On the non-sequential nature of the interval-domain model of real-number computation

Martín Escardó, Martin Hofmann, Thomas Streicher

April 1999, version of 18th October 2002

Abstract

We show that real-number computations in the interval-domain environment are "inherently parallel" in a precise mathematical sense. We do this by reducing computations of the weak parallel-or operation on the Sierpinski domain to computations of the addition operation on the interval domain.

1 Introduction

Suppose we extend a functional programming language with an abstract data type *real* meant to represent infinite-precision real numbers [3]. What is a sensible semantics for such a type or how does one explain to users what their programs involving *real* actually mean? It is probably not sensible to interpret *real* as just the set of real numbers as this would not assign meaning to nonterminating computations ("$\perp$") nor to partial computations, say one that outputs the information that a number lies between 0 and 0.5 but does not terminate upon requests for higher accuracy.

Scott proposed the *interval domain* as a possible answer to this question [14] (other possibilities are discussed in Section 4 below). For simplicity and without essential loss of generality, we consider here only its restriction to the unit interval $[0, 1]$. The elements of the interval domain $\mathcal{I}$ are the closed intervals $x = [\underline{x}, \overline{x}]$ where $0 \leq \underline{x} \leq \overline{x} \leq 1$. These are ordered by reverse inclusion, i.e., $x \sqsubseteq y$ if and only if $\underline{x} \leq \underline{y}$ and $\overline{y} \leq \overline{x}$. Then $(\mathcal{I}, \sqsubseteq)$ becomes a directed complete partial order and thus supports general recursion via fixed points. The idea of the relation $x \sqsubseteq y$ is that y gives at least as much information as x about an unknown real number in the unit interval, and possibly more. If y is a singleton, that is, a maximal element in the ordering, then it gives complete or *total* information. The ordered set $\mathcal{I}$ is a typical example of a *continuous Scott domain*. For an exposition of the general theory, the reader is referred to Abramsky and Jung [1], Amadio and Curien [2] or to the forthcoming treatise by Gierz et al. [8]. For a topological view, see the survey paper by Mislove [10], and, for recent applications, the reader is referred to the survey paper by Edalat [4].

The first-named author has developed a lazy functional programming language called Real PCF based on the interval domain. One of the central results concerning Real PCF is that its interpretation in terms of the interval domain is *computationally adequate* with respect to a lazy operational semantics [5]. This operational semantics associates with expressions of type *real* a shrinking sequence of rational intervals. Computational adequacy asserts that the intersection of these intervals agrees with the interval-domain based denotational semantics of such an expression.

In contrast to most ordinary functional languages, Real PCF contains a construct whose operational semantics requires parallel evaluation (or "dovetailing" in the terminology of recursion theory [13]). More precisely, one has a construct $\mathsf{pif} : \mathit{bool} \times \mathit{real} \times \mathit{real} \to \mathit{real}$ defined by $\mathsf{pif}(\mathsf{true}, x, y) = x$ and $\mathsf{pif}(\mathsf{false}, x, y) = y$ and $\mathsf{pif}(\bot, x, y) = x \sqcap y$ where $x \sqcap y$ is the least interval containing both x and y. In particular, $\mathsf{pif}(\bot, x, x) = x$. It is clear that when evaluating a term $\mathsf{pif}(b, x, y)$ we must evaluate b, x, y in parallel and output partial results while the computation of b has not yet finished. This construct is used in order to ensure that Real PCF is Turing-universal, that is, that all real functions and functionals that are computable in the sense of recursion theory are programmable in the language [6].

It is natural to ask whether the presence of this parallel construct is an artifact of Real PCF or whether it is required for intrinsic reasons. The result of this paper confirms that the latter is the case. More precisely, we show that, under very mild and reasonable assumptions, any language that is computationally adequate with respect to the interval domain must necessarily allow the definition of the "weak parallel-or" construct wpor defined on the Sierpinski domain $\{\bot, \top\}$ (interpreting the unit type or arising as a retract of any other type), given by $\mathrm{wpor}(x, y) = \top$ if and only if $x = \top$ or $y = \top$. Clearly, to evaluate $\mathrm{wpor}(x, y)$ one must evaluate x and y in parallel in order to be able to terminate as soon as either x or y has terminated.

The presence of such a parallel construct is a disadvantage from a practical point of view because it renders the evaluation process inefficient due to the exponential spawning of processes in recursive definitions involving the construct. However, the ultimate practical consequences of our result, which, we emphasize, is specific to the interval-domain model, are not clear cut. On the one hand, it may be that there exists an alternative mathematical model, yet to be found, which doesn't exhibit this parallel behaviour. In this case, the interpretation of our result would be that the interval-domain model is at fault, and hence one should look for another, sequential model. On the other hand, it may be that this is not a fault of the interval-domain model, but rather an intrinsic property of the real numbers. This is supported, for instance, by the work of Luckhardt [9]; see also the discussion about strongly stable models in Section 4 below. In this case, one should look for a proof that any reasonable model exhibits the same behaviour. We have at this point little theoretical or empirical evidence for any of these two conclusions and therefore leave the interpretation to the reader and to future work.

We now proceed to a more technical introduction to our result, in which we give a simple proof of an important particular case.

Let's take intervals with rational end-points as the "observable tokens" in infinite-precision real-number computations (of course, one could instead work with dyadic numbers or any other convenient recursively enumerable dense set). A computation can be taken as an r.e. ascending sequence r_i of tokens. This computes the interval $\bigsqcup_i r_i = \bigcap_i r_i = [\sup_i \underline{r_i}, \inf_i \overline{r_i}]$. If the end-points of this are equal, then this is a computation of a real number; otherwise, it is a *partial computation* of a real number, or, as we prefer to see it, a computation of a *partial(ly defined) real number*.

Suppose that we wish to compute the midpoint operation $m : [0,1]^2 \to [0,1]$ defined by

$$m(x,y) = (x+y)/2.$$

Then we need a process that, given computations r_i of x and s_i of y, produces a computation t_i of $m(x,y)$. In this case, it is obvious what to do: just define $t_i = m(r_i, s_i)$, where the midpoint of two intervals is taken pointwise. This process not only computes the midpoint operation $m : [0,1]^2 \to [0,1]$, but also its pointwise extension to partial numbers $\hat{m} : \mathcal{I}^2 \to \mathcal{I}$ defined by

$$\hat{m}(x,y) = [m(\underline{x}, \underline{y}), m(\overline{x}, \overline{y})].$$

But this is just one among many possible ways of performing the computation. Another, perhaps artificial, way is this: At stage i of the computation, read two tokens r_i and s_i from the input. If they have an upper bound in the ordering (that is, overlap as intervals) output their meet $t_i = r_i \sqcap s_i$ (that is, their union as intervals). Otherwise, as in the previous example, output the pointwise midpoint $t_i = m(r_i, s_i)$. For total(ly defined) numbers, this behaves as the midpoint operation. More generally, this gives rise to the extension $\tilde{m} : \mathcal{I}^2 \to \mathcal{I}$ defined by

$$\tilde{m}(x,y) = \begin{cases} x \sqcap y & \text{if } x \text{ and } y \text{ have an upper bound,} \\ \hat{m}(x,y) & \text{otherwise.} \end{cases}$$

The above two extensions of the midpoint operation are easily seen to be Scott continuous. Our last example of a computation illustrates a different phenomenon: At stage i, read two tokens r_i and s_i from the input. If the sum of their lengths is strictly bigger than 1, output $t_i = [0,1]$, the bottom element of $\mathcal{I}$. Otherwise, as in the first example, output the pointwise midpoint $t_i = m(r_i, s_i)$. For total numbers, this again computes the midpoint. However, this computation cannot be modelled by a third extension of the midpoint operation to the interval domain. In fact, consider the case in which the inputs are $x = y = [0, 1/2]$. One possible pair of computations of x and y are just the constant sequences $r_i = s_i = [0, 1/2]$, for which we get the constant sequence $t_i = [0, 1/2]$ as output. But another possible pair of computations is $r_i = s_i = [0, 1/2 + 1/(i+1)]$, for which we get the constant sequence $t_i = [0,1]$ as output. Thus, the output of this algorithm depends not only on the input but also on the way the input is presented.

If one is interested only in total numbers, there is nothing wrong with the last example. But if one works with the interval domain to attach mathematical meaning to partial computations and uses functions to account for the input-output behaviour of algorithms, such computations are ruled out. Notice that the phenomenon illustrated by the last example is not particular to the interval domain and can be reproduced in more traditional Scott domains, where the tokens are elements of a countable basis of the domain.

In practice, one doesn't work with computations as above. A paradigmatic example is the functional programming language PCF [11]. The terms of the language can be interpreted either as mathematical expressions (via the denotational semantics) or as algorithms (via the operational semantics). The operational semantics consists of a set of mathematically valid equations and algorithmic rules for applying them. For example, out of the term "3+7" the calculation produces the term "10". The case of real-number computation is a bit subtler, as the "final value" cannot be computed in finitely many steps and exhibited at once, but is based on the same principle [5].

In both cases, it is of interest to know whether the operational semantics is "sequential". To explain this, we refer to the process of replacing a subterm of a term by an equal term, according to the equations of the operational semantics, as *reduction*. Then the language is said to be sequential if any unevaluated term M has a subterm that has to be reduced first in order to evaluate M. In the case of PCF, this is made precise by the Activity Lemma [11].

The aim of this paper is to show that, under reasonable assumptions, a language with a data type for real numbers which is mathematically interpreted as the interval domain and whose constructs are interpreted as Scott continuous functions cannot have a sequential operational semantics. Of course, we assume that computational adequacy holds: the mathematical and operational meanings of the language coincide for terms of base types.

A simple instance of the failure of sequentiality is incorporated in the first extension of the midpoint operation. In order to see this, consider the Sierpinski domain $\Sigma = \{\bot, \top\}$ with $\bot \sqsubseteq \top$, where, computationally, $\bot$ denotes non-termination and $\top$ denotes termination (without any further output), and the two computable functions $e : \Sigma \to \mathcal{I}$ and $p : \mathcal{I} \to \Sigma$ defined by

$$e(b) = \begin{cases} 1 & \text{if } b = \top, \\ [0,1] & \text{otherwise,} \end{cases} \qquad\qquad p(x) = \begin{cases} \top & \text{if } \underline{x} > 1/3, \\ \bot & \text{otherwise.} \end{cases}$$

Here 1 stands for the singleton interval $[1, 1]$, and, following the philosophy discussed above, we generally identify, conceptually and notationally, singleton intervals with real numbers. These functions are intuitively sequential: For the first, we just wait for the input to terminate; if it does, then we output any computation of 1; otherwise we just wait for ever in vain, so that our output is a non-terminating computation, which corresponds to complete absence of information (that is, the bottom interval $[0, 1]$). For the second, we just read tokens from the input until some satisfies the required condition, in which case we terminate; if this condition is never met, then our output is a non-terminating computation, as required.

Now consider the function $f : \Sigma^2 \to \Sigma$ defined by

$$f(b, c) = p(m(e(b), e(c))).$$

If m were sequential, then so would be f. But f is a prototypical example of a nonsequentially computable function, the so-called weak parallel or discussed above:

$$\begin{aligned} f(\bot,\bot) &= p(m([0,1],[0,1])) = p([0,1]) = \bot, \\ f(\bot,\top) &= p(m([0,1],1)) = p([1/2,1]) = \top, \\ f(\top,\bot) &= p(m(1,[0,1])) = p([1/2,1]) = \top, \\ f(\top,\top) &= p(m(1,1)) = p(1) = \top. \end{aligned}$$

That is, the computation of f terminates if and only if the computation of at least one of its arguments terminates, so that it is not possible to choose one argument to evaluate first in order to correctly compute the output — one has to alternate between watching the first and second arguments until one becomes $\top$.

One might suspect that the problem here is that one is working with the *greatest* continuous extension of the midpoint operation. With a more complicated argument, we show in Section 3 below that, no matter which continuous extension is chosen, weak parallel or is sequentially definable from it.

The midpoint operation and the unit interval domain $\mathcal{I}$ have been chosen for technical and expositional reasons. One can equally well work with addition and the full interval domain $\mathcal{R}$. But the corresponding result for this follows by reduction to the previous case. We just observe that (1) the pointwise extension of the operation $x \mapsto x/2$ to the full interval domain is manifestly sequential, (2) the function $r : \mathbb{R} \to [0,1]$ given by

$$r(x) = \max(0, \min(x, 1)),$$

extends to a computable function $r : \mathcal{R} \to \mathcal{I}$ which is manifestly sequential, and (3) any continuous extension of addition to the full interval domain gives rise to a continuous extension of the midpoint operation on the unit interval domain by composition with the functions defined in (1) and (2).

2 Inherently parallel functions

For the purposes of the present paper, it is convenient to refer to the sublanguage of Real PCF consisting of terms in which the parallel conditional doesn't occur as Weak Real PCF. The operational semantics of the sublanguage is clearly sequential, and the "manifestly sequential" functions considered in the introduction and in the development that follows are programmable in the sublanguage. Notice that all of them are *unary* functions. But, we emphasize, for the purposes of the arguments that follow, no previous knowledge of Real PCF is needed

After briefly summarizing some of the notation and terminology established in the introduction, we precisely formulate the theorem to be proved.

Let $\mathcal{I}$ be the continuous Scott domain of non-empty closed subintervals of $[0,1]$ ordered by reverse inclusion denoted by $\sqsubseteq$ and referred to as the information order. Hence $\bot = [0,1]$

is the least element of $\mathcal{I}$. Any $x \in \mathcal{I}$ is of the form $[\underline{x}, \overline{x}]$ with $0 \leq \underline{x} \leq \overline{x} \leq 1$. We write $|x|$ for the size of an interval x, given as $\overline{x} - \underline{x}$. The elements x with $\underline{x} < \overline{x}$ are called partial real numbers; those with $\underline{x} = \overline{x}$ are called total real numbers; we identify the total elements of $\mathcal{I}$ with points of $[0, 1]$. A rational interval is one with both endpoints rational numbers.

Any two members of $\mathcal{I}$ have a meet, or greatest lower bound, in the information order, namely the smallest closed interval containing their set-theoretical union. Two elements of $\mathcal{I}$ are *compatible* if they have an upper bound in the information order (that is, overlap as intervals). In this case, their set-theoretical union is already an interval and hence gives their meet.

We denote by Σ the Sierpinki domain $\{\bot, \top\}$ with the ordering $\bot \sqsubseteq \top$. The "weak parallel-or" function $\mathrm{wpor} : \Sigma^2 \to \Sigma$ is given by $\mathrm{wpor}(x, y) = \top$ if and only if $x = \top$ or $y = \top$. Our goal is to define wpor from any continuous extension of the midpoint operation and certain basic continuous functions which are regarded as undoubtedly sequential and we now describe. As explained above, all of them are Weak Real PCF definable.

First, we need test functions from $\mathcal{I}$ to Σ. We can leave these largely unspecified and simply assume that for any $a, b \in \mathcal{I}$ with

$$a \sqsubseteq b \text{ and } a \neq b$$

there is a test function

$$\mathrm{test}_{a,b} : \mathcal{I} \to \Sigma$$

such that

$$\mathrm{test}_{a,b}(a) = \bot, \qquad \mathrm{test}_{a,b}(b) = \top.$$

For any rational $p, q \in \mathcal{I}$ with

$$p \sqsubseteq q$$

we have the path function

$$\mathrm{path}_{p,q} : \Sigma \to \mathcal{I}$$

with

$$\mathrm{path}_{p,q}(\bot) = p, \qquad \mathrm{path}_{p,q}(\top) = q.$$

For a rational interval q we let

$$\mathrm{cons}_q : \mathcal{I} \to \mathcal{I}$$

be the unique increasing affine transformation sending $\bot = [0, 1]$ to q. In fact, we have

$$\mathrm{cons}_q(x) = ax + b = [a\underline{x} + b, a\overline{x} + b]$$

when

$$q = [b, a + b].$$

Definition 2.1 Let $f : \mathcal{I}^m \to \mathcal{I}$ be a function. A function $h : \mathcal{I}^n \to \mathcal{I}$ is called f-definable if it can be obtained from f, the test functions $\mathrm{test}_{a,b}$, the path functions $\mathrm{path}_{p,q}$, the affine transformations cons_q, and the constants $\bot, \top \in \Sigma$, using projections and substitution.

In other words, the f-definable functions are the clone generated by f and the basic functions. We consider the function wpor as indisputably parallel and use definability of wpor as definition of inherent parallelism.

Definition 2.2 A function f is *inherently parallel* if the function wpor is f-definable.

A function $f : \mathcal{I}^2 \to \mathcal{I}$ is an extension of the midpoint operation if for total elements $x, y \in \mathcal{I}$ the value $f(x, y)$ is also total and moreover $f(x, y) = (x + y)/2$. We are interested in (Scott) *continuous* extensions of the midpoint operation. Two examples of extensions have been considered in the introduction.

The next section is devoted to showing that any continuous extension of the midpoint operation is inherently parallel.

3 Mediation is inherently parallel

We begin our study of inherent parallelism with a design for a definition of wpor which works in particular for the greatest continuous extension of the midpoint operation.

Lemma 3.1 *Any continuous function $f : \mathcal{I}^2 \to \mathcal{I}$ for which there are $a, a', b, b' \in \mathcal{I}$ with $a \sqsubseteq a'$, $b \sqsubseteq b'$ and $f(a, b) \neq f(a', b) \sqcap f(a, b')$ is inherently parallel.*

Proof. By monotonicity, we have $f(a, b) \sqsubseteq f(a', b) \sqcap f(a, b')$, and, by continuity, we can assume w.l.o.g. that a, a', b, b' are rational intervals. The result then follows from the fact that

$$\mathrm{wpor}(x, y) = \mathrm{test}_{f(a,b), f(a',b) \sqcap f(a,b')}(f(\mathrm{path}_{a,a'}(x), \mathrm{path}_{b,b'}(y))).$$

□

Corollary 3.1 *The greatest continuous extension of the midpoint operation is inherently parallel.*

Proof. Apply Lemma 3.1 with $a = b = \bot$ and $a' = b' = 1$. □

The next result shows that unless a function is inherently parallel one can construct from it a unary non-stable function which moreover exhibits its non-stability at an a priori chosen pair of intervals.

Lemma 3.2 *Let $f : \mathcal{I}^2 \to \mathcal{I}$ be a continuous extension of the midpoint operation. Then f is inherently parallel or for any two compatible intervals $a, b \in \mathcal{I}$ such that $a \sqcap b \neq a$ and $a \sqcap b \neq b$ there is an f-definable function $h : \mathcal{I} \to \mathcal{I}$ with*

$$h(a \sqcap b) \neq h(a) \sqcap h(b).$$

Proof. Assume that f is not inherently parallel. We use Lemma 3.1 to produce the desired function h. By the assumption on a, b, there is a number $\lambda < 1$ such that $\max(|a|, |b|) < \lambda |a \sqcap b|$. Let u be the lower endpoint of $a \sqcap b$ which w.l.o.g. we assume to be the lower endpoint of a as well. Let v be any rational point in the interval $a \sqcap b$ but not in the interval a (and hence distinct from u). Lemma 3.1 shows that

$$f(a \sqcap b, a \sqcap b) = f(a \sqcap b, v) \sqcap f(v, a \sqcap b). \tag{1}$$

Since f is an extension of the midpoint operation, $u \in f(a \sqcap b, a \sqcap b)$, and hence either $u \in f(a \sqcap b, v)$ or $u \in f(v, a \sqcap b)$. W.l.o.g., we assume the former:

$$u \in f(a \sqcap b, v). \tag{2}$$

We then claim that there exists a rational interval d such that

$$f(\mathrm{cons}_d(a \sqcap b), v) \neq f(\mathrm{cons}_d(a), v) \sqcap f(\mathrm{cons}_d(b), v) \tag{3}$$

so we would be done with

$$h(x) = f(\mathrm{cons}_d(x), v). \tag{4}$$

For the sake of contradiction, assume that

$$f(\mathrm{cons}_d(a \sqcap b), v) = f(\mathrm{cons}_d(a), v) \sqcap f(\mathrm{cons}_d(b), v) \tag{5}$$

for each rational interval d. We define inductively a sequence $(d_i)_{i \in \mathbb{N}}$ of rational intervals such that

$$u \in f(\mathrm{cons}_{d_i}(a \sqcap b), v). \tag{6}$$

and $\mathrm{cons}_{d_i}(a \sqcap b)$ converges to a total real number. This will give the desired contradiction as $u \neq (z+v)/2 = f(z, v)$ for any total $z \in a \sqcap b$. We put $d_0 = \perp$; then $f(\mathrm{cons}_{d_0}(a \sqcap b), v) = f(a \sqcap b, v)$ and $u \in f(\mathrm{cons}_{d_0}(a \sqcap b), v)$ by assumption (2). If d_i has already been constructed then by (6) we know that $u \in f(\mathrm{cons}_{d_i}(a \sqcap b), v)$ and assumption (5) shows that u belongs to either $f(\mathrm{cons}_{d_i}(a), v)$ or $f(\mathrm{cons}_{d_i}(b), v)$. W.l.o.g. we may assume the former and let d_{i+1} be a rational interval satisfying

1. $\mathrm{cons}_{d_{i+1}}(a \sqcap b) \sqsubseteq \mathrm{cons}_{d_i}(a)$,

2. $|\operatorname{cons}_{d_{i+1}}(a \sqcap b)| \leq \lambda |\operatorname{cons}_{d_i}(a \sqcap b)|$.

(Notice that if a, b are rational intervals, we can simply choose d_i as the unique rational interval making the first part of the specification an equality. Otherwise we slightly enlarge it so that the second part still holds.) By (2), using the fact that $\lambda < 1$, the sequence of intervals $\operatorname{cons}_{d_i}(a \sqcap b)$ shrinks and converges to a total number $w \in a \sqcap b$. Now $f(w, v) = (w + v)/2$ is a total number different from u. Therefore, by continuity of f, we must have

$$u \notin \bigsqcup_i f(\operatorname{cons}_{d_i}(a \sqcap b), v)$$

and hence $u \notin f(\operatorname{cons}_{d_i}(a \sqcap b), v)$ for some i, contradicting (6). □

Remark 3.1 *If f is inherently parallel, it is still the case that a unary non-stable function can be defined from f. In fact, consider intervals $c \sqsubseteq c'$ with $c \neq c'$. Since* wpor *is f-definable by assumption, we can take*

$$h(x) = \operatorname{path}_{c,c'}(\operatorname{wpor}(\operatorname{test}_{a \sqcap b, a}(x), \operatorname{test}_{a \sqcap b, b}(x))).$$

But this is not needed for our purposes.

We now establish a relationship between inherent parallelism and a naive generalization of sequentiality in the sense of Vuillemin [12]. Assume that in a computation of $z = f(x, y)$, approximations $u \sqsubseteq x$ and $v \sqsubseteq y$ have been read from the input and that $w = f(u, v)$ is the partial output so far. If f were "intuitively sequential" then it would have to be that in this situation either f awaits progress in its left argument, i.e., $f(u, v') = w$ for all $v' \sqsupseteq v$ or the other way round, i.e., $f(u', v) = w$ for all $u' \sqsupseteq u$. The following asserts that any f that is not inherently parallel exhibits this property.

Lemma 3.3 *If a continuous extension $f : \mathcal{I}^2 \to \mathcal{I}$ of the midpoint operation is* not *inherently parallel, then for any two intervals u, v one of the following cases holds:*

1. *$f(u, v) = f(u, v')$ whenever $v \sqsubseteq v'$, or*

2. *$f(u, v) = f(u', v)$ whenever $u \sqsubseteq u'$.*

Proof. Suppose, for the sake of contradiction, that we are given intervals $u, u', v, v' \in \mathcal{I}$ with $u \sqsubseteq u'$ and $v \sqsubseteq v'$ such that

1. $f(u, v) \neq f(u', v)$, and

2. $f(u, v) \neq f(u, v')$.

Let $a = f(u', v)$ and $b = f(u, v')$. By Lemma 3.1 we have $f(u, v) = a \sqcap b$. So, in this case $a \sqcap b \neq a$ and $a \sqcap b \neq b$. Hence by Lemma 3.2 there is an f-definable function $h : \mathcal{I} \to \mathcal{I}$ such that

$$h(a \sqcap b) \neq h(a) \sqcap h(b).$$

It follows that

$$\mathrm{wpor}(x, y) = \mathrm{test}_{h(a \sqcap b), h(a) \sqcap h(b)}(h(f(\mathrm{path}_{u,u'}(x), \mathrm{path}_{v,v'}(y)))),$$

as required. $\square$

We have now laid the groundwork to prove our main result.

Theorem 3.1 *Any continuous extension of the midpoint operation is inherently parallel.*

Proof. For a given continuous extension f of the midpoint operation, define subsets L, R of $\mathcal{I}^2$ by

$$\begin{array}{lcl} (a, b) \in L & \Leftrightarrow & \forall b' \sqsupseteq b.\ f(a, b) = f(a, b'), \\ (a, b) \in R & \Leftrightarrow & \forall a' \sqsupseteq a.\ f(a, b) = f(a', b). \end{array}$$

In other words, L consists of all arguments (a, b) where improving the second argument does not improve output, i.e. "f looks at its left argument first", and analogously for R.

We claim that the sets L and R are each closed under under directed suprema. Suppose that $(a_i, b_i)_{i \in I}$ is a directed family in L. Suppose $b' \sqsupseteq \bigsqcup_{i \in I} b_i$. Then for all a_i we have $f(a_i, b') = f(a_i, b_i)$. Accordingly, we have

$$f(\bigsqcup_{i \in I} a_i, \bigsqcup_{i \in I} b_i) = \bigsqcup_{i \in I} f(a_i, b_i) = \bigsqcup_{i \in I} f(a_i, b') = f(\bigsqcup_{i \in I} a_i, b') \tag{7}$$

and, therefore, $(\bigsqcup_{i \in I} a_i, \bigsqcup_{i \in I} b_i) \in L$. The proof of the claim for R is symmetric.

Now assume for the sake of contradiction that f is not inherently parallel. Then Lemma 3.3 states that $L \cup R = \mathcal{I}^2$. Then for all total $x \in [0, 1]$ we have $(x, \bot) \notin L$ as otherwise $f(x, y) = f(x, \bot)$ for all total $y \in [0, 1]$ which is impossible as $f(x, y_1) \neq f(x, y_2)$ for different $y_1, y_2 \in [0, 1]$ as f agrees with the midpoint operation at total elements. In particular, we have that $(0, \bot) \notin L$. Pick any $\epsilon > 0$. Since L is closed under directed suprema this means that already $(a, \bot) \notin L$ for some non-total element $a \in \mathcal{I}$ with $|a| \leq \epsilon$ and $0 \in a$. Since $L \cup R = \mathcal{I}^2$ we must therefore have $(a, \bot) \in R$.

By a similar argument as above we have for all total $y \in [0, 1]$ that the pair $(a, y) \notin R$ and that, therefore, for each such y we can find a non-total element b_y containing y such that $(a, b_y) \notin R$, and hence $(a, b_y) \in L$. Since the intervals b_y cover $[0, 1]$, compactness of $[0, 1]$ gives finitely many intervals $b_1, \ldots, b_n$ with

(1) $b_1 \cup \cdots \cup b_n = [0, 1]$ and

(2) $(a, b_i) \in L$ for $i = 1, \ldots, n$.

It follows from (2) that whenever $z \in b_i \cap b_j$ then $f(a, b_i) = f(a, z) = f(a, b_j)$. By iterating this argument using (1) we get that $f(a, b_i) = f(a, b_j)$ for any two i, j. It follows from (2) that $f(a, 0) = f(a, 1)$ because there are i and j with $0 \in b_i$ and $1 \in b_j$.

Summing up, for each $\epsilon > 0$ we can find an interval a with $0 \in a$ and with $|a| \leq \epsilon$ such that $f(a, 0) = f(a, 1)$. In particular $1/2 \in f(a, 0)$, and continuity of f then gives $1/2 \in f(0, 0)$, which contradicts the assumption that the function f extends the midpoint operation. □

4 Remarks and questions

This work could be improved in a number of directions.

Firstly, we ask whether the main theorem holds for any domain (among a class of domains of interest) with subspace of maximal elements (with the relative Scott topology) homeomorphic to the Euclidean real line or unit interval. More generally, we can consider domains with a (densely) embedded copy of line or unit interval.

In fact, there are many well-known variations of the interval domain, which have been used for different purposes. For example, one can consider: (1) The set of all non-empty closed and bounded real intervals — this fails to be a continuous Scott domain only by lacking a bottom element; to remedy this, just include the interval $(-\infty, +\infty)$. (2) The previous together with the empty interval $\emptyset$, a top element, so that one gets a continuous lattice — this is the original interval domain considered by Scott. (3) The set of non-empty compact intervals of the two-point compactification $[-\infty, +\infty]$ of the real line. (4) The set of non-empty compact connected subsets of the one-point compactification of the line. This is isomorphic to the poset of non-empty closed intervals of the circle. It is not a continuous Scott domain, but it is an FS-domain in the sense of Jung. (5) The set of all non-empty compact subsets of (some compactification of) the real line or of some subspace of the real line such as the unit interval.

It is easy to see that our result applies to these domains as well, with slight adaptations in the proofs. However, the domains having the real line as the subspace of maximal points are endless. For example, if R is any such domain and D is any domain with a top element, then $R \times D$ and $[D \to R]$ are also domains with the real line as the subspace of maximal points. We suspect that our result should apply to any domain environment for the line or unit interval.

But notice that this doesn't rule out the possibility of sequential models of (higher-type) real-number computation other than Scott domains. The following recent observation of Escardó and Streicher in discussions with Andrej Bauer in September 2002 rules out the category of strongly stable domains of Bucciarelli and Ehrhard [2]. Suppose we have a subspace embedding $\mathbb{R} \subseteq \mathcal{R}$ of the space of reals into some object $\mathcal{R}$ of the category (considered as a topological space under its Scott topology). Because the objects are countably based dI-domains and the morphisms are stable, there are only countably many open subsets of $\mathbb{R}$ of the form $\{x \in \mathbb{R} \mid p(x) = \top\}$ with $p : \mathcal{R} \to \Sigma$ a morphism of the category. Thus, there is no morphism $f : \mathcal{R} \times \mathcal{R} \to \Sigma$ realizing the less-than relation on the reals, that is, with $f(x, y) = \top$ of and only if $x < y$ for all $x, y \in \mathbb{R}$. As a consequence of these two observations we get that there is no subspace embedding $\mathbb{R} \subseteq \mathcal{R}$ for which addition and the predicate $(0 < x)$ for $x \in \mathbb{R}$ are stably realizable because such realizers would give rise to a stable realization of the binary less-than relation on the reals.

It remains to be investigated whether reals can be implemented in a sequential way as *subquotients* of strongly stable domains. From Kleene's function realizability and Weihrauch's Type 2 Effectivity approach we know that we can represent $\mathbb{R}$ as a quotient of Baire space and, therefore, as a subquotient of the strongly stable domain $\mathbb{N} \to \mathbb{N}_\perp$ in such a way that all desired operations and open subsets can be implemented in terms of strongly stable functions on $\mathbb{N} \to \mathbb{N}_\perp$. The reason is that sequences of natural numbers provide sufficient intensional information to simulate parallel search in a sequential way. However, a problem shows up at higher types. In function realizability $\mathbb{R} \to \mathbb{R}$ is implemented as a subquotient of $\mathbb{N} \to \mathbb{N}_\perp$ whereas one would rather desire a representation of $\mathbb{R} \to \mathbb{R}$ as a subquotient of $(\mathbb{N} \to \mathbb{N}_\perp) \to (\mathbb{N} \to \mathbb{N}_\perp)$. Whether this is possible remains as an open problem for future work.

Secondly, we have claimed that Weak Real PCF is manifestly sequential. While hardly anyone would cast doubt on this claim, one would like to have a mathematical formulation and proof. Amin Farjudian has recently performed some steps in this direction, by showing that all first-order Weak Real PCF definable functions are Vuillemin sequential [7] (moreover, his argument shows that any extension of the language with continuous first-order unary functions has the same property). A further case for manifest sequentiality of Weak Real PCF would be its conservativity over PCF.

Thirdly, it would be interesting to know how parallel the midpoint operation is. For instance, is (strong) parallel-or sequentially definable from it?

Acknowledgment This work was supported by an EPSRC grant number GR/M64840.

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] R.M. Amadio and P.-L. Curien. *Domains and Lambda-Calculi*, volume 46 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1998.

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

[4] A. Edalat. Domains for computation in mathematics, physics and exact real arithmetic. *Bulletin of Symbolic Logic*, 3(4):401–452, 1997.

[5] M.H. Escardó. PCF extended with real numbers. *Theoret. Comput. Sci.*, 162(1):79–115, 1996.

[6] M.H. Escardó and Th. Streicher. Induction and recursion on the partial real line with applications to Real PCF. *Theoret. Comput. Sci.*, 210(1):121–157, 1999.

[7] A. Farjudian. Sequentiality of wRPCF. School of Computer Science, University of Birmingham, UK, June 2002.

[8] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott. *Continuous lattices and domains*. Cambridge University Press, 2002. to appear.

[9] H. Luckhardt. A fundamental effect in computations on real numbers. *Theoret. Comput. Sci.*, 5:321–324, 1977.

[10] M. Mislove. Topology, domain theory and theoretical computer science. *Topology and its Applications*, 89(1–2):3–59, 1998.

[11] G.D. Plotkin. LCF considered as a programming language. *Theoret. Comput. Sci.*, 5(1):223–255, 1977.

[12] G.D. Plotkin. Pisa notes on domains. Department of Computer Science, University of Edinburgh. Available at http://www.dcs.ed.ac.uk/home/gdp/publications/, 1983.

[13] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967.

[14] D.S. Scott. Lattice theory, data types and semantics. In R. Rustin, editor, *Formal semantics of programming languages*, volume 2 of *Courant Computer Science Symposis*, pages 65–106. Prentice-Hall, 1972.