Real PCF extended with $\exists$ is universal (EXTENDED ABSTRACT*)

Martín Hötzel Escardó
Department of Computing, Imperial College, London SW7 2BZ.

Friday 21st June 1996

Abstract

Real PCF is an extension of the programming language PCF with a data type for the real line, introduced elsewhere. We show that Real PCF extended with $\exists$ is universal, in the sense that all computable elements of all types of its universe of discourse are definable. We also show that $\exists$ is not necessary to define first order computable functions in Real PCF. In order to obtain our definability results, we consider a domain-equation-like structure on the real numbers data type.
Keywords and phrases: Real number computability, λ-calculus, domain theory.

Introduction

Real PCF is an extension of the programming language PCF [13] with a ground type for the real line [8]. Real PCF has denotational and operational semantics, related by an adequacy relation.

The real numbers data type is interpreted as a domain of compact real intervals, with subspace of maximal elements homeomorphic to the real line. It is natural to effectively present this domain by enumerating its basis of rational intervals in some standard way. Then all Real PCF definable elements are computable. However, as in PCF, the converse fails. For instance, a computable function $\exists : [\mathcal{N} \to \mathcal{T}] \to \mathcal{T}$ is not definable. We show that Real PCF extended with $\exists$ is *universal*, in the sense that all computable elements and functions of its universe of discourse are definable. We do not need to add parallel-or, because Real PCF already has a parallel conditional.

It is natural to ask if the notion of computability depends on the choice of bases and their enumerations, and indeed this is the case [10]. We say that an effective presentation of the Real PCF domains is *sound* if every Real PCF definable element is computable. The idea is that the definable elements are concretely computable, because Real PCF has

*To appear in Proceedings of the 3rd Theory and Formal Methods Workshop, A. Edalat, I.S. Jourdan, G.A. McCusker eds., IC Press

an effective operational semantics. We show that any two sound effective presentations of the Real PCF domains induce the same notion of computability. In this sense, Real PCF extended with $\exists$ is *absolutely universal*.

In order to establish universality, we use a technique due to Thomas Streicher [18], originally used by him to establish that PCF extended with recursive types, parallel-or and $\exists$ is universal. The technique consists of taking a universal PCF domain $\mathcal{U}$, and showing that (in our case) every Real PCF domain is a definable retract of $\mathcal{U}$. Since PCF extended with parallel-or and $\exists$ is universal [13], all computable $x \in \mathcal{U}$ and all computable $f : \mathcal{U} \to \mathcal{U}$ are definable, and the universality result follows.

In order to obtain our definability results, we consider a domain-equation-like structure on the real numbers data type.

The Real PCF notion of computability induces classical notions of computability on real numbers and real valued functions of real variables [3, 11, 12, 16, 19, 20], but this material is not included in this extended abstract due lack of space. Also, we only consider the unit interval type of Real PCF, although we indicate how the type for the whole real line can be handled.

Contents

1 The real numbers domains

Our main reference to domain theory is [1]; see also [15, 9]. We use several results from [14]. In this paper, a *domain* is a bounded complete, countably based continuous cpo. Domains are implicitly considered as topological spaces under their Scott topology.

Plotkin [14] defines a *coherent domain* to be a domain in which every pairwise consistent subset has a join. Notice that the word *coherence* has been used in several different senses in theoretical computer science and related fields.

We denote by **CDom** the category of coherent domains and continuous maps. This category is cartesian closed, and closed under lifting, amalgamated sum, and smash product. It contains the flat domains and the domains $\mathcal{R}_\perp$ and $\mathcal{I}$, where $\mathcal{R}$ is the continuous dcpo of compact real intervals ordered by superset, and $\mathcal{I}$ is the subdomain of intervals contained in the unit interval $[0, 1]$. Moreover, **CDom** is closed under the formation of retracts and canonical solutions of domain equations involving the above constructions. Thus, **CDom** contains all domains of interest for the purpose of this work.

Let $\mathcal{T} = \{\mathsf{tt}, \mathsf{ff}\}_\perp$ be the domain of truth values. In [14] it is shown that $\mathcal{T}^\omega$ is a *universal coherent domain*, in the sense that a domain is coherent iff it is a retract of $\mathcal{T}^\omega$.

Notice that there are two notions of universality in this paper.

The elements of the domains $\mathcal{I}$ and $\mathcal{R}_\perp$ are considered as "partial real numbers", and their maximal elements (singleton intervals) are identified with real numbers. This identification makes sense from a topological view, because the subspace of maximal elements of $\mathcal{R}_\perp$ is homeomorphic to the Euclidean real line. We thus sometimes notationally identify real numbers and singleton intervals.

The reason for introducing partial numbers is similar to the reason for introducing partial functions $\mathbb{N}^k \to \mathbb{N}$ in recursion theory. For instance, the set of computable real numbers is countable but not r.e. [20], whereas the set of computable partial real numbers is r.e., because the set of computable elements of any domain is r.e. [6, 15]. In particular, no programming language with a real numbers data type can define all computable real numbers without having some divergent programs of real number type. As opposed to natural numbers, in which divergence corresponds to $\perp$, divergence is a matter of degree for real numbers; $\perp$ is in the worst possible degree, maximal partial numbers are in the best possible degree (no divergence at all), and the remaining partial real numbers are in between.

Let $f : \mathbb{R} \to \mathbb{R}$ be continuous, and define $\mathbf{I}f : \mathcal{R} \to \mathcal{R}$ by $\mathbf{I}f(x) = f(x)$, for all $x \in \mathcal{R}$. This function is well-defined, because continuous functions preserve compactness and connectedness, so that $f(x)$ is a compact interval if x is a compact interval. Moreover, it is continuous w.r.t. the Scott topology [4]. Similarly, for $f : [0,1] \to [0,1]$ continuous, the above equation also defines a continuous function $\mathbf{I}f : \mathcal{I} \to \mathcal{I}$. The function $\mathbf{I}f$ is an *extension* of f, in the sense that $\mathbf{I}f(\{x\}) = \{f(x)\}$. A continuous function has uncountably many continuous extensions. The extension $\mathbf{I}f$ is characterized as the greatest one. If $f : \mathbb{R} \to \mathbb{R}$ is *non-decreasing* then the greatest extension is given pointwise, in the sense that $\mathbf{I}f(x) = [f(\underline{x}), f(\overline{x})]$, where $\underline{x} = \inf x$ and $\overline{x} = \sup x$ are the left and right end-points of the interval x. When there is no danger of ambiguity, we write f instead of $\mathbf{I}f$ or $\mathbf{I}_\perp f$.

For $x, y \in \mathcal{R}_\perp$, we define $(x < y) \in \mathcal{T}$ by

$$(x < y) = \begin{cases} \mathsf{tt} & \text{if } \overline{x} < \underline{y}, \\ \mathsf{ff} & \text{if } \underline{x} > \overline{y}, \\ \perp & \text{otherwise.} \end{cases}$$

This definition reflects the fact that equality of real numbers is not decidable [20]. The map $(x, y) \mapsto (x < y)$ can be regarded as the best continuous approximation of the characteristic function of the inequality predicate $<$ on real numbers.

Recall that the *parallel* conditional $\mathrm{pif} : \mathcal{T} \times D \times D \to D$ is defined by

$$\text{pif } t \text{ then } y \text{ else } z \quad = \quad \begin{cases} y & \text{if } t = \mathsf{tt}, \\ z & \text{if } t = \mathsf{ff}, \\ y \sqcap z & \text{if } t = \perp. \end{cases}$$

Proposition 1.1 *Let R be any domain with* $\mathrm{Max}(R)$ *homeomorphic to the real line or the unit interval, let D be a 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 $f(x) =$ if $p(x)$ then $g(x)$ else $h(x)$. If there are maximal elements x and y such that $p(x) = \mathsf{tt}$ and $p(y) = \mathsf{ff}$, then $f(z) = \bot$ for some maximal element z.

Proof The non-empty disjoint sets $U = p^{-1}(\mathsf{tt}) \cap \mathrm{Max}(R)$ and $V = p^{-1}(\mathsf{ff}) \cap \mathrm{Max}(R)$ are open in $\mathrm{Max}(R)$, because p is continuous, and $\{\mathsf{tt}\}$ and $\{\mathsf{ff}\}$ 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 $p(z) = \bot$. □

Thus, the sequential conditional is not appropriate for definition by cases of total functions on the real numbers domains, because it produces non-total functions for non-trivial continuous predicates.

In most definitions by cases of the form $f(x) = \mathrm{pif}\ p(x)$ then $g(x)$ else $h(x)$ which occur in practice, one has that $g(x) = h(x)$ for all maximal x with $p(x) = \bot$. In such a situation, if x is maximal and $p(x) = \bot$, then $f(x) = g(x) \sqcap h(x) = g(x) = h(x)$. An example is given by the following definition of the absolute value function: $|x| = \mathrm{pif}\ x < 0$ then $-x$ else x. For the case $x = 0$ one has $|0| = \mathrm{pif}\ \bot$ then -0 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.

A domain-equation-like structure for the real numbers domains

Define continuous non-decreasing functions $\mathrm{cons}_L, \mathrm{cons}_R, \mathrm{tail}_L, \mathrm{tail}_R : [0,1] \to [0,1]$ by

$$\begin{array}{rclrcl} \mathrm{cons}_L(x) & = & x/2, & \mathrm{tail}_L(x) & = & \min(2x, 1), \\ \mathrm{cons}_R(x) & = & (x+1)/2, & \mathrm{tail}_R(x) & = & \max(0, 2x-1), \end{array}$$

and define $\mathrm{head} : \mathcal{I} \to \mathcal{T}$ by $\mathrm{head}(x) = (x < 1/2)$, where the inequality map was defined in Section 1. For motivation and a detailed discussion about these functions see [8, 7, 5], where effective computation rules for them are given.

The main property of the functions $\mathrm{cons}_L, \mathrm{cons}_R, \mathrm{tail}_L, \mathrm{tail}_R : \mathcal{I} \to \mathcal{I}$ and $\mathrm{head} : \mathcal{I} \to \mathcal{T}$ is given by the following lemma:

Lemma 1.2 *The identity of $\mathcal{I}$ is the unique continuous function $f : \mathcal{I} \to \mathcal{I}$ such that $f(x) = \mathrm{pif}\ \mathrm{head}(x)$ then $\mathrm{cons}_L(f(\mathrm{tail}_L(x)))$ else $\mathrm{cons}_R(f(\mathrm{tail}_R(x)))$.*

Proof (Outline) That $f = \mathrm{id}$ satisfies the above equation is routinely checked by cases on the value of $\mathrm{head}(x)$ (see [8]). Let $A = \{m/2^n \in [0,1] | m, n \in \mathbb{Z}\}$ be the set of dyadic numbers in the unit interval. In order to prove that a property holds for every $x \in A$, it suffices to prove that it holds for 0 and 1, and that it holds for $x/2$ and $(x+1)/2$ whenever it holds for x. We call this principle *dyadic induction* [7]. Let B be the basis of $\mathcal{I}$ consisting of intervals with dyadic end-points. By dyadic induction on the end-points of x, we see that for any f satisfying the above equation, $f(x) = x$ for every $x \in B$. It follows that if f is continuous then f is the identity. □

The above lemma gives rise to domain-equation-like structure on $\mathcal{I}$. Recall that a *section-retraction* pair between objects X and Y of a category $\mathbf{X}$ consists of morphisms $X \underset{s}{\overset{r}{\leftrightarrows}} Y$ with $r \circ s = id_X$. In this case $s \circ r$ is an idempotent on Y and X is called a *retract* of Y.

Definition 1.1 Let $\mathbf{X}$ be a category and $\mathbf{F} : \mathbf{X} \to \mathbf{X}$ be a functor. A section-retraction pair $X \underset{s}{\overset{r}{\leftrightarrows}} \mathbf{F}X$ is $\mathbf{F}$-*stable* if $f = r \circ \mathbf{F}f \circ s$ implies $f = \mathrm{id}_X$. □

Lemma 1.3 *Define* $\mathbf{F} : \mathbf{CDom} \to \mathbf{CDom}$ *by* $\mathbf{F}D = \mathcal{T} \times D \times D$, *and define* $\mathrm{cons} = \mathrm{pif} \circ \mathbf{F}(\mathrm{id}, \mathrm{cons}_L, \mathrm{cons}_R)$ *and* $\mathrm{destr} = [\mathrm{head}, \mathrm{tail}_L, \mathrm{tail}_R]$. *Then* $\mathcal{I} \underset{\mathrm{destr}}{\overset{\mathrm{cons}}{\leftrightarrows}} \mathbf{F}\mathcal{I}$ *is an* $\mathbf{F}$-*stable section-retraction pair.*

Proof $f = \mathrm{cons} \circ \mathbf{F}f \circ \mathrm{destr}$ holds iff the equation in Lemma 1.2 holds. □

The domain $\mathcal{R}_\perp$ is treated similarly, by showing that its identity is the unique continuous function satisfying an appropriate equation.

Let $\mathbf{X}$ be a category and $\mathbf{F} : \mathbf{X} \to \mathbf{X}$ be a functor. Recall that an $\mathbf{F}$-*algebra* is a morphism $\alpha : \mathbf{F}X \to X$, and that an $\mathbf{F}$-*algebra homomorphism* from an algebra $\alpha : \mathbf{F}X \to X$ to an algebra $\beta : \mathbf{F}Y \to Y$ is a morphism $h : X \to Y$ such that $h \circ \alpha = \beta \circ \mathbf{F}h$. Dually, an $\mathbf{F}$-*coalgebra* is a morphism $\alpha : X \to \mathbf{F}X$, and an $\mathbf{F}$-*coalgebra homomorphism* from a coalgebra $\alpha : X \to \mathbf{F}X$ to a coalgebra $\beta : X \to \mathbf{F}X$ is a morphism $h : X \to Y$ such that $\mathbf{F}h \circ \alpha = \beta \circ h$. Algebras (resp. coalgebras) compose in the obvious way and form a category. Let $\mathbf{F} : \mathbf{CDom} \to \mathbf{CDom}$ be a functor. Recall that a *canonical solution* for a domain equation $D \cong \mathbf{F}D$ is given by an initial algebra $i : \mathbf{F}C \to C$, which is necessarily an isomorphism. A functor $\mathbf{F} : \mathbf{CDom} \to \mathbf{CDom}$ is *locally continuous* if for all D and E, the map $f \mapsto \mathbf{F}f : [D \to E] \to [\mathbf{F}D \to \mathbf{F}E]$ is continuous. For every locally continuous functor $\mathbf{F} : \mathbf{CDom} \to \mathbf{CDom}$, the domain equation $D \cong \mathbf{F}D$ has a canonical solution $i : \mathbf{F}C \to C$, and $i^{-1} : C \to \mathbf{F}C$ is a final coalgebra.

Proposition 1.4 *Let* $\mathbf{F} : \mathbf{CDom} \to \mathbf{CDom}$ *be a locally continuous functor,* $\mathbf{F}C \xrightarrow{i} C$ *be an initial algebra, and let* $D \underset{\beta}{\overset{\alpha}{\leftrightarrows}} \mathbf{F}D$ *be an* $\mathbf{F}$-*stable section-retraction pair. If* r *is the unique algebra homomorphism from* i *to* α, *and* s *is the unique coalgebra homomorphism from* β *to* i^{-1}, *then* $D \underset{s}{\overset{r}{\leftrightarrows}} C$ *is a section-retraction pair.*

Proof By hypothesis, $r \circ i = \alpha \circ \mathbf{F}r$ and $i^{-1} \circ s = \mathbf{F}s \circ \beta$. Hence $r \circ s = r \circ i \circ i^{-1} \circ s = \alpha \circ \mathbf{F}r \circ \mathbf{F}s \circ \beta = \alpha \circ \mathbf{F}(r \circ s) \circ \beta$. By stability, $r \circ s = \mathrm{id}_D$. □

Proposition 1.5 *Let* $\mathbf{F} : \mathbf{CDom} \to \mathbf{CDom}$ *be a locally continuous functor, and* $D \underset{\psi}{\overset{\phi}{\leftrightarrows}} \mathbf{F}D$ *be an isomorphism. Then* ϕ *is an initial algebra iff the isomorphism pair is* $\mathbf{F}$-*stable.*

2 Effectively given coherent domains

Algebraic domains have a relatively simple theory of effectivity [6, 15]. Unfortunately, this is not the case for continuous domains. Smyth [17] defines three notions of effectivity for continuous domains, namely effectively given *domain*, effectively given *M-domain*, and effectively given *A-domain*. The notions are not strictly equivalent, but they are essentially equivalent, in the sense that they can be effectively translated to each other. The notion of effectively given A-domain gives rise to a simple notion of effectively given coherent domain, as shown below. We slightly modify Smyth's definition in order to make effective presentations explicit, as Kanda and Park [10] show that it is possible to effectively present some domains in essentially different ways:

Definition 2.1 An *effective presentation of an algebraic domain D* consists of an enumeration of the finite elements of D such that (1) it is decidable whether $a \sqsubseteq b$ for finite $a, b \in D$; (2) it is decidable whether A is consistent, for an arbitrary finite set A of finite elements of D; (3) for A ranging over consistent finite sets of finite elements, the map $A \mapsto \bigsqcup A$ is recursive. An *effectively given algebraic domain* is an algebraic domain D together with an effective presentation. Let D and E be effectively given algebraic domains. An element $d \in D$ is *computable* if it is it the join of an r.e. directed set of finite elements. A continuous function $f : D \to E$ is *computable* if the relation $b \sqsubseteq f(a)$ is r.e. for a and b finite. An *effectively given A-domain* is a list (D, E, s, r) where D is a domain, E is an effectively given algebraic domain, $(s : D \to E, r : E \to D)$ is a section-retraction pair, and the idempotent $s \circ r : E \to E$ is computable. □

Notice that this definition of effectively given algebraic domain coincides with the one given in [6] and discussed in detail in [15].

An element $p \in \mathcal{T}^\omega$ is finite iff $p^{-1}(\mathsf{tt})$ and $p^{-1}(\mathsf{ff})$ are finite subsets of ω. Plotkin [14] considers the following effective presentation of $\mathcal{T}^\omega$: b_n is the unique finite $p \in \mathcal{T}^\omega$ such that $n = \sum_{i \in p^{-1}(\mathsf{tt})} 2 \cdot 3^i + \sum_{i \in p^{-1}(\mathsf{ff})} 3^i$. Then $p \in \mathcal{T}^\omega$ is computable iff $p^{-1}(\mathsf{tt})$ and $p^{-1}(\mathsf{ff})$ are r.e. subsets of ω. We call this effective presentation the *standard effective presentation of $\mathcal{T}^\omega$*, and from now on we implicitly assume the standard effective presentation of $\mathcal{T}^\omega$, unless otherwise stated. This induces effective presentations on the product $\mathcal{T}^\omega \times \mathcal{T}^\omega$ and the function space $[\mathcal{T}^\omega \to \mathcal{T}^\omega]$ (see [6, 17, 14, 15]).

It is natural to define effectivity for coherent domains as follows:

Definition 2.2 An *effective presentation of a coherent domain D* is a section-retraction pair $(s : D \to \mathcal{T}^\omega, r : \mathcal{T}^\omega \to D)$ such that $s \circ r : \mathcal{T}^\omega \to \mathcal{T}^\omega$ is computable. An *effectively given coherent domain* is a list (D, s, r) where D is a coherent domain and (s, r) is an effective presentation of D. Let (D, s_D, r_D) and (E, s_E, r_E) be effectively given coherent domains. An element $x \in D$ is *computable* if $s_D(x) \in \mathcal{T}^\omega$ is computable. A continuous function $f : D \to E$ is *computable* if $(r_D \to s_E)(f) : \mathcal{T}^\omega \to \mathcal{T}^\omega$ is computable. □

Since coherent domains form a cartesian closed category and $\mathcal{T}^\omega$ is a universal domain, we know that $\mathcal{T}^\omega \times \mathcal{T}^\omega$ and $[\mathcal{T}^\omega \to \mathcal{T}^\omega]$ are retracts of $\mathcal{T}^\omega$. In fact, $\mathcal{T}^\omega \times \mathcal{T}^\omega$ is isomorphic to $\mathcal{T}^\omega$. An isomorphism is given by $\mathrm{Pair} : \mathcal{T}^\omega \to \mathcal{T}^\omega \times \mathcal{T}^\omega$ defined by $\mathrm{Pair}(p) = \langle n \mapsto p(2n), n \mapsto p(2n+1)\rangle$. Then $(\mathcal{T}^\omega, \mathrm{Pair}^{-1}, \mathrm{Pair})$ is an effectively given coherent domain. We assume the section-retraction pair $(\mathrm{Pred} : [\mathcal{T}^\omega \to \mathcal{T}^\omega] \to \mathcal{T}^\omega, \mathrm{Fun} : \mathcal{T}^\omega \to [\mathcal{T}^\omega \to \mathcal{T}^\omega])$ constructed in [14]. Since Pred and Fun are computable, so is $\mathrm{Pred} \circ \mathrm{Fun} : \mathcal{T}^\omega \to \mathcal{T}^\omega$. Therefore $([\mathcal{T}^\omega \to \mathcal{T}^\omega], \mathrm{Pred}, \mathrm{Fun})$ is an effectively given coherent domain.

Theorem 2.1 *Effectively given coherent domains and computable functions form a cartesian closed category.*

We denote by **ECDom** the category of effectively given coherent domains and computable functions. From now on, given effective presentations of D and E, we implicitly assume the effective presentations of $D \times E$ and $[D \to E]$ constructed in the above theorem, unless otherwise stated. Let D be any effectively given coherent domain. Then fixed-point combinator $\mathrm{fix}_D : [D \to D] \to D$, and the conditionals $\mathrm{if}, \mathrm{pif} : \mathcal{T} \times D \times D \to D$ are computable. In particular, the least fixed-point of a computable function is computable.

Equivalence of effectively given coherent domains

A coherent effective presentation of $\mathcal{T}^\omega$ is given by $(\mathrm{id}_{\mathcal{T}^\omega}, \mathrm{id}_{\mathcal{T}^\omega})$. Let $f : \omega \to \omega$ be a non-recursive permutation of ω, and define $\phi : \mathcal{T}^\omega \to \mathcal{T}^\omega$ by $\phi(p) = p \circ f$. Since $\phi \circ \phi^{-1} = \mathrm{id}_{\mathcal{T}^\omega}$, (ϕ, ϕ^{-1}) is an effective presentation of $\mathcal{T}^\omega$. But this effective presentation is intuitively not "really" effective, by construction. The same phenomenon takes place for the usual notion of effectively given algebraic domain. If we define $b'_n = b_{f(n)}$, where b is the standard algebraic effective presentation of $\mathcal{T}^\omega$, it is easy to check that the axioms for effectively given algebraic domains given in definition 2.1 are satisfied for b'.

Moreover, $(\mathcal{T}^\omega, \mathrm{id}, \mathrm{id})$ and $(\mathcal{T}^\omega, \phi, \phi^{-1})$ are isomorphic objects of **ECDom**, and this is also not intuitive. In fact, ϕ, considered as a morphism $(\mathcal{T}^\omega, \mathrm{id}, \mathrm{id}) \to (\mathcal{T}^\omega, \phi, \phi^{-1})$ and as a morphism $(\mathcal{T}^\omega, \phi, \phi^{-1}) \to (\mathcal{T}^\omega, \mathrm{id}, \mathrm{id})$ gives the desired isomorphism. Again, in the category of effectively given algebraic domains and computable maps, $(\mathcal{T}^\omega, b)$ and $(\mathcal{T}^\omega, b')$ are isomorphic objects, with isomorphism also given by ϕ.

On the other hand, the identity of $\mathcal{T}^\omega$ is not computable as a morphism $(\mathcal{T}^\omega, \mathrm{id}, \mathrm{id}) \to (\mathcal{T}^\omega, \phi, \phi^{-1})$ or as a morphism $(\mathcal{T}^\omega, \phi, \phi^{-1}) \to (\mathcal{T}^\omega, \mathrm{id}, \mathrm{id})$. This *is* reasonable, and shows us that we cannot access within $\mathcal{T}^\omega$ "correctly presented" the computable elements of $\mathcal{T}^\omega$ "incorrectly presented", which are not computable in $\mathcal{T}^\omega$ "correctly presented".

We can summarize the above remarks by saying that effective domain theory does not give an absolute notion of effectivity. In this section we show that additional structure on effectively given coherent domains can be used to achieve absoluteness.

We recall the following definitions from [2], and we make explicit a notion which we call representability:

Definition 2.3 A *concrete category* over a category $\mathbf{X}$ is a category $\mathbf{A}$ together with a faithful functor $U : \mathbf{A} \to \mathbf{X}$, called the *underlying functor*. Let $(\mathbf{A}, U)$ be a concrete category over $\mathbf{X}$. If A and B are $\mathbf{A}$-objects, then an $\mathbf{X}$-morphism $f : UA \to UB$ is (A, B)-*representable* if there is a (necessarily unique) $\mathbf{A}$-morphism $\bar{f} : A \to B$ such that $U(\bar{f}) = f$, called the (A, B)-*representation* of f. The *fibre* of an $\mathbf{X}$-object X is the preordered class consisting of all $\mathbf{A}$-objects A with $U(A) = X$, ordered by: $A \leq B$ iff id_X is (A, B)-representable. Two $\mathbf{A}$-objects A and B are *equivalent*, written $A \equiv B$, if $A \leq B$ and $B \leq A$. □

Clearly, **ECDom** is concrete over **CDom**, with the underlying functor given by the forgetful functor that forgets effective presentations. Using the language of the above definition, we have that $(\mathcal{T}^\omega, \mathrm{id}, \mathrm{id})$ and $(\mathcal{T}^\omega, \phi, \phi^{-1})$ are isomorphic objects of the category **ECDom**, but inequivalent objects of the concrete category $(\mathbf{ECDom}, U)$.

Definition 2.4 We say that two effective presentations (s_1, r_1) and (s_2, r_2) of a coherent domain D are *equivalent* if (D, s_1, r_1) and (D, s_2, r_2) are equivalent objects of the concrete category $(\mathbf{ECDom}, U)$. □

It is immediate that (s_1, r_1) and (s_2, r_2) are equivalent iff $s_1 \circ r_2$ and $s_2 \circ r_1$ are computable. Notice that these two functions are inverses of each other, and hence they are automorphisms of $\mathcal{T}^\omega$.

Clearly, a morphism $f : UD \to UE$ in **CDom** is (D, E)-representable iff it is computable.

Proposition 2.2 *Let $(\mathbf{A}, U)$ be a concrete category over $\mathbf{X}$, let Y be an $\mathbf{X}$-object, and let B and B' be $\mathbf{A}$-objects in the fibre of Y. Then the following are equivalent:* (1) $B \leq B'$. (2) *For every $\mathbf{A}$-object A, if a morphism $f : UA \to Y$ is (A, B)-representable then it is (A, B')-representable.* (3) *For every $\mathbf{A}$-object C, if a morphism $f : Y \to UC$ is (B', C)-representable then it is (B, C)-representable.*

Proof (1) $\Rightarrow$ (2): Assume that $f : UA \to Y$ is (A, B)-representable, and let $\bar{f} : A \to B$ be the (A, B)-representation of f, and $\bar{\mathrm{id}}_Y : B \to B'$ be the (B, B')-representation of id_Y. Then $f' = \bar{\mathrm{id}}_Y \circ \bar{f}$ is a (B, B')-representation of f, because $U(f') = U(\bar{\mathrm{id}}_Y \circ \bar{f}) = U(\bar{\mathrm{id}}_Y) \circ U(\bar{f}) = \mathrm{id}_Y \circ f = f$. (1) $\Rightarrow$ (3): Similar. (2) $\Rightarrow$ (1): Take $A = B$ and $f = \mathrm{id}_Y$. (3) $\Rightarrow$ (1): Similar. □

This can be expressed by saying that two effective presentations of a coherent domain are equivalent iff they induce the same notion of computability on the domain. The following definition is also taken from [2]:

Definition 2.5 A concrete category $(\mathbf{A}, U)$ over $\mathbf{X}$ is called *concretely cartesian closed* provided that $\mathbf{A}$ and $\mathbf{X}$ are cartesian closed and that U preserves finite products, exponentials, and evaluation. □

Clearly, $(\mathbf{ECDom}, U)$ is concretely cartesian closed.

Proposition 2.3 *Let $(\mathbf{A}, U)$ be a concretely cartesian closed category over $\mathbf{X}$. If 1 and $1'$ are terminal objects of $\mathbf{A}$ in the same fibre then $1 \leq 1'$, and if $A \leq A'$ and $B \leq B'$ are $\mathbf{A}$-objects then $A \times B \leq A' \times B'$ and $[A' \to B] \leq [A \to B']$.*

We can thus say that equivalence is a "cartesian closed congruence".

Proposition 2.4 *Let $(\mathbf{A}, U)$ be a concretely cartesian closed category over $\mathbf{X}$, and let A and B and be equivalent $\mathbf{A}$-objects. Then any two products of A and B are equivalent, and any two exponentials of B to the power A are equivalent.*

Corollary 2.5 *Let D and E be effectively given coherent domains. Then any two effective presentations of $D \times E$ which make the projections $D \times E \to D$ and $D \times E \to E$ computable are equivalent, and any two effective presentations of $[D \to E]$ which make the evaluation map $[D \to E] \times D \to E$ computable are equivalent.*

Thus, as soon as effective presentations of D and E are specified, the effective presentations of $D \times E$ and $[D \to E]$ are implicitly specified up to equivalence. Clearly, any two effective presentations of a finite coherent domain are equivalent.

Proposition 2.6 *Define continuous maps* $\mathrm{cons} : \mathcal{T} \times \mathcal{T}^\omega \to \mathcal{T}^\omega$, $\mathrm{head} : \mathcal{T}^\omega \to \mathcal{T}$, $\mathrm{tail} : \mathcal{T}^\omega \to \mathcal{T}^\omega$, *by* $\mathrm{cons}(t,p)(0) = t$, $\mathrm{cons}(t,p)(n+1) = p(n)$, $\mathrm{head}(p) = p(0)$, *and* $\mathrm{tail}(p)(n) = p(n+1)$. *Then any two effective presentations of* $\mathcal{T}^\omega$ *which make* cons, head, *and* tail *computable are equivalent.*

Proof Similar to the proof of Proposition 3.1 below. □

This can be expressed by saying that there is a unique notion of computability for $\mathcal{T}^\omega$ such that cons, head, and tail are computable. A similar holds for $\mathcal{N}$ w.r.t the test-for-zero and successor functions.

3 Computability and definability

Proposition 3.1 *Any two effective presentations of* $\mathcal{I}$ *which make* cons_L, cons_R, head, tail_L, *and* tail_R *computable are equivalent.*

Proof By Lemma 1.2, the identity of $\mathcal{I}$ is the least fixed point of the map $F : [\mathcal{I} \to \mathcal{I}] \to [\mathcal{I} \to \mathcal{I}]$ defined by $F(f)(x) = \mathrm{pif\ head}(x)\ \mathrm{then\ cons}_L(f(\mathrm{tail}_L(x)))\ \mathrm{else\ cons}_R(f(\mathrm{tail}_R(x)))$. If D and E are effectively given coherent domains in the fibre of $\mathcal{I}$ such that cons_L, cons_R, head, tail_L, and tail_R are computable w.r.t. each of the corresponding effective presentations, then F is both $([D \to E], [D \to E])$-representable and $([E \to D], [E \to D])$-representable. Since the fixed point combinator is computable, $\mathrm{id}_\mathcal{I} = \mathrm{fix}(F)$ is both (D, E) and (E, D)-representable. Therefore D and E are equivalent. □

A similar fact holds for $\mathcal{R}_\perp$.

Lemma 3.2 *Define* $\mathbf{F} : \mathbf{CDom} \to \mathbf{CDom}$ *by* $\mathbf{F}D = \mathcal{T} \times D \times D$. *Then there is a computable initial algebra* $\phi : \mathbf{F}\mathcal{T}^\omega \to \mathcal{T}^\omega$.

Proof $\mathcal{T}^\omega$ is the canonical solution of the domain equation $D \cong \mathcal{T} \times D$. An initial algebra is given by $\mathrm{cons} : \mathcal{T} \times \mathcal{T}^\omega \to \mathcal{T}^\omega$. Since $\mathrm{Pair} : \mathcal{T}^\omega \to \mathcal{T}^\omega \times \mathcal{T}^\omega$ is an isomorphism, so is $\phi(t,p,q) = \mathrm{cons}(t, \mathrm{Pair}^{-1}(p,q))$. This isomorphism is clearly computable. A routine verification shows that (ϕ^{-1}, ϕ) is an $\mathbf{F}$-stable isomorphism pair. By Proposition 1.5, ϕ is an initial algebra. □

Proposition 3.3 *There is an effective presentation of* $\mathcal{I}$ *which makes* cons_L, cons_R, head, tail_L, *and* tail_R *computable.*

Proof (Sketch) Let $\mathbf{F} : \mathbf{CDom} \to \mathbf{CDom}$ and $\mathcal{I} \underset{\mathrm{destr}}{\overset{\mathrm{cons}}{\leftrightarrows}} \mathbf{F}\mathcal{I}$ be defined as in Lemma 1.3, $\phi : \mathbf{F}\mathcal{T}^\omega \to \mathcal{T}^\omega$ be a computable initial algebra, r be the unique algebra homomorphism from ϕ to cons, s be the unique coalgebra homomorphism from destr to ϕ^{-1}. Then (s, r) is an effective presentation of $\mathcal{I}$. $\square$

Propositions 3.1 and 3.3 can be interpreted by saying that there is a unique notion of computability for $\mathcal{I}$ such that cons_L, cons_R, head, tail_L, and tail_R are computable. A similar fact holds for $\mathcal{R}_\perp$.

Real PCF is an extension of PCF with ground types interpreted as $\mathcal{R}_\perp$ and $\mathcal{I}$, and new constants. In this extended abstract we only consider the unit interval type of Real PCF. The type for the whole real line is treated similarly. The constants for the unit interval type include constants for the primitive operations defined in Section 3 and used to recursively define the identities of $\mathcal{I}$ and $\mathcal{R}_\perp$. For more details see [8]. By Propositions 3.1 and 3.3, and Theorem 2.1, we can say that there is a unique notion of computability for Real PCF such that every Real PCF definable element is computable.

Proposition 3.4 $\exists$ *is computable but not Real PCF definable.*

Proof The inductive proof given in [13] for the fact that $\exists$ is not definable in PCF extended with parallel-or goes through for Real PCF, by adding obvious inductive steps for the Real PCF constants. $\square$

Theorem 3.5 *Real PCF extended with* $\exists$ *is universal.*

Proof $\mathcal{T}^\omega$ can be identified with the domain of strict functions $\mathcal{N} \to_\perp \mathcal{T}$. Hence $\mathcal{U} = [\mathcal{N} \to \mathcal{T}]$ is a universal domain too. Since PCF extended with $\exists$ is universal and $\mathcal{U}$ is a PCF domain, it suffices to show that every ground domain D is a definable retract of $\mathcal{U}$, in the sense that there is a section-retraction pair $(r : D \to \mathcal{U}, s : \mathcal{U} \to D)$ with s and r definable. This is immediate for $\mathcal{T}$ and $\mathcal{N}$. For $\mathcal{I}$, let s and r be defined as in Proposition 3.3, with $\phi : \mathbf{F}\mathcal{T}^\omega \to \mathcal{T}^\omega$ as defined in Lemma 3.2. Call ψ the inverse of ϕ, and define $[\psi_H, \psi_L, \psi_R] = \psi$. Then

$$\begin{aligned} r(p) &= \mathrm{cons} \circ \mathbf{F}r \circ \phi^{-1}(p) = \mathrm{cons} \circ (\mathrm{id} \times r \times r) \circ (\psi_H(p), \psi_L(p), \psi_R(p)) \\ &= \mathrm{pif}\ \psi_H(p)\ \mathrm{then}\ \mathrm{cons}_L(r(\psi_L(p))\ \mathrm{else}\ \mathrm{cons}_R(r(\psi_R(p)), \end{aligned}$$

$$\begin{aligned} s(x) &= \phi \circ \mathbf{F}s \circ \mathrm{destr}(x) = \phi \circ (\mathrm{id} \times s \times s) \circ (\mathrm{head}(x), \mathrm{tail}_L(x), \mathrm{tail}_R(x)) \\ &= \phi(\mathrm{head}(x), s(\mathrm{tail}_L(x)), s(\mathrm{tail}_R(x))). \end{aligned}$$

Since s and r are the unique continuous maps which satisfy the above equations, we can use the fixed-point combinator to recursively define them. It is easy to see that

$$\begin{aligned} \phi(t, p, q)(n) &= \mathrm{if}\ n = 0\ \mathrm{then}\ t\ \mathrm{else\ if\ odd}(n)\ \mathrm{then}\ p((n-1)/2)\ \mathrm{else}\ q((n-2)/2), \\ \psi_H(p) &= p(0), \qquad \psi_L(p)(n) = p(2n+1), \qquad \psi_R(p)(n) = p(2n+2). \end{aligned}$$

In order to get into the Real PCF universe of discourse, define $\Phi : \mathcal{T} \to \mathcal{U} \to \mathcal{U} \to \mathcal{U}$, $\Psi_H : \mathcal{U} \to \mathcal{T}$ and $\Psi_L, \Psi_R : \mathcal{U} \to \mathcal{U}$ by (almost the same equations)

$$\begin{aligned} \Phi(t)(p)(q)(n) &= \mathrm{if}\ n = 0\ \mathrm{then}\ t\ \mathrm{else\ if\ odd}(n)\ \mathrm{then}\ p((n-1)/2)\ \mathrm{else}\ q((n-2)/2) \\ \Psi_H(p) &= p(0), \qquad \Psi_L(p)(n) = p(2n+1), \qquad \Psi_R(p)(n) = p(2n+2). \end{aligned}$$

Then these functions are clearly PCF definable. Now recursively define $\alpha : \mathcal{U} \to \mathcal{I}$ and $\beta : \mathcal{I} \to \mathcal{U}$ by

$$\begin{aligned} \alpha(p) &= \text{pif } \Psi_H(p) \text{ then } \mathrm{cons}_L(\alpha(\Psi_L(p)) \text{ else } \mathrm{cons}_R(\alpha(\Psi_R(p)), \\ \beta(x) &= \Phi(\mathrm{head}(x), \beta(\mathrm{tail}_L(x)), \beta(\mathrm{tail}_R(x)). \end{aligned}$$

Then α and β are Real PCF definable too, and they clearly (co)restrict to r and s. Therefore $\alpha \circ \beta = \mathrm{id}_{\mathcal{I}}$. □

Acknowledgements

I am grateful to Thomas Streicher for letting me know his proof technique, which allowed me to fully solve the universality problem for Real PCF. I have been supervised by Mike Smyth. I have had endless discussions with him and Abbas Edalat. This work has been supported by an ARC project "A Computational Approach to Measure and Integration Theory" and the Brazilian agency CNPq.

References

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

[2] J Adamek, H Herrlich, and G.E. Strecker. *Abstract and Concrete Categories*. John Wiley & Sons, Inc., 1990.

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

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

[5] A. Edalat and M.H. Escardó. Integration in Real PCF. In *Proceedings of the Eleventh Annual IEEE Symposium on Logic In Computer Science*, New Brunswick, New Jersey, USA, July 1996.

[6] H. Egli and Constable R.L. Computability concepts for programming languages. *Theoretical Computer Science*, 2:133–145, 1976.

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

[8] M.H. Escardó. PCF extended with real numbers. *Theoretical Computer Science*, 162, To appear in 1996.

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

[10] A. Kanda and D. Park. When are two effectively given domains identical? In K. Weihrauch, editor, *Theoretical Computer Science* 4^{th} *GI Conference*, LNCS, 1979.

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

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

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

[14] G. Plotkin. $\mathcal{T}^{\omega}$ as a universal domain. *Journal of Computer and System Sciences*, 17:209–236, 1978.

[15] G. Plotkin. Domains. Post-graduate Lectures in advanced domain theory, University of Edinburgh, Department of Computer Science, 1980.

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

[17] M.B. Smyth. Effectively given domains. *Theoretical Computer Science*, 5(1):256–274, 1977.

[18] T. Streicher. A universality theorem for PCF with recursive types, parallel-or and $\exists$. *Mathematical Structures for Computing Science*, 4(1):111 – 115, 1994.

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

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