

Total Correctness for Sequential Real Programs¹

Author: Thomas Anberrée, University of Birmingham

Real PCF is a higher type language for exact real number computation introduced by Martín Escardó [1]. It is universal, in the sense that it can define all computable functions over the reals, at all types. But, in place of the usual, sequential *if* operator, it includes a *parallel conditional* satisfying

$$\text{pif true then } x \text{ else } y = x, \quad \text{pif false then } x \text{ else } y = y, \quad \text{pif } \perp \text{ then } x \text{ else } x = x,$$

whose use is unavoidable, at least in a deterministic setting with a domain-based model (cf. [2]).

In our approach, a real-number computation is an infinite sequence of shrinking rational intervals. This infinite character of real numbers makes the tests $x = y$ and $x \leq y$ undecidable. In particular, such tests cannot be used to control the execution flow of real-number programs. An alternative solution to the use of a parallel operator is the use of a non-deterministic test: for any two numbers $p < q$ and any number x , at least one of the relations $p < x$ or $x < q$ can be determined to hold (Boehm and Cartwright [3]). One can use a construct $\text{rtest}_{p,q}$, for $p < q$ rational, such that, for any real number x ,

1. $\text{rtest}_{p,q}(x)$ evaluates to true or to false,
2. $\text{rtest}_{p,q}(x)$ may evaluate to true iff $x < q$, and
3. $\text{rtest}_{p,q}(x)$ may evaluate to false iff $p < x$.

It is important here that evaluation never diverges.

Since a program can in general produce different results in different runs, Escardó and Marcial-Romero took the view that programs of real-number type denote sets of real numbers rather than real numbers, and the question arose as to which power domains would be suitable for modeling the behaviour of rtest .

Contributions. Building on Escardó and Marcial-Romero’s work [4], we show that although the Smyth power domain cannot be used to faithfully model the $\text{rtest}_{p,q}$ operator, as showed by these authors, it can be used to give an approximation which is sufficiently precise for applications of interest, and bears the further quality over other choices that *total correctness* of programs is expressed in the model. If one writes $\llbracket M \rrbracket$ and $[M]$ for the denotational and operational meaning of a program M , respectively, we show that, in general, one has $\llbracket M \rrbracket \sqsubseteq [M]$, but not necessarily $\llbracket M \rrbracket = [M]$. However, if $\llbracket M \rrbracket$ is total and M is of ground type, then $\llbracket M \rrbracket$ is maximal, and hence $\llbracket M \rrbracket = [M]$. Here is what the denotation of the program construct $\text{rtest}_{p,q}(x)$ is taken to be:

1. $\llbracket \text{rtest}_{p,q} \rrbracket(x) = \{\text{true}, \text{false}\}$ if $p \leq x \leq q$,
2. $\llbracket \text{rtest}_{p,q} \rrbracket(x) = \{\text{true}\}$ if $x < p$, and
3. $\llbracket \text{rtest}_{p,q} \rrbracket(x) = \{\text{false}\}$ if $q < x$.

This approximate semantics doesn’t agree with the operational definition of $\text{rtest}_{p,q}$, because $\llbracket \text{rtest}_{p,q} \rrbracket(p) = \{\text{true}\} \sqsupseteq \{\text{true}, \text{false}\} = \llbracket \text{rtest}_{p,q} \rrbracket(p)$ and, similarly, $\llbracket \text{rtest}_{p,q} \rrbracket(q) = \{\text{false}\} \sqsupseteq \{\text{true}, \text{false}\} = \llbracket \text{rtest}_{p,q} \rrbracket(q)$, where the ordering \sqsubseteq is that of the Smyth power domain. But this is necessary to render the denotational interpretation of $\text{rtest}_{p,q}$ continuous.

To illustrate the usefulness of the approximate semantics in practice, we show that, for many programs of interest, this denotation allows to easily show that they are correct without resorting to operational methods.

References

- [1] Escardó, M.: PCF extended with real numbers. *Theoretical Computer Science* **162**(1) (1996) 79–115
- [2] Escardó, M., Hofmann, M., Streicher, T.: On the non-sequential nature of the interval-domain model of real-number computation. *Mathematical Structures in Computer Science* **14**(6) (2004) 803–814
- [3] Boehm, H.J., Cartwright, R.: Exact real arithmetic formulating real numbers as functions. *Research topics in functional programming* (1990) 43–64

¹A longer version is available at <http://www.cs.bham.ac.uk/~txa/domains8.pdf> but it doesn’t include relevant examples of programs yet, apart from the absolute value.

- [4] Marcial-Romero, J.R., Escardo, M.: Semantics of a sequential language for exact real-number computation. *Theoretical Computer Science* (2007)