, 1\}_\perp) \rightarrow \mathcal{P}(\mathbb{R})$ defined by:

$$\bar{v}(s) = \{v(t) \mid t : \mathbb{N} \rightarrow \{-1, 1\}, s \sqsubseteq t\}.$$

The set $\bar{v}(s)$ is an interval if and only if

$$\begin{aligned} & \forall n. (s(n) \uparrow \wedge s(n+1) \downarrow) \\ & \Rightarrow \forall m < n. s(m) \downarrow \wedge s(n+1) = -1 \wedge \forall m > n+1. (s(m) \uparrow \vee s(m) = 1). \end{aligned}$$

Let S^∞ denote the set of partial functions s such that $\bar{v}(s)$ is an interval. S^∞ is a complete partial order. If we repeat the construction of Section 3, with the representation v and the set S^∞ of partial elements we obtained a new domain for real numbers. We call the new domain RD' . In this case no pair of elements in S^∞ contain the same information. It follows that S^∞ and RD' are isomorphic. The structures of RD and RD' are quite similar. The main difference consists in the fact that RD' contains for each natural number n the intervals $[-\infty, -n]$ and $[n, +\infty]$ and, as a consequence, the infinite points $-\infty$ and $+\infty$.

Proposition 3. There exists an effective embedding-projection pair $\langle e, p \rangle$ from S^∞ to $[N \rightarrow \{-1, 1\}_\perp]$, $p : [N \rightarrow \{-1, 1\}_\perp] \rightarrow S^\infty$ is defined by:

$$p(s) = \bigsqcup \{s' \in S^\infty \mid s' \sqsubseteq s\}$$

$e : S^\infty \rightarrow [N \rightarrow \{-1, 1\}_\perp] \rightarrow S^\infty$ is the identity functions.

It follows that there exists an effective embedding-projection pair $\langle e_r, p_r \rangle$ from RD' to $[Z_\perp \rightarrow \{\text{tt}, \text{ff}\}_\perp]$. The embedding-projection can be extended to the functions spaces.

$$\begin{aligned} e_{\sigma \rightarrow \tau}(f) &= e_\tau \circ f \circ q_\sigma \\ q_{\sigma \rightarrow \tau}(f) &= q_\tau \circ f \circ e_\sigma \end{aligned}$$

Repeating the considerations presented in Section 5, it is possible to represent elements in RD' by their embeddings in $[Z_\perp \rightarrow \{\text{tt}, \text{ff}\}_\perp]$ and functions on RD' (S^∞) by the corresponding embeddings on functions spaces of $[Z_\perp \rightarrow \{\text{tt}, \text{ff}\}_\perp]$. Let C be the set of the new constants in \mathcal{L}_r , for each $c^\sigma \in C$ let M_{c^σ} be a term in $\mathcal{L}_{PA+\exists}$ defining the function $e_\sigma(\llbracket c^\sigma \rrbracket_\rho)$. By the universality of $\mathcal{L}_{PA+\exists}$ the terms M_{c^σ} exists. An operational semantics for \mathcal{L}_r can be given adding to the set single-step reduction rules for $\mathcal{L}_{PA+\exists}$ the new set of rules $\{c \rightarrow M_c \mid c \in C\}$. For lack of space we do not present the actual set of rules.

References

- [BC90] H.-J. Boehm and R. Cartwright. Exact real arithmetic: formulating real numbers as functions. In David Turner, editor, *Research topics in functional programming*, pages 43–64. Addison-Wesley, 1990.
- [BCRO86] H.-J. Boehm, R. Cartwright, M. Riggle, and M.J. O'Donnell. Exact real arithmetic: a case study in higher order programming. In *ACM Symposium on lisp and functional programming*, 1986.
- [DG93] P. Di Gianantonio. *A functional approach to real number computation*. PhD thesis, University of Pisa, 1993.
- [DG96] P. Di Gianantonio. Real number computability and domain theory. *Information and Computation*, 127(1):11–25, May 1996.
- [EE96] A. Edalat and M. Escardo. Integration in real pcf. In *IEEE Symposium on Logic in Computer Science*, 1996.
- [Esc96] M. Escardo. Pcf extended with real numbers. *Theoret. Comput. Sci.*, July 1996.
- [Lac59] D. Lacombe. Quelques procédés de définitions en topologie récursif. In *Constructivity in mathematics*, pages 129–158. North-Holland, 1959.
- [ML70] P. Martin-Löf. *Note on Constructive Mathematics*. Almqvist and Wiksell, Stockholm, 1970.
- [MM] V. Ménissier-Morain. Arbitrary precision real arithmetic: design and algorithms. Submitted to the Journal of Symbolic Computation. Available at <http://pauillac.inria.fr/~menissier>.
- [Plo77] G.D. Plotkin. Lcf considered as a programming language. *Theoret. Comput. Sci.*, 5:223–255, 1977.
- [Sco70] Dana Scott. Outline of the mathematical theory of computation. In *Proc. 4th Princeton Conference on Information Science*, 1970.
- [Str94] T. Streicher. A universality theorem for pcf with recursive types, parallel-or and \exists . *Mathematical Structures for Computing Science*, 4(1):111–115, 1994.
- [Vui88] J. Vuillemin. Exact real computer arithmetic with continued fraction. In *Proc. A.C.M. conference on Lisp and functional Programming*, pages 14–27, 1988.
- [Wei87] K. Weihrauch. *Computability*. Springer-Verlag, Berlin, Heidelberg, 1987.