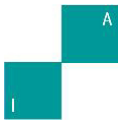

Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial
ISSN: 1137-3601
revista@aepia.org
Asociación Española para la Inteligencia Artificial
España

Marcial-Romero, J. Raymundo; Hernández, J. A.
Functional fi rst order de finability of LRTp
Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 14, núm. 48, 2010, pp. 28-40
Asociación Española para la Inteligencia Artificial
Valencia, España

Inteligencia Artificial 48(2010), 28-40
doi: 10.4114/ia.v14i48.1620



INTELIGENCIA ARTIFICIAL

http://erevista.aepia.org/

# Functional first order definability of $LRT_p$

J. Raymundo Marcial-Romero and J. A. Hernández

Facultad de Ingeniería
Universidad Autónoma del Estado de México
rmarcial,xoseahernandez@fi.uaemex.mx

**Abstract** The language $LRT_p$ is a non-deterministic language for exact real number computation. It has been shown that all computable first order relations in the sense of Brattka are definable in the language. If we restrict the language to single-valued total relations (e.g. functions), all polynomials are definable in the language. This paper is an expanded version of [12] in which we show that the non-deterministic version of the limit operator, which allows to define all computable first order relations, when restricted to single-valued total inputs, produces single-valued total outputs. This implies that not only the polynomials are definable in the language but also all computable first order functions.

**Keywords**: Exact real-number computation; Sequential Computation; PCF; Semantics of programming languages.

## 1 Introduction

Several papers on real number computation follow an idea originally due to Scott [20] of interpreting a type for real numbers in the domain of compact intervals (for simplicity, often restricted to the closed unit interval). In particular, extensions to $PCF$ following this approach are investigated in [5, 3, 19, 7, 9]. One of the most striking results along this line is Escardó, Hofmann and Streicher's proof [6] that "parallel if" can be implemented in a language that includes addition extended canonically to the domain of partial reals. This means that in order to have a reasonably expressive language with sequential interpretation, one must give up the canonical extension of addition. One way to do this is to introduce non-deterministic choice into the language. In [10, 11], the sequential, non-deterministic language LRT is defined. In those papers, it is also shown that the non-determinism must be interpreted via the Hoare power domain. So, the ground types of the language are interpreted as Hoare power domains. It is the interaction of partiality and non-determinism that characterizes the basic idea of LRT.

The first aim in the construction of LRT was to show its expressivity when restricted to single-valued total relations, e.g. functions. In that direction Marcial et al. [9, 10, 11] show that all polynomial functions are definable in the language.

LRT, with its sequential, non-deterministic semantics, seemed naturally suited to a relational view of computation. In [13] the language $\mathsf{LRT}_p$ is introduced. This languague is an extension of LRT with a `let` construct added. The interpretation of `let` is parameterized by a positive real number $p$. The extended part is used to define the recursive relations defined by Brattka [2]. The corresponding denotational semantics employs several ideas familiar to domain theorists, including measurement as defined by Martin in [14] and a monadic treatment of the distinction between value and computation as in Moggi [16]. Furthermore, product types were also included in the language to have explicit products of ground types.

**ISSN: 1988-3064(on-line)**

As a result, all computable first order relations in Brattka sense were shown to be definable in the language. However, it was not explicitly shown that all computable functions of first order type are defined using the extended language. Obviously $\mathsf{LRT}_p$ allows to define computable functions, but it has to be proved that a computable first order single-valued function when defined in the language, produces correct singled-value total outputs. In this paper we prove that it is the case.

In order to verify that all computable first order functions are definable in the language, we show that the *limit* operator when restricted to single-valued total inputs, produces single-valued total outputs. We use the argument stated by Farjudian [8] that the polynomials together with the limit operator allow to define all computable first order functions in the language. As previously mentioned, all polynomials are defined in the language $\mathsf{LRT}$ and in the extended language $\mathsf{LRT}_p$.

$\mathsf{LRT}_p$ is tied to the call-by-partial-value evaluation defined in [13], because the parameter $p$ does not allow to have a call-by-name evaluation strategy as is the case in $\mathsf{LRT}$. In particular, call-by-value simply makes no sense for the real number type in $\mathsf{LRT}_p$ because a "value" only corresponds to a converging sequence of partial results. In this paper we use the call-by-partial-value of $\mathsf{LRT}_p$.

The paper is organized as follows: after the foundations, in Section 3 we present the notions of strongly convergence of programs. In Section 4, we introduce the language $\mathsf{LRT}_p$. In Section 5, we present the program that computes the limitation (in some places called limit) operator and we prove that it strongly convergence for single-valued inputs. Finally, Section 6 is devoted to conclusions.

# 2 Foundations

## 2.1 Continuous relations

In [2], Brattka extends Kleene's system of recursive partial functions on the natural numbers to other metric spaces, particularly to $\mathbb{R}$. Continuity is a necessary condition for effectiveness, and yet the fact that $\mathbb{R}$ is connected means there are no non-constant continuous functions, e.g., from $\mathbb{R}$ to the discrete space $\mathbb{N}$. So Brattka gives up functionality and retains a generalization of continuity to relations.

**Definition 2.1.** *For binary relation $R$ between sets $X$ and $Y$ and element $x \in X$, define $R(x) := \{y \in Y \mid xRy\}$. For $B \subseteq Y$, define $R^{-1}(B) := \{x \in X \mid R(x) \cap B \neq \emptyset\}$, and let $\mathsf{dom}(R) = R^{-1}(Y) = \{x \in X \mid R(x) \neq \emptyset\}$. Thus we think of a relation as a partial function from $X$ to non-empty subsets of $Y$. For this reason, we follow Brattka by usually writing $f$, $g$, etc., as names for binary relations. Binary relations from $X$ to $Y$ will be indicated by $f\colon X \leftrightarrow Y$.*

*If $X$ and $Y$ are topological spaces, then $f$ is said to be* continuous *if and only if $f^{-1}(V)$ is open in $X$ whenever $V$ is open in $Y$. Also $f$ is said to have* closed images *if $f(x)$ is closed in $Y$ for every $x \in X$. Additionally, $f$ is said to be single-valued if $f(x)$ is a singleton in $Y$ for every $x \in X$.*

*If $X$ and $Y$ are topological (or metric) spaces, then $X \times Y$ denotes the standard topological (metric) product.*

Clearly, for a function $h$ between spaces, the graph of $h$ is a continuous relation if and only if $h$ is continuous in the usual sense. In particular, the graphs of projection maps for cartesian products are continuous. If the codomain is $T_1$, graphs of functions also have closed images. Furthermore, any relation $f$ is continuous if and only if $f(\overline{A}) \subseteq \overline{f(A)}$ for every $A \subseteq \mathsf{dom}(f)$. Note that $A$ ranges only over subsets of $\mathsf{dom}(f)$, not over all subsets of $X$. This jibes with our interpretation of $f(x) = \emptyset$ as meaning that $f$ is undefined at $x$.

Continuous relations are not closed under the usual relational composition. On the other hand, for continuous relations $f\colon X \leftrightarrow Y$ and $g\colon Y \leftrightarrow Z$, define $g \odot f$ by

$$(g \odot f)(x) := \begin{cases} \overline{(g \circ f)(x)}, & \text{if } f(x) \subseteq \mathsf{dom}(g); \\ \emptyset, & \text{otherwise.} \end{cases}$$

where $g \circ f$ is the usual relational composition. A simple exercise shows that continuous relations are closed under $\odot$.

By definition, $g \odot f$ has closed images. The graph of the identity function on a space $Y$ satisfies $f = I \odot f$ if and only if $f$ has closed images, and similarly for $g = g \odot I$. So $\odot$ defines composition for

a category of topological (or metric) spaces in which the morphisms are continuous relations with closed images. This can be taken to be the ambient category for Brattka's recursive relations. Note that the graphs of projections on products of $T_1$ spaces are continuous with closed image. So the category can be given a monoidal structure.

In addition to the $\odot$ composition, Brattka defines combinators on binary relations for juxtaposition, iteration, minimization and limitation, in this section, we only present the limitation combinator as it will be used in our later discussion.

**Limitation** In a (complete) metric space, a sequence $\{B_n\}_n$ of subsets is *strongly Cauchy* provided that for each $i$ and each $b_i \in B_i$, $b_i$ is the $i$-th element of a strong Cauchy sequence $\{b_n\}_n$ for which $b_n \in B_n$ for each $n$. In other words, all elements of $B_i$ participate in *some* strong Cauchy sequence obtained from the sets $B_n$. For such a sequence of subsets, define $\lim_{i\to\infty} B_i$ to consist of all limits (in the usual sense) of all strong Cauchy sequences $\langle b_n\rangle_n$ such that $b_n \in B_n$.

For a relation $C\colon X \times N \leftrightarrow Y$, the limitation combinator is defined by

$$\lim[C](a) := \begin{cases} \lim_{n\to\infty} C_n(a), & C_n \text{ strongly Cauchy, where } C_n(a) = C(a,n); \\ \emptyset, & \text{otherwise.} \end{cases}$$

## 2.2 The interval domain

The ideas discussed in this section are considered in more detail in [5].

The set $\mathcal{R}$ of non-empty connected compact subsets of the Euclidean real line forms a continuous dcpo when ordered by reverse inclusion: $x \sqsubseteq y$ iff $x \supseteq y$.

We regard elements of $\mathcal{R}$ as "partial real numbers"; the $\sqsubseteq$-maximal intervals are singletons, corresponding to "total numbers". That is, the continuous map $x \mapsto \{x\}$ embeds $\mathbb{R}$ as maximal elements, making $\mathcal{R}$ into a domain model for $\mathbb{R}$. The dcpo $\mathcal{R}$, however, does not have a least element. By adding a least element, corresponding to the completely under-specified partial real number $\mathbb{R}$, we obtain a bounded complete continuous domain $\mathcal{R}_\perp$.

For any $x \in \mathcal{R}_\perp$, we write $\underline{x} = \inf x$ and $\overline{x} = \sup x$ so that $x = [\underline{x}, \overline{x}]$, and define $\kappa_x := \overline{x} - \underline{x}$.

The upper bound of a subset $A \subseteq \mathcal{R}_\perp$ is $\bigcap A$ when this is not empty. Alternatively,

$$\bigsqcup A = \bigcap A = \left[\sup_{x\in A} \underline{x}, \inf_{x\in A} \overline{x}\right].$$

The way-below relation of $\mathcal{R}_\perp$ is given by $x \ll y$ iff $\underline{x} < \underline{y}$ and $\overline{y} < \overline{x}$. This amounts to $y$ being a subset of the interior of $x$. Of course $\mathbb{R} = \perp \ll a$ for any compact interval $a$. The intervals with distinct rational end-points form a basis for $\mathcal{R}_\perp$.

For basis element $a$, consider the partial function $x \mapsto a \sqcup x$ defined when $a$ and $x$ are consistent. This join map has a total continuous extension:

$$\mathsf{join}_a x = \begin{cases} a \sqcup x, & a \text{ and } x \text{ are consistent;} \\ \{\underline{a}\}, & \overline{x} < \underline{a}; \\ \{\overline{a}\}, & \overline{a} < \underline{x}. \end{cases}$$

**Lemma 2.2.** *For basis elements $a$ and $b$,*

1. $\mathsf{join}_a\mathsf{join}_b = \mathsf{join}_{a\sqcup b}$ *if $a \sqcup b$ exists;*
2. $\mathsf{join}_a\mathsf{join}_b = k_{\underline{a}}$ *if $\overline{b} < \underline{a}$;*
3. $\mathsf{join}_a\mathsf{join}_b = k_{\overline{a}}$ *if $\overline{a} < \underline{b}$;*

*where $k_x$ denotes the constant map $x \mapsto \{x\}$. Thus $\mathsf{join}_a \sqsubseteq \mathsf{join}_a\mathsf{join}_b$ always holds.*

*Proof.* We present the proof of case (1) when $b$ is consistent with any given input $x$. The other cases are similar.

$$\begin{aligned} \mathsf{join}_a\mathsf{join}_b(x) &= \mathsf{join}_a(x \sqcup b) \\ &= a \sqcup (x \sqcup b) \\ &= (a \sqcup b) \sqcup x \\ &= \mathsf{join}_{a \sqcup b}(x) \end{aligned}$$

□

Each basis element $a$ is also associated with a positive affine map $\mathsf{rrcons}_a : \mathbb{R} \to \mathbb{R}$ given by $x \mapsto \kappa_a x + \underline{a}$. Taking images, $\mathsf{rrcons}_a$ extends to a strict continuous map on $\mathcal{R}_\perp$. These maps form a left group action on $\mathcal{R}_\perp$. Because of this, we will think of the basis of $\mathcal{R}_\perp$ as itself forming a group, writing $ab$ for concatenation, $a^{-1}$ for inverse and $I$ for the identity (that is, the interval $[0, 1]$, corresponding to the identity affine map).

Composites of joins and affine transformations interact as follows:

**Lemma 2.3.** *For basis elements $a$ and $b$,*

1. $\mathsf{rrcons}_a\mathsf{join}_b = \mathsf{join}_{ab}\mathsf{rrcons}_a$;
2. $\mathsf{rrcons}_a\mathsf{rrcons}_b = \mathsf{rrcons}_{ab}$;

*Proof.* 1.

$$\begin{aligned} \mathsf{rrcons}_a\mathsf{join}_b(x) &= \mathsf{rrcons}_a(\mathsf{join}_b(x)) \\ &= \kappa_a(\mathsf{join}_b(x)) + \underline{a} \\ &= \mathsf{join}_{\kappa_a(b)+\underline{a}}(\kappa_a(x) + \underline{a}) \\ &= \mathsf{join}_{ab}(\mathsf{rrcons}_a(x)) \end{aligned}$$

2. An straightforward algebraic manipulation of the expresion.

□

The functions $\mathsf{rrcons}_a$ and $\mathsf{join}_a$ are said to be *strongly convergent*, meaning that they send maximal elements to maximal elements. In addition, the functions $\mathsf{rrcons}_a$ are all homeomorphisms ($\mathsf{rrcons}_a\mathsf{rrcons}_{a^{-1}} = \mathsf{rrcons}_I = \mathsf{id}$), so they also send non-maximal elements to non-maximal elements.

## 2.3 The Hoare powerdomain

In [9, 10, 11], the first author shows that under certain assumptions, a suitable semantics for sequential, non-deterministic real number computation requires the Hoare powerdomain ($\mathcal{P}^H$). That is, starting from the assumption that *some* functorial powerdomain is needed to model non-determinism, general considerations about continuity show that the Hoare powerdomain is the only one that can be used. We refer the reader to the cited work for an explanation. In that work, however, the fact that $\mathcal{P}^H$ is actually a free construction is not used explicitly (though certain definitions in the semantics depend on it implicitly). In this section, we review the basic facts about $\mathcal{P}^H$ as the construction of free inflationary semi-lattices. The reader may consult [1] for a general theory of free domain constructions defined by inequalities.

A *semi-lattice* in the category of domains is simply a domain equipped with a continuous binary operation $\uplus : X \times X \to X$ that satisfies the usual semi-lattice laws. Such a semi-lattice is *inflationary* if $x \sqsubseteq x \uplus y$. It is not hard to see that idempotency is equivalent to $\uplus \circ \delta = \mathsf{id}_X$, and inflationarity to $\mathsf{id}_{X \times X} \sqsubseteq \delta \circ \uplus$, where $\delta : X \to X \times X$ is the diagonal map. Since these two conditions constitute a Galois connection between $\uplus$ and $\delta$, if $\uplus$ exists it is unique.

The Hoare powerdomain is the free construction for inflationary semi-lattices [1]. If $f\colon X \to Y$ is a continuous map and $(Y, \uplus)$ is an inflationary semi-lattice, then there is a unique continuous map $\overline{f}\colon \mathcal{P}^H(X) \to Y$ that preserves $\uplus$ for which $f = \overline{f}\eta$, where $\eta$ is the unit of the powerdomain monad. There is also a unique continuous map $\widehat{f}\colon \mathcal{P}^H(X) \to \mathcal{P}^H(Y)$ defined by $\widehat{f} := \mathcal{P}^H(f)$.

In domains, the binary formal join of an inflationary semi-lattice extends automatically to formal joins of non-empty sets: For $A \subseteq X$, take the closure of $A$ under $\uplus$. This is automatically a directed set and hence has a supremum, which we denote by $\bigcup A$. If the generating domain has a least element, then so does $\mathcal{P}^H(X)$. So $\bigcup$ is defined for all subsets of $\mathcal{P}^H(X)$.

Concretely, elements of $\mathcal{P}^H(X)$ are non-empty Scott closed subsets of $X$, the unit sends $x \in X$ to the closure of $\{x\}$. Also, $\uplus$ is simply binary union, and $\bigcup$ is closure of union.

To mediate between products and powerdomains, we exploit the fact that the Hoare powerdomain is a monoidal monad with natural transformation $m\colon \mathcal{P}^H(X) \times \mathcal{P}^H(Y) \to \mathcal{P}^H(X \times Y)$ satisfying the usual coherence conditions. In concrete terms, $m(A, B) := A \times B$.

Thus the relevant structure of the Hoare powerdomain, for our purposes, is given by the functor $\mathcal{P}^H$ itself, the unit $\eta : X \to \mathcal{P}^H(X)$, the formal union $\uplus\colon \mathcal{P}^H(X) \times \mathcal{P}^H(X) \to \mathcal{P}^H(X)$ and the transformation $m : \mathcal{P}^H(X) \times \mathcal{P}^H(Y) \to \mathcal{P}^H(X \times Y)$.

A continuous map $f$ between inflationary semi-lattice domains $X$ and $Y$ *preserves* $\uplus$ if $f(x \uplus y) = f(x) \uplus f(y)$.

# 3 Hoare power domains of domain environments specialized to functions

If $D$ and $E$ are domain environments for spaces $X$ and $Y$ respectively, we can ask when a continuous function $F\colon \mathcal{P}^H(D) \to \mathcal{P}^H(E)$ corresponds naturally to a continuous single-valued relation from $X$ to $Y$. The next definition answer this question.

**Definition 3.1.** *Suppose $X$ is a topological space, $E_X$ is a domain model for $X$ with embedding $e_X$ and $d \in \mathcal{P}^H(E_X)$. Let*

$$u_X(d) := \{x \in X \mid \nu_X(x) \sqsubseteq d\} = (\nu_X)^{-1}(\downarrow d) \qquad \nu_X := \eta \circ e_X.$$

*We said that $u$ is single-valued if there is a unique $x \in X$ such that $u_X(d) = \{x\}$. The subscripts will be ommited when possible.*

*Furthermore, suppose that $Y$ is a second space and $E_Y$ is a corresponding domain model. It is said that a relation $f$ from $X$ to $Y$ is* captured *by $F\colon \mathcal{P}^H(E_X) \to \mathcal{P}^H(E_Y)$ (written $f \sim F$) iff for each $x \in dom(f)$, $f(x) = u(F(\nu_X(x)))$. Moreover, say that a single-valued relation $f$ from $X$ to $Y$ is* strongly captured by $F\colon \mathcal{P}^H(E_X) \to \mathcal{P}^H(E_Y)$ *(written $f \overset{\otimes}{\sim} F$) if and only if for each $x \in \mathsf{dom}(f)$, $f(x) = u(F(\nu_X(x)))$, i.e. $u$ is single-valued. Say that $f$ is exactly captured by $F$ (written $f \simeq F$) iff*

$$f \sim F \text{ and } dom(f) = \{x \in X \mid u(F(\nu(x))) \neq \emptyset\}.$$

*Say that $f$ is* exactly strongly captured by $F$ *(written $f \overset{\otimes}{\simeq} F$) if and only if $f \overset{\otimes}{\sim} F$ and*

$$\mathsf{dom}(f) = \{x \in X \mid u(F(\nu(x))) \neq \emptyset, u \text{ is single-valued}\}.$$

*Say that $d \in \mathcal{P}^H(E_X)$ is* convergent *provided that $d = \bigcup \{\nu(x) \mid x \in u(d)\}$. Say that $d \in \mathcal{P}^H(E_X)$ is* strongly convergent *provided that $d = \nu(x)$ where $x$ is the unique value of the single-valued function $u(d)$. Also say that $d$ is* divergent *provided that $\nu(x) \not\sqsubseteq d$ for all $x \in X$. Moreover, say that $d$ is* strongly divergent *provided that either $\nu(x) \not\sqsubseteq d$ for all $x \in X$ or $u(d)$ is not single-valued. Say that continuous $F\colon \mathcal{P}^H(E_X) \to \mathcal{P}^H(E_Y)$ is* disciplined *provided that it preserves $\uplus$ and for each $x \in X$, $F(\nu_X(x))$ is either convergent or divergent. Also, say that continuous $F\colon \mathcal{P}^H(E_X) \to \mathcal{P}^H(E_Y)$ is* strongly disciplined *provided that for each $x \in X$, $F(\nu_X(x))$ is either strongly convergent or strongly divergent.*

For the remainder of this section, we assume that $X$, $Y$, $E_X$, $E_Y$, and embeddings $e_X$, $e_Y$ are fixed.

We can verified that, for any singleton $\{x\} \subseteq X$, $\{x\} = u(\nu(x))$, to see this, $\nu(x)$ is a directed set with supremum $x$, hence $x \in u(\nu(x))$, as $x$ is the only element belonging to $\{x\}$ then $\{x\} = u(\nu(x))$.

In [13] it was proved that for any $F\colon \mathcal{P}^H(E_X) \to \mathcal{P}^H(E_Y)$, there is a unique relation that is exactly captured by $F$. If $F$ is disciplined, the exactly captured relation is a continuous relation with closed images.

**Lemma 3.2.** *For any continuous $F\colon \mathcal{P}^H(E_X) \to \mathcal{P}^H(E_Y)$, if $F$ is strongly disciplined, there is a unique relation that is exactly strongly captured by $F$, this relation is a continuous relation with single-valued image.*

*Proof.* The relations exactly strongly captured by $F$, understood as subsets of $X \times Y$ are closed under unions. So there is a maximal relation exactly strongly captured by $F$. Clearly, the condition on domains means that this maximal relation is exactly strongly captured and any exactly strongly captured relation is contained in the maximal exactly strongly captured relation.

Suppose $F$ is strongly disciplined. Consider the composition $u \odot F$. We claim that this is a continuous relation ($u$ itself is not generally continuous) with single-valued image. For open $U$, the inverse image is

$$\{d \in E_X \mid u(F(d)) \cap U \neq \emptyset, u \text{ single-valued}\}$$

which is clearly an upper subset of $E_X$. For directed $D$, if $u(F(\bigsqcup D)) \cap U \neq \emptyset$, and $u$ is single-valued then for some $x \in X$, $\eta(e_X(x)) \sqsubseteq \bigsqcup F(D)$.

Define $f\colon X \leftrightarrow Y$ by $f(x) = u(F(\eta(e_X(x))))$ for single valued $u$. Because $u$ only yields singletons, $f$ has single-valued images. As, F is strongly disciplined, $\mathsf{dom}(f)$ agrees precisely with the definition of "exact strongly capture." It remains to verify that $f \overset{\otimes}{\sim} F$. Suppose $\{x\} = u(d)$. Then $d = \eta(e_X(x)) \uplus d'$ where $u(d') = \emptyset$. As $F$ is strongly disciplined, $u(F(d)) = u(F(\eta(e_X(x)))) = f(x)$. □

There is a fundamental connection between continuous relations with closed images and disciplined functions. Disciplined functions are closed under composition. Moreover, if $F$ and $G$ are disciplined, $f$ and $g$ are continuous with closed images, $f \sim F$ and $g \sim G$ and these "type check" in the obvious way, then $(g \odot f) \sim (G \circ F)$ [13].

The previous result can be extended to continuous relatons with single-valued image and strongly disciplined functions.

**Theorem 3.3.** *Strongly disciplined functions are closed under composition. Moreover, if $F$ and $G$ are strongly disciplined, $f$ and $g$ are continuous with single-valued images, $f \overset{\otimes}{\sim} F$ and $g \overset{\otimes}{\sim} G$ and these "type check" in the obvious way, then $(g \odot f) \overset{\otimes}{\sim} (G \circ F)$.*

*Proof.* Closure under composition for preservation of $\uplus$ follows from the general theory of power domains. By definition of strongly disciplined, there is a unique $z$ such that $\{z\} = u(G(F(\nu(x))))$ for all $x$ in the domain of $f$, hence $z \in G(F(\nu(x)))$.

Conversely, if there are unique $y$ and $z$ such that $\{z\} = u(G(\nu(y)))$ and $\{y\} = u(F(\nu(x)))$ then $\nu(z) \sqsubseteq G(\nu(y)) \sqsubseteq G(F(\nu(x)))$. So $G \circ F$ is strongly disciplined.

The second statement is now routinely checked. □

Strongly discipline is related to the operational concept of strong convergence discussed at length in [11]. There a closed *term* of ground type is strongly convergent if it denotes $\nu(x)$ for some $x$ in the modeled space (although the definition is given operationally and adequacy of the operational semantics justifies the present characterization). A closed first-order term is strongly convergent if it preserves strong convergence of inputs. The reason an operational definition is given is that proof of strong convergence typically involves the operational semantics. The reader may consult [9], [10] or [11] for discussion and examples.

The ground spaces about which we are concerned have additional structure that allow a form of call-by-value, which we refer to as *call-by-partial-value*. In [15], Martin introduces the concept of a *measurement* on a continuous domain, $D$, as a Scott continuous function $M\colon D \to ([0,\infty], \geq)$. That is, $M$ assigns a positive extended real to each element of $D$ so that $M(\bigsqcup A) = \inf_{a \in A} M(a)$ for directed $A$. A measurement is also required to satisfy $M(a) = 0$ if and only if $a \in \max D$.

The domains $\mathcal{R}_\perp$, $\mathbb{T}_\perp$ and $\mathbb{N}_\perp$ clearly can be equipped with measurements: in $\mathcal{R}_\perp$, $M(a) = \kappa_a$; in $\mathbb{T}_\perp$, $M(\mathsf{true}) = M(\mathsf{false}) = 0$; in $\mathbb{N}_\perp$, $M(n) = 0$; and in all of these $M(\perp) = \infty$. In a finite product of domains with measurements, a measurement on a tuple is obtained by taking the minimum measurement of the components. For any positive $p$, any domain $D$ with least element and with measurement $M$, the function $\mathsf{pv}_p \colon D \to D$ given by $\mathsf{pv}_p(a) = a$ if $M(a) < p$ and $\mathsf{pv}_p(a) = \perp$ otherwise is continuous. Its extension to $\mathcal{P}^H D$ satisfies $\widehat{\mathsf{pv}_p}(d) \sqsubseteq d$ and allows us to isolate the maximal part of an element $d \in \mathcal{P}^H(D)$ that can be written $\bigsqcup_{a \in A} \eta(a)$ where all elements of $A$ have "small" measurement. As $p$ decreases, $\mathsf{pv}_p$ decreases as well. Importantly, each $\widehat{\mathsf{pv}_p}(d)$ is the identity map when restricted to convergent $d$, and $\bigsqcup_p \widehat{\mathsf{pv}_p}$ is the identity on $\mathcal{R}_\perp$.

# 4 The $\mathsf{LRT}_p$ Language

The language $\mathsf{LRT}$ is a modification of $\mathsf{RealPCF}$ considered by Escardó [4] for real number computation. In $\mathsf{LRT}$, parallel conditional `pif` is replaced by a non-deterministic test $\mathtt{rtest}_{l,r}$. In this section, we describe $\mathsf{LRT}_p$ a variant of $\mathsf{LRT}$. The language $\mathsf{LRT}_p$ differs from $\mathsf{LRT}$ in three ways: products of ground types are made explicit, the type $I$ for the compact interval $[0, 1]$ is eliminated in favor of a type corresponding to $\mathbb{R}$, and a `let` construct is introduced that provides for call-by-partial-value semantics. This language is described at length in [13].

## 4.1 Syntax

Syntactically, the type system for $\mathsf{LRT}_p$ is given by

$$\begin{array}{lcl} \gamma & := & \mathtt{nat} \mid \mathtt{bool} \mid \mathtt{real} \\ \beta & := & \gamma \mid \gamma \times \beta \\ \tau & := & \beta \mid (\tau \to \tau) \end{array}$$

Types in the first clause are *ground types*; in the second clause, *basic types*; and in the third clause, *general types*. As usual, we associate $\to$ right to left, and omit parentheses when we can.

The raw syntax of the language is given by

$$\begin{array}{lcl} x & \in & Variable, \\ P & ::= & x \mid \mathtt{n} \mid \mathtt{true} \mid \mathtt{false} \mid (+1)(P) \mid (-1)(P) \mid (=0)(P) \mid \\ & & \mathtt{if}\, P\, \mathtt{then}\, P\, \mathtt{else}\, P \mid \mathtt{rrcons}_a(P) \mid \mathtt{join}_a(P) \mid \mathtt{rtest}_{l,r}(P) \mid \\ & & \lambda x^\tau.P \mid PP \mid \mathtt{Y}P \mid \mathtt{let}\ x = P\ \mathtt{in}\ P \mid \mathtt{pr}_i P \mid \langle P_0, \ldots, P_n \rangle \end{array}$$

where $(+1)(P), (-1)(P)$ and $(=0)(P)$ amount for successor, predecessor and equality for zero respectively; the subscripts of the constructs `rrcons` and `join` are proper rational intervals and those of `rtest` are rational numbers. In the `let` construct, the first term $P$ must be of basic type.

In addition, we allow ourselves the syntactic sugar of writing
$\mathtt{let}\ \langle x_0, \ldots, x_n \rangle = P_1\ \mathtt{in}\ P_2$ where the notation $\langle x_0, \ldots, x_n \rangle$ stands for a variable of the appropriate product type and where free occurrences of $x_i$ in $P_2$ abbreviate $\mathtt{pr}_i \langle x_0, \ldots, x_n \rangle$.

Terms can be associated with types in the familiar style by proof rules and judgements, but in the interest of brevity, we trust the reader to fill in the details.

## 4.2 Denotational Semantics

We define denotational semantics $[\![-]\!]^p$ for $\mathsf{LRT}_p$ subject to a positive real number parameter $p$ in such a way that $[\![M]\!]^p$ is semi-continuous in $p$ and $\bigsqcup_p [\![M]\!]^p$ corresponds to call-by-name interpretation. The idea is to employ $\mathsf{pv}_p$ (see page 34) in the interpretation of the `let` construct to ignore differences

due to "badly" divergent behavior. As $p$ increases, the semantics ignores less. We use $B\,[\![]\!]$ to denote basic types, which includes ground types and product types.

The ground types $\mathtt{bool}, \mathtt{nat}$ and $\mathtt{real}$ are interpreted, first, as the domains of booleans ($\mathbb{T}_\perp$), natural numbers ($\mathbb{N}_\perp$) and compact intervals ($\mathcal{R}_\perp$), respectively. That is,

$$B\,[\![\mathtt{bool}]\!] := \mathbb{T}_\perp, \qquad B\,[\![\mathtt{nat}]\!] := \mathbb{N}_\perp, \qquad B\,[\![\mathtt{real}]\!] := \mathcal{R}_\perp.$$

Finite products are interpreted the usual way: $B\,[\![\gamma \times \beta]\!] := B\,[\![\gamma]\!] \times B\,[\![\beta]\!]$. Basic types are interpreted as Hoare powerdomains of finite products:

$$[\![\beta]\!] := \mathcal{P}^H(B\,[\![\beta]\!]).$$

Function types are interpreted as function spaces in the category of dcpos:

$$[\![\sigma \to \tau]\!] := [\![\sigma]\!] \to [\![\tau]\!].$$

These definitions reflect a call-by-name semantics in which product types are interpreted as consisting of computations of tuples, rather than tuples of computations.

The interpretation of constants is given as follows:

$$[\![\mathtt{true}]\!]^p = \eta(\mathsf{true}), \quad [\![\mathtt{false}]\!]^p = \eta(\mathsf{false}), \quad [\![\mathtt{n}]\!]^p = \eta(\mathsf{n}), \quad [\![(+1)]\!]^p = \widehat{(+1)},$$

$$[\![(-1)]\!]^p = \widehat{(-1)}, \quad [\![(=0)]\!]^p = \widehat{(0=)}, \quad [\![\mathtt{join}_a]\!]^p = \widehat{\mathsf{join}_a},$$

$$[\![\mathtt{rrcons}_a]\!]^p = \widehat{\mathsf{rrcons}_a}, \quad [\![\mathtt{rtest}_{l,r}]\!]^p = \overline{\mathsf{rtest}}_{l,r}, \qquad [\![\mathtt{Y}]\!]^p(F) = \bigsqcup_{n\geq 0} F^n(\perp),$$

$$[\![\mathtt{if}]\!]^p(B,X,Y) = \begin{cases} X, & \text{if } B = \eta(\mathsf{true}), \\ Y, & \text{if } B = \eta(\mathsf{false}), \\ X \uplus Y, & \text{if } B = \eta(\mathsf{true}) \uplus \eta(\mathsf{false}), \\ \perp, & \text{if } B = \perp, \end{cases}$$

with syntactic sugar $[\![\mathtt{if}\ M\ \mathtt{then}\ N\ \mathtt{else}\ P]\!]^p_\rho := [\![\mathtt{if}]\!]^p(\,[\![M]\!]^p_\rho,\ [\![N]\!]^p_\rho,\ [\![P]\!]^p_\rho)$

$$[\![\mathtt{pr}_i]\!]^p = \widehat{\pi}_i,$$

where $\pi_i$ is the usual projection map. Tuples are interpreted by

$$[\![\langle X_1, \ldots, X_n\rangle]\!]^p_\rho := m(\,[\![X_1]\!]^p_\rho, \ldots,\ [\![X_n]\!]^p_\rho)$$

Note that so far, none of these definitions depend on the parameter $p$. The `let` construct enforces what we refer to as *call-by-partial-value*.

$$[\![\mathtt{let}\ x = M\ \mathtt{in}\ N]\!]^p_\rho := [\![N]\!]^p \rho(x/\widehat{\mathsf{pv}_p}(\,[\![M]\!]^p_\rho))$$

Here the symbols $\eta, \widehat{\ }, \overline{\ }$ and $m$ derive from the Hoare powerdomain monad: $\eta$ is the unit, $\widehat{f} := \mathcal{P}^H(f)$, $\overline{f}$ denotes the transpose of $f\colon X \to \mathcal{P}^H(Y)$, $m$ is the natural transformation $\mathcal{P}^H(X_0) \times \ldots \mathcal{P}^H(X_n) \to \mathcal{P}^H(X_0 \times \ldots \times X_n)$. The functions $(+1)$, $(-1)$, $(=0)$ are the standard interpretations in the Scott model of PCF [17], the functions $\mathsf{join}_a, \mathsf{rrcons}_a$ are defined in section 2.2, and the function $\mathsf{rtest}_{l,r}$ is defined by:

$$\mathsf{rtest}_{l,r}(x) = \begin{cases} \eta(\mathsf{true}) \uplus \eta(\mathsf{false}), & \text{if } l < \underline{x} < \overline{x} < r; \\ \eta(\mathsf{true}), & \text{if } \overline{x} \leq r; \\ \eta(\mathsf{false}), & \text{if } \underline{x} \geq l; \\ \perp, & \text{otherwise.} \end{cases}$$

In [13] it was proved that the constants $\widehat{(+1)}$, $\widehat{(-1)}$, $\widehat{(=0)}$, $\widehat{\mathtt{join}_a}$, $\widehat{\mathtt{rrcons}_a}$, $\widehat{\mathtt{pr}_i}$ and $\overline{\mathtt{rtest}}_{l,r}$ denote disciplined functions.

**Lemma 4.1.** *The constants* $\widehat{(+1)}$, $\widehat{(-1)}$, $\widehat{(=0)}$, $\widehat{\mathtt{join}}_a$, $\widehat{\mathtt{rrcons}}_a$ *and* $\widehat{\mathtt{pr}}_i$ *are strongly disciplined functions but not the constant* $\overline{\mathtt{rtest}}_{l,r}$.

*Proof.* The convergence requirement is straightforward. That $\overline{\mathtt{rtest}}_{l,r}$ is not strongly disciplined follows by definition. □

Notice that although $\overline{\mathtt{rtest}}_{l,r}$ is not strongly disciplines, it can be used to define strongly disciplined single valued relations as is shown in [11].

**Lemma 4.2.** *The semantics* $[\![-]\!]^p$ *is semi-continuous in p: for bounded* $A \subseteq \mathbb{R}^+$,

$$\bigsqcup_{p\in A} [\![M]\!]^p = [\![M]\!]^{\sup A}.$$

*Moreover, define* $[\![]\!]^\infty$ *exactly as* $[\![]\!]^p$ *for all cases except*

$$[\![\mathtt{let}\ x = M\ \mathtt{in}\ N]\!]^\infty_\rho := [\![N]\!]^\infty \rho(x/\ [\![M]\!]^\infty_\rho))$$

*Then* $[\![M]\!]^\infty = \bigsqcup_p [\![M]\!]^p$.

*Proof.* The proof is by induction on the structure of $M$. If $M$ is of the form of $(+1)$, $(-1)$, $(=0)$, $\mathtt{join}_a, \mathtt{rrcons}_a, \mathtt{pr}_i$ and $\mathtt{rtest}_{l,r}$ the semantics interpretation does not consider $p$ hence it coinsides with the proof in LRT.

Let $M = [\![\mathtt{let}\ x = P\ \mathtt{in}\ Q]\!]^p_\rho$ hence

$$\begin{aligned}
\bigsqcup_{p\in A} [\![\mathtt{let}\ x = P\ \mathtt{in}\ Q]\!]^p_\rho &= \bigsqcup_{p\in A} [\![Q]\!]^p \rho(x/\widehat{\mathtt{pv}}_p(\ [\![P]\!]^p_\rho)) \\
&= [\![Q]\!]^{\sup A} \rho(x/\widehat{\mathtt{pv}}_p(\ [\![P]\!]^p_\rho)) \text{ by Inductive Hipotesis} \\
&= [\![\mathtt{let}\ x = P\ \mathtt{in}\ Q]\!]^{\sup A}
\end{aligned}$$

□

## 4.3 Operational Semantics

We now develop single-step operational semantics, also parametric in $p$, so that the "$p$-th" operational interpretation is complete for $[\![-]\!]^p$. We do not need an operational semantics corresponding to $[\![-]\!]^\infty$.

**Definition 4.3.** *For each basic type* $\beta$*, we define a subset of the closed terms to be* output terms*, and for each output term* $M$ *we define it's* output $o(M)$ *to be a value in* $B\ [\![\beta]\!]$. *For* `real`*, a term of the form* $\mathtt{join}_a M$ *is an output term, and* $o(\mathtt{join}_a M) := a$. *For* `nat`*, a term of the form* `n` *is an output term, and* $o(\mathtt{n}) = n$. *For* `bool`*, a term of the form* `true` *or* `false` *is an output term, and* $o$ *is defined obviously. For* $\gamma \times \beta$*, a term of the form* $\langle M, N\rangle$ *is an output term provided* $M$ *and* $N$ *are output terms, and* $o(\langle M, N\rangle) = \langle o(M), o(N)\rangle$.

**Lemma 4.4.** *For an output term* $M$ *and* $p > 0$,

$$\eta(o(M)) \sqsubseteq [\![M]\!]^p \sqsubseteq \bigsqcup \{\nu(x) \mid o(M) \sqsubseteq \nu(x)\}.$$

**proof.** [13].

We define $\rightarrow_p$ to be the least relation that includes single-step reduction rules for PCF [17] and is closed under rules for the type `real` and for `let` as follows.

| | | |
|---|---|---|
| (1) | $\mathtt{rrcons}_a(\mathtt{rrcons}_b M) \to_p \mathtt{rrcons}_{ab} M$ | |
| (2) | $\mathtt{join}_a \mathtt{join}_b M \to_p \mathtt{join}_{a \sqcup b} M$ | if $\overline{b} > \underline{a}$ or $\overline{a} > \underline{b}$ |
| (3) | $\mathtt{join}_a \mathtt{join}_b M \to_p \mathtt{rrcons}_a Y(\mathtt{rrcons}_{(-1,0)} \mathtt{join}_I)$ | if $\overline{b} \leq \underline{a}$ |
| (4) | $\mathtt{join}_a \mathtt{join}_b M \to_p \mathtt{rrcons}_a Y(\mathtt{rrcons}_{(0,1)} \mathtt{join}_I)$ | if $\overline{a} \leq \underline{b}$ |
| (5) | $\mathtt{rrcons}_a(\mathtt{join}_b M) \to_p \mathtt{join}_{ab}(\mathtt{rrcons}_a M)$ | |
| (6) | $\mathtt{rtest}_{l,r} \mathtt{join}_a M \to_p \mathtt{true}$ | $\overline{a} < r$ |
| (7) | $\mathtt{rtest}_{l,r} \mathtt{join}_a M \to_p \mathtt{false}$ | $l < \underline{a}$ |
| (8) | $\mathtt{if\ true\ then}\ M\ \mathtt{else}\ M' \to_p M$ | |
| (9) | $\mathtt{if\ false\ then}\ M\ \mathtt{else}\ M' \to_p M'$ | |
| (10) | $\mathtt{pr}_i\langle M_0, \ldots, M_n\rangle \to_p M_i$ | |
| (11) | $\mathtt{let}\ x = M\ \mathtt{in}\ N \to_p [M/x]N$ | $M$ is an output term and $\mu(o(M)) < p$ |
| (12) | $\dfrac{N \to_p N'}{MN \to_p MN'}$ | if $M$ is $\mathtt{join}_a$, $\mathtt{rrcons}_a$, $\mathtt{rtest}_{l,r}$, $\mathtt{if}$, $\mathtt{pr}_i$, $\mathtt{let}$. |

**Definition 4.5.** *We define the operational meaning of a closed term $M$ of basic type $\beta$ in $i$ steps of computation, written $[M]^p_i \in [\![\beta]\!]$.*

*For a closed term of basic type $\beta$, define $[M]_i$ as follows:*

$$[M]^p_i = \bigsqcup \{\eta(o(M')) \mid \exists M' \exists k \leq i, M' \text{ is an output term and } M \stackrel{k}{\to}_p M' \},$$

*where an empty formal join is $\perp$, and $\stackrel{k}{\to}_p$ denotes the $k$-fold composition of the relation $\to_p$.*

*Finally, $[M]^p = \bigsqcup_i [M]^p_i$. which is justified by the obvious fact that $[M]^p_i \sqsubseteq [M]^p_{i+1}$.*

Note that implicit in this definition is the fact that the operational rules are such that $M \stackrel{k}{\to}_p M'$ can only hold for finitely many output terms $M'$. This can be established easily by induction on the operational rules.

The operational interpretation of closed terms is adequate with respect to the denotational semantics.

**Lemma 4.6.** $[\![M]\!]^p = \biguplus \{ [\![N]\!]^p \mid M \to_p N\}$ *(this is a finite union).*

*Proof.* [13].

□

**Lemma 4.7.** *For all closed terms $M$ of ground type, $[M]^p \sqsubseteq [\![M]\!]^p$.*

*Proof.* [13].

□

**Definition 4.8.** *A closed term is said to be p-computable as follows:*

1. *A closed term $M$ of basic type is p-computable whenever $[\![M]\!]^p \sqsubseteq [M]^p$,*
2. *A closed term $M : \sigma \to \tau$ is p-computable whenever $MQ : \tau$ is p-computable for every closed p-computable term $Q$ of type $\sigma$,*

*An open term $M : \sigma$ with free variables $x_1, \ldots, x_n$ of type $\sigma_1, \ldots, \sigma_n$ is p-computable whenever $[N_1/x_1] \cdots [N_n/x_n]M$ is p-computable for every family $N_i : \sigma_i$ of closed p-computable terms.*

**Lemma 4.9.** *Every term of $\mathsf{LRT}_p$ is p-computable.*

*Proof.* [13].

□

**Theorem 4.10.** $[M]^p = [\![M]\!]^p$, *for all closed $\mathsf{LRT}_p$ terms $M$ and all positive reals $p$.*

*Proof.* Lemma 4.7 and Lemma 4.9 ■. □

**Remark 4.11.** *We are interested in those continuos functions $F\colon \mathcal{P}^H(X) \to \mathcal{P}^H(Y)$ for which $u(F(\nu(x)))$ is a singleton. As we observed previously, this property can be verified using the denotational semantics. Although $\perp$ cannot be distinguised from any other element of the Hoare powerdomain, $u$ allows to get those maximal elements of a given element of the Hoare powerdomain, hence, we just have to verified that the image of $u$ is a singleton, which can be done denotationally. By adequacy there is a program which denotes such continuous function. Notice that $\mathsf{LRT}_p$ is more expressive because we can define relations, however, in this paper, we are not concerned about the full expressivity of $\mathsf{LRT}_p$, we only care about the definability of first order computable functions.*

# 5 The *limitation* operator

**Definition 5.1.** *For basic $\mathsf{LRT}_p$ types, we define a set-theoretic interpretation as follows:*

$$R\,[\![\mathtt{nat}]\!] = \mathbb{N}, \quad R\,[\![\mathtt{bool}]\!] = \mathbb{T}, \quad R\,[\![\mathtt{real}]\!] = \mathbb{R}, \quad R\,[\![\gamma \times \beta]\!] = R\,[\![\gamma]\!] \times R\,[\![\beta]\!]$$

Theorem 3.3 establishes that composition in $\mathsf{LRT}_p$ corresponds to $\odot$-composition. That is, if $F$ and $G$ are closed terms of type $\beta_1 \to \beta_2$ and $\beta_2 \to \beta_3$, both are strongly disciplined and $f \overset{\otimes}{\sim} [\![F]\!]$ and $g \overset{\otimes}{\sim} [\![G]\!]$, then $g \odot f \overset{\otimes}{\sim} [\![\lambda x.\mathtt{let}\ y = F(x)\ \mathtt{in}\ G(y)]\!]^p$. In [13] it is extended to the combinators presented in section 2. Here, we only present the limit combinator.

**Definition 5.2.** *Define the following closed term of $\mathsf{LRT}_p$:*

$$\mathtt{Lim}[F](x) \quad := \quad \mathtt{aux\_lim}\ F(x,0)\ \mathtt{id}$$

*where*

$\mathtt{aux\_lim}\ F(x,n)\ G :=$
$\quad \mathtt{let}\ r = G(F(x,n))\ \mathtt{in}$
$\quad\quad \mathtt{if}\ \mathtt{rtest}_{-1,1}(r)$
$\quad\quad \mathtt{then\ if}\ \mathtt{rtest}_{-1,-1/2}(r)$
$\quad\quad\quad \mathtt{then}\ \mathtt{join}_{[-2,-1]}(\mathtt{aux\_lim}\ F(x,n)(\mathtt{rrcons}_{[1,2]} \circ G))$
$\quad\quad\quad \mathtt{else}\ \mathtt{aux\_lim}'F(x,n)G$
$\quad\quad \mathtt{else\ if}\ \mathtt{rtest}_{1/2,1}(r)$
$\quad\quad\quad \mathtt{then}\ \mathtt{aux\_lim}'F(x,n)G$
$\quad\quad\quad \mathtt{else}\ \mathtt{join}_{[1,2]}(\mathtt{aux\_lim}\ F(x,n)(\mathtt{rrcons}_{[-2,-1]} \circ G))$

$\mathtt{aux\_lim}'\ F(x,n)\ G :=$
$\quad \mathtt{let}\ r = G(F(x,n))\ \mathtt{in}$
$\quad\quad \mathtt{if}\ \mathtt{rtest}_{-5/16,5/16}(r)$
$\quad\quad \mathtt{then\ if}\ \mathtt{rtest}_{-5/16,-4/16}(r)$
$\quad\quad\quad \mathtt{then}\ \mathtt{cons}_L(\mathtt{aux\_lim}\ F(x,(+1)(n))(\mathtt{tail}_L \circ G))$
$\quad\quad\quad \mathtt{else}\ \mathtt{cons}_C(\mathtt{aux\_lim}\ F(x,(+1)(n))(\mathtt{tail}_C \circ G))$
$\quad\quad \mathtt{else\ if}\ \mathtt{rtest}_{4/16,5/16}(r)$
$\quad\quad\quad \mathtt{then}\ \mathtt{cons}_C(\mathtt{aux\_lim}\ F(x,(+1)(n))(\mathtt{tail}_C \circ G))$
$\quad\quad\quad \mathtt{else}\ \mathtt{cons}_R(\mathtt{aux\_lim}\ F(x,(+1)(n))(\mathtt{tail}_R \circ G))$

$$\mathtt{cons}_a := \mathsf{join}_A \circ \mathtt{rrcons}_a \mathtt{tail}_a := \mathsf{join}_A \circ \mathtt{rrcons}_{a^{-1}}$$
$$A := [-1,1] L := [-1/2,0] C := [-1/4,1/4] R := [0,1/2]$$

*In this definition we understand* Lim *to be a second-order term, where $F$ is an argument. We set them apart for readability using square brackets. The* aux_lim *definition, translate the range of the function $F(x,n)$ to the interval [-1,1]. As $F(x,n)$ is assume to be a Cauchy Sequence, once the value of $F(x,n)$ is in the interval $[-1,1]$, for any $m > n$ it holds that $F(x,m) \in [-1,1]$. The definition of* aux_lim′ *computes the limit of a Cauchy sequence in the interval $[-1,1]$. It evaluates the function $G(F(x,n))$ until its output is guarantee to be contained in either $L$, $C$ or $R$. Then it outputs an interval and continue evaluating $F(x,n+1)$. Plume [18] and Farjudian [8] give a detailed explanation of the limit algorithm used here.*

**Theorem 5.3.** *For any $p < 1/4$, in the semantics $[\![-]\!]^p$, the term* `Lim` *preserves disciplined. Moreover, if $f \overset{\otimes}{\sim} F$ for single-valued $f$ and these "type check" in the obvious way, then $\lim f \overset{\otimes}{\sim} [\![\texttt{Lim}]\!]^p F$. In other words if the input of* `Lim` *is a strongly disciplined $F$ then* `Lim`$[F]$ *convergences to a single-valued output.*

*Proof.* The assumption $p < 1/4$ is needed to ensure that the limit of the strong Cauchy sequence in which $r_i$ appears as the $i$-th term is bounded within a distance of $2^{-(i+2)}$ from $r_i$. In fact, this is the only point at when the assumption that $p$ is small is required. As $F$ is strongly disciplined it consist of a single strongly Cauchy sequence which converges to a single point, hence the above program computes different paths, but all of them converge to the same maximal value. □

# 6 Conclusions

By allowing a reasonable definition of strongly convergence for relations, we get a characterization of single-valued relations (e.g. functions). This characterization differs from the one presented in [11] in that the former is denotational while the further is operational. We have already noticed in [11] that the proofs of partial correctness and strong convergence agreed, but they had to be presented. In this paper, we proved that for first order computable functions a denotational proof is sufficient to show if a program written in $\mathsf{LRT}_p$ converges to a single valued or diverges.

# 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, 1994.

[2] V. Brattka. Recursive characterization of computable real-valued functions and relations. *Theoretical Computer Science*, 162:45–77, 1996.

[3] P. Di Gianantonio. *A functional approach to computability on real numbers.* PhD thesis, University of Pisa, 1993. Thecnical Report TD 6/93.

[4] M. H. Escardó. PCF extended with real numbers. *Theoretical Computer Science*, 162(1):79–115, August 1996.

[5] M. H. Escardó. *PCF Extended with Real Numbers: a domain-theoretic approach to higher-order exact real number computation.* PhD thesis, University of London, 1997.

[6] M. H. Escardó, M. Hofmann, and T. Streicher. On the non-sequential nature of the interval-domain model of real-number computation. *Mathematical Structures in Computer Science*, 14(6):803–814, 2004.

[7] A. Farjudian. *Sequentiality in Real Number Computation.* PhD thesis, University of Birmingham, 2004.

[8] A. Farjudian. Shrad: A language for sequential real number computation. *Theoretical Computer Science*, 41(1):49–105, 2007.

[9] J. R. Marcial-Romero. *Semantics of a sequential language for exact real-number computation.* PhD thesis, University of Birmingham, December 2004.

[10] J. R. Marcial-Romero and M. H. Escardó. Semantics of a sequential language for exact real-number computation. In Harald Ganzinger, editor, *Proceedings of the Nineteenth Annual IEEE Symp. on Logic in Computer Science, LICS 2004*, pages 426–435. IEEE Computer Society Press, July 2004.

[11] J. R. Marcial-Romero and M. H. Escardó. Semantics of a sequential language for exact real-number computation. *Theoretical Computer Science*, 379(1-2):120–141, 2007.

[12] J. R. Marcial-Romero and J. A. Hernández. Functional first order definability of $LRT_p$. In Mauricio Osorio, editor, *Proceedings of the Fifth Latinoamerican workshop on Non-monotonic reasoning, LANMR 2009*, pages 46–61. CEUR, November 2009.

[13] J. R. Marcial-Romero and A. Moshier. Sequential real number computation and recursive relations. *Mathematical Logic Quarterly*, 54(5):492–507, 2008.

[14] K. Martin. Domain theoretic models of topological spaces. In Abbas Edalat, Achim Jung, Klaus Keimel, and Marta Kwiatkowska, editors, *Proceedings of Comprox III, ENTCS*, volume 13, pages 173–181. Elsevier, 1998.

[15] K. Martin. The measurement process in domain theory. In *Automata, Languages and Programming*, pages 116–126, 2000.

[16] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.

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

[18] D. Plume. A calculator for exact real number computation. Master's thesis, Department of Computer Science and Artificial Intelligence, 1998.

[19] P. J. Potts, A. Edalat, and M.H Escardó. Semantics of exact real arithmetic. In *In Proceedings of the Twelveth Annual IEEE Symposium on Logic In Computer Science*. IEEE Computer Society Press, 1997.

[20] D. S. Scott. Lattice theory, data type and semantics. In *Formal semantics of programming languages*, pages 66–106. Englewood Cliffs, Prentice-Hall, 1972.