Computability on the Interval Space: A Domain Approach

Benjamín Callejas Bedregal and Benedito Melo acióly

Departamento de Informática e Matemática Aplicada - DIMAp
Universidade Federal do Rio Grande do Norte - UFRN
Campus Universitário - Lagoa Nova, Natal-RN, Brasil
bedregal@dimap.ufrn.br and bma@dimap.ufrn.br

Abstract

The effectively given continuous domains aims characterize the computable functions (as opposed to the merely continuous ones) and the computable elements of types represented by continuous domains. In this paper we show that computability on arbitrary effectively given continuous domain depends strongly upon the Church-Turing computability (classical computability on countable sets) on a countable base. In order to introduce the notion of computability on the interval space we need the concepts of effectively given continuous domain. In this approach, several desired properties of a computable interval analysis are obtained.

Key words: Interval Analysis, Domain Theory, Continuum Computability.

1 Introduction

Scientific computation is mainly directionated to solution of numerical problems. Of course convincing theoretical foundations are indispensible for computable analysis. Very different approaches have been used to investigate, from a constructive standpoint, concepts arising in real analysis such as real numbers, limits, derivatives and measure. An important difference among these approaches lies in the way real numbers are represented [Gia93]. In the notion of computable real functions given by Grzegorczyk's in [Grz57] is used the concept of computable operators on the set of natural numbers sequences. This notion was investigated and generalized by [Wei95]. In his approach an approximation of the output with arbitrary precision is computed from a suitable approximation of the input [Bra95]. Another approach was developed by Blum, Shub and Smale [BSS89]. Here real numbers are viewed as entities and the computable functions are generated from a class of basic functions (in a similar way to partial recursive functions). Although each of these approaches has its merits, none of them has been accepted by the majority of mathematicians or computer scientists.

Interval Mathematics is the branch of mathematics concerned with techniques and methods to compute real objects (like real numbers, real functions, etc....), mantaining a rigorous analysis of approximation errors. In exact real computation instead the result of a computation can be obtained with arbitrary precision, getting rid of the unfortunate phenomenon of the "round-off-error" [Gia93]. Using Moore interval arithmetic [Moo79] the round-off-error is known.

Computability on real intervals has not been object of specific studies, perhaps because we can derive computability notions on real intervals extending naturally the real functions to real interval ones. But, the natural extensions of the real computability to real interval computability (for example, those obtained by the extensions of real function to real interval functions as given by [Moo79]), because their behaviour when restrict to the reals numbers is a real function (i.e. sending degenerate intervals in degenerate intervals), do not consider severals functions, such as $f([a, b]) = [a - 1, b + 1]$, which are naturally computable. So it is necessary one defines a computability notion in real interval spaces which does not depend on the computabilty on real numbers.

In this paper, by using the well known domain theory, we will introduce an internal notion of computability for interval analysis supporting the computability properties which we would like that interval real functions had. For example, it is desirable that the interval arithmetic be computable, also each computable function be continuous, as well as that this notion of computability extends the computability on reals, etc.

2 Basic Concepts in Domain Theory

Let $\mathbf{D} = \langle D, \leq\rangle$ be a partially ordered set (poset). A set $\Delta \subseteq D$ is called *directed* if each of its finite subset has an upper bound, or equivalently, $\forall a, b \in \Delta \;\; \exists c \in \Delta$ such that $a \leq c$ and $b \leq c$. A poset $\mathbf{D}$ is *directed complete* (*dcpo* for short) if each directed set Δ has a least upper bound or supremum (denoted by $\bigsqcup\Delta$) and a *cpo* if it has a least element. We say that a is *way below* b (denoted by $a \ll b$) if for every directed set Δ such that $b \leq \bigsqcup\Delta$ then $a \leq x$ for some $x \in \Delta$. We let $\mathop{\Downarrow} x = \{y \in D : \; y \ll x\}$[1]. A dcpo $\mathbf{D}$ is called *continuous* if, for all $x \in D$, the set $\mathop{\Downarrow} x$

[1] Analogously, we let $\mathop{\Uparrow} x = \{y \in D : \; x \ll y\}$

is directed and $x = \bigsqcup \Downarrow x$. A set $B \subseteq D$ is a *base* of $\mathbb{D}$ if for each $x \in D$, $\Downarrow x \cap B$ is a directed set with $x = \bigsqcup \Downarrow x \cap B$. A dcpo is continuous if, and only if, it has a base.

An element $a \in D$ is *compact* if $a \ll a$. We will denote the set of all compact elements of $\mathbb{D}$ by $\mathbb{D}^0$. A continuous dcpo $\mathbb{D}$ is said to be *algebraic* if $\mathbb{D}^0$ is a base. A subset of a poset is *consistent* if it has an upper bound in the poset. A continuous dcpo is a *continuous domain* if it has a countable base and each consistent set has a supremum. An algebraic dcpo is a *Scott domain* if the set of compact elements is countable and each consistent set has a supremum.

Let $\mathbb{D}$ and $\mathbb{E}$ be dcpo's. A function $f : E \longrightarrow D$ is called *continuous* w.r.t. the orders or simplely continuous if it is monotonic ($x \leq y$ implies $f(x) \leq f(y)$) and preserves least upper bounds ($f(\bigsqcup \Delta) = \bigsqcup f(\Delta)$).

An element x of a poset $\mathbb{D}$ is said *total* if $x \sqsubseteq y$ implies that $x = y$. Let $Tot(\mathbb{D})$ denotes the set of total elements of $\mathbb{D}$.

There are several constructions on continuous domains, such as lifting, Cartesian product, function space, etc. Let $\mathbb{D} = \langle D, \sqsubseteq_D \rangle$ and $\mathbb{E} = \langle E, \sqsubseteq_E \rangle$ be continuous domains. The *Cartesian product* of $\mathbb{D}$ and $\mathbb{E}$ is the continuous domain $\mathbb{D} \times \mathbb{E} = \langle D \times E, \sqsubseteq \rangle$, where $(x, y) \sqsubseteq (x', y')$ if, and only if, $x \sqsubseteq_D x'$ and $y \sqsubseteq_E y'$. We will abreviate $\mathbb{D} \times \cdots \times \mathbb{D}$ by $\mathbb{D}^n$. The *function space* from $\mathbb{D}$ to $\mathbb{E}$ is the continuous domain $\mathbb{D} \rightarrow \mathbb{E} = \langle E^D, \sqsubseteq \rangle$ where E^D is the set of continuous functions from D to E and $f \sqsubseteq g$ if, and only if, $f(x) \sqsubseteq_E g(x)$ for each $x \in D$.

The partial orders are endowed with a topology, called Scott topology such that in some sense the notion of continuous function and order coincide.

Definition 1 *Let $\mathbb{D} = \langle D, \sqsubseteq \rangle$ be a dcpo. $\mathcal{O} \subseteq D$ is taken as open in the* **Scott topology** *on D if*

1. *$x \in \mathcal{O}$ and $x \sqsubseteq y$ then $y \in \mathcal{O}$*
2. *If $\Delta \subseteq D$ is a directed set and $\bigsqcup \Delta \in O$ then $\Delta \cap \mathcal{O} \neq \emptyset$*

The Scott topology on dcpo $\mathbb{D}$ will be denoted by $\mathcal{T}_{\mathbb{D}}$.

The idea by defining this topology is as follows: 1) if information x suffices to indicate that test $\mathcal{O}$ has succeeded, then any greater information is sufficient a fortiori; 2) if the limit of a sequence of better and better approximations passes a test $\mathcal{O}$, then some of the approximants already passes. The last requirement is connected with the idea that open sets correspond to finitary tests.

Proposition 2 *[Smy92] Let $\mathbb{D}$ be a dcpo. Then $\mathcal{T}_{\mathbb{D}}$ is a T_0 topology.* ■

Theorem 3 *[Smy92] Let $\mathbb{D}_1 = \langle D_1, \sqsubseteq_1 \rangle$ and $\mathbb{D}_2 = \langle D_2, \sqsubseteq_2 \rangle$ be dcpo's. A function $f : D_1 \longrightarrow D_2$ is continuous (as partial order) if, and only if, f is continuous topologically.* ■

Let $\mathcal{T}$ be a topology over a set X and $Y \subseteq X$. The subspace topology on Y induced by $\mathcal{T}$, denoted by $\mathcal{T}/Y$, is the relativization of $\mathcal{T}$ to Y, i.e. the sets open in the subspace are precisely those of the form $Y \cap \mathcal{O}$, where $\mathcal{O} \in \mathcal{T}$. Clearly, $\mathcal{T}/Y$ constitutes a topology on Y.

3 Effectively Given Continuous Domains

The original notion of effectively given domain was first presented by Smyth [Smy77] for continuous cpo's.

Definition 4 *Let $\mathbf{D} = \langle D, \sqsubseteq \rangle$ be a continuous domain, B a countable basis. The pair $\langle \mathbf{D}, B \rangle$ is an* **effectively given continuous domain**, *EGCO in short, if $\{(a, b) \in B \times B : a \ll b\}$ is a r.e. set.*

In the original definition of effectively given domain is used an enumeration for a base [Smy77, WS83]. But, by using the informal notion of r.e. set [2] as can be found in [Rog67] we does not need this enumeration simplifying the notion of effectively given continuous domain.

It is possible also naturally to extend all the constructors on continuous domains to effectively given continuous domains. For example, the Cartesian product of the effectively given continuous domains $\langle \mathbf{D}, B_D \rangle$ and $\langle \mathbf{E}, B_E \rangle$ is the effectively given continuous domain $\langle \mathbf{D} \times \mathbf{E}, B_D \times B_E \rangle$.

The notion of EGCD brings up discussions on computability. First to the elements of the domain and then to the continuous functions between domains.

Definition 5 *Let $\langle \mathbf{D}, B \rangle$ be an EGCD. An element $x \in D$ is* **computable** *if $\{b \in B : b \ll x\}$ is a r.e. set.*

Definition 6 *Let $\langle \mathbf{D}_1, B_1 \rangle$ and $\langle \mathbf{D}_2, B_2 \rangle$ be EGCD's. A continuous function $f : D_1 \longrightarrow D_2$ is* **computable** *if $G(f) = \{(a, b) \in B_1 \times B_2 : b \ll f(a)\}$ is a r.e. set.*

Proposition 7 *[Bed96] Let $\langle \mathbf{D}_1, B_1 \rangle$ and $\langle \mathbf{D}_2, B_2 \rangle$ be EGCD's. If $f : D_1 \longrightarrow D_2$ is a continuous function such that $graph(f) = \{(a, b) \in B_1 \times B_2 : b = f(a)\}$ is r.e. then f is Scott-computable.* ∎

In the following we will show that our notion of computable function agrees with that of the Church-Turing-computable function, in the sense that the computability of a function depends on the Church-Turing computability of a function between their respective basis (which are countable).

[2] Informally, a set is recursively enumerable (r.e. in short) if there exists an effective procedure which lists each element of the set (also it is permited repetitions).

Theorem 8 *Let $\langle \mathbf{D}_1, B_1\rangle$ and $\langle \mathbf{D}_2, B_2\rangle$ be EGCD's. If $f : D_1 \longrightarrow D_2$ is a computable function then there is a Church-Turing computable monotonic function $g : \mathbb{N}\times B_1 \longrightarrow B_2$ such that $f(x) = \bigsqcup\{b \in B_2 : g(n,a) = b$ for some $a \ll x$ and $n \in \mathbb{N}\}$.*

PROOF: Define the function $g : \mathbb{N} \times B_1 \longrightarrow B_2$ as following $g(n,x) = \pi_2(a,b) = b$ where (a,b) is the $n^{\underline{th}}$ pair in the set $G(f)$ such that $f(a) \ll f(x)$. Since $G(f)$ is a r.e. set, g is Church-Turing computable. So,

$$\begin{array}{rll} f(x) & = \bigsqcup\{b \in B_2 : b \ll f(x)\} & \text{by def. of base} \\ & = \bigsqcup\{b \in B_2 : b \ll f(x) \text{ for some } (a,b) \in G(f)\} & \text{by def. of } G(f) \\ & = \bigsqcup\{b \in B_2 : g(n,x) = b \text{ for some } a \ll x \text{ and } n \in \mathbb{N}\} & \text{by def. of } g \ \blacksquare \end{array}$$

Theorem 9 *[Bed96] Let $\langle \mathbf{D}_1, B_1\rangle$ and $\langle \mathbf{D}_2, B_2\rangle$ be EGCD's. If $g : B_1 \longrightarrow B_2$ is a Church-Turing computable monotonic function then there exists a computable function such that $f(x) = \bigsqcup\{g(y) : y \ll x\}$.* ■

Clearly, as consequence of this theorem the converse of the theorem 8 holds, i.e. if $g : \mathbb{N} \times B_1 \longrightarrow B_2$ is a Church-Turing computable monotonic function then $f(x) = \bigsqcup\{g(n,a) : n \in \mathbb{N} \text{ and } a \ll x\}$ is a computable function from D_1 into D_2. Thus the computability notion on continuous domains agrees strongly with the notion of Church-Turing computability, in the sense that computability on continuous domains extends the notion of Church-Turing computability to sets with the continuum cardinality. So, we can generalize the Church thesis to

Generalized Church-Thesis: A function $f : A \longrightarrow B$ is computable informally or by a physical device, if and only if there exists EGCD's $\langle\langle A, \sqsubseteq_1\rangle, B_1\rangle$ and $\langle\langle B, \sqsubseteq_2\rangle, B_2\rangle$ such that f is computable in the sense of the definition 6.

4 Computability on the Interval space

The notion of effective procedure is intuitive and not limited to countable sets, while the notion of computability on Turing machines and partial recursive functions limits the computable functions to countable ones. The theory of computability on countable sets is called *type 1 computability* by Weirhauch [Wei95] (here called Church-Turing computability) whereas the computability on sets with the cardinality of continuum is called by him *type 2 computability.*

The computability on sets with cardinality of the continuum has been more directioned to computability on the real numbers. Very different approaches have been used to investigate, from a constructive standpoint, main concepts arising in analysis such as real numbers, limits, derivatives and measures. Basically there exist two lines of researches to real computability. In the most accepted of them we computes with approximations of real numbers in order to obtain approximations of the output, with arbitrary precision [Gia93, Wei95, Bra95]. In other approachy seing the reals numbers are seen as finished entities and the computable functions are generated from a class of basic functions, the most known of the these approaches is is BSS [BSS89].

For other sets with the cardinality of the continuous, such as complex numbers and $\mathbb{R}^n$, the computability is derived from real computability. However, any tentative to extend the real computability for real intervals, denoted by $I(\mathbb{R}) = \{[r, s] : r \in \mathbb{R}$ and $r \leq s\}$, will be unnatural. This is so because, the interval extensions of a real function must preserve the behaviours of the real function to degenerate intervals (intervals with equal extremes, identified with the real numbers). For example, let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a function. An interval extension of f, is a function $F : I(\mathbb{R})^n \longrightarrow I(\mathbb{R})$, which must satisfy the following condition [Moo79]:

$$F([x_1, x_1], \cdots, [x_n, x_n]) = [f(x_1, \cdots, x_n), f(x_1, \cdots, x_n)]$$

So, interval functions such as $F([a, b]) = [a - 1, b + 1]$ are not interval extensions of any real function. Therefore, it is not derived from a real computable function. Clearly "interval extensions" which does not satisfy the Moore condition will be unsatisfactory (the interval identity function is not considered). So, we can not get a satisfatory computability theory for interval analysis from real computable analysis.

In one of his early papers on domain theory, D. Scott [Sco70] suggested that a cpo consisting of intervals with real numbers as end points, real intervals in short, could be used to study computability on the real numbers. The continuous domain of real intervals is defined by $\mathcal{R} = \langle \mathbb{I}(\mathbb{R}), \sqsubseteq \rangle$ where

- $\mathbb{I}(\mathbb{R}) = \{[r, s] : r, s \in \mathbb{R}$ and $r \leq s\} \cup \{[-\infty, +\infty]\}$
- $[r, s] \sqsubseteq [t, u]$ if, and only if, $r \leq t$ and $u \leq s$

with $\leq$ being the usual "lesser or equal" order on the extended real numbers (i.e. $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$). Observe that $[r, s] \sqsubseteq [t, u]$ if, and only if, $[t, u] \subseteq [r, s]$. The way below relation associated to this continuous domain is defined by

$$[r, s] \ll [t, u] \text{ if, and only if, } r < t \text{ and } u < s$$

A countable base for this continuous domain are the rational intervals, i.e. the set $\mathbb{I}(\mathbb{Q}) = \{[p, q] : p, q \in \mathbb{Q}$ and $p < q\}$. Since $\mathbb{I}(\mathbb{Q})$ is r.e. and $\ll$ is clearly decidible, the set $\{(I, J) \in \mathbb{I}(\mathbb{Q}) \times \mathbb{I}(\mathbb{Q}) : I \ll J\}$ also is r.e. So, $\langle \mathcal{R}, \mathbb{I}(\mathbb{Q}) \rangle$ is an EGCD and therefore we have a notion of computable real interval and computability for interval functions.

The *computable elements* (the computable elements of $< \mathcal{R}, \mathbb{I}(\mathbb{Q}) >$) are the real intervals which are supremum of r.e. chains of rational intervals. Clearly, from the definition of computable real number given by [ML70], the computable elements are exactly the intervals with Turing-computable real numbers[3] as end point, i.e. the intervals $[r, r]$ where r is Turing-computable.

Proposition 10 *[Bed96] Let x and y be real numbers such that $x \leq y$. $[x, y]$ is computable if, and only if, x and y are Turing-computable.* ■

By theorems 9 and 8, the computable interval functions on the EGCD $\mathcal{R}$ (*computable functions*) depend on the Church-Turing computability on the rational interval functions. So, a function $f : \mathbb{I}(\mathbb{R}) \longrightarrow \mathbb{I}(\mathbb{R})$ is **computable** if there exists a monotonic computable function $g : \mathbb{I}(\mathbb{Q}) \longrightarrow \mathbb{I}(\mathbb{Q})$ such that for each real interval $[r, s]$

$$f([r, s]) = \bigsqcup\{g([p, q]) : p < r \text{ and } s < q\}$$

or equivalenty if for each nested interval chain, $(I_i)_{i\in\mathbb{N}}$, of rational intervals converging to I

$$f(I) = \lim_{i\in\mathbb{N}} g(I_i). \tag{1}$$

The **interval space** is the class of EGCD which can be constructed from the EGCD $\mathcal{R}$ using the usual construction on continuous domain. We will call *interval domain* for the domain of interval space. Thus, $\mathcal{R}$ is an interval domain. If $\mathbb{D}$ and $\mathbb{E}$ are also interval domains then $\mathbb{D} \times \mathbb{E}$ and $\mathbb{D} \rightarrow \mathbb{E}$ are interval domains. Since, product and function space are closed on EGCD's, the notions of computable function and computable elements of a domains are extended to the interval space.

Proposition 11 *The following functions from $\mathbb{I}(\mathbb{R})^n$ to $\mathbb{I}(\mathbb{R})$ are computable.*

1. *The interval arithmetic functions and the identity function*[4].

2. $f_c(I) = c$ *where* $c \in \mathbb{I}(\mathbb{R})_e$, *i.e.* c *is a computable interval.*

3. $$|[r, s]| = \begin{cases} [0, max\{|r|, s\}] & , \textit{if } 0 \in [r, s] \\ [r, s] & , \textit{if } 0 < r \\ [-s, -r] & , \textit{otherwise} \end{cases}$$

[3]Basically, a Turing-computable real number is a real numbers which can be generated by a Turing machine [ML70].

[4]Since the interval division is not a total function (it is not defined to intervals containing 0), it is a partial computable function.

4. $exp\ [r,s] = [min\{e^r, e^s\}, max\{e^r, e^s\}]$

5. $log\ [r,s] = [log\ r, log\ s]$[5]

6.

$$sin[r,s] = \begin{cases} [-1,1] & , \text{if } 2\pi \leq s-r \\ [minS, 1] & , \text{if } \pi \leq s-r < 2\pi \text{ and } sin\ r + \frac{s-r}{2} > 0 \\ [-1, maxS] & , \text{if } \pi \leq s-r < 2\pi \text{ and } sin\ r + \frac{s-r}{2} < 0 \\ [minS, maxS] & , \text{otherwise} \end{cases}$$

where $S = \{sin\ r, sin\ s\}$. *item* $cos\ [r,s] = sin\ [r + \frac{\pi}{2}, s + \frac{\pi}{2}]$, $tg\ I = \frac{sin\ I}{cos\ I}$ *and* $sec\ I = \frac{1}{cos\ I}$

7. *Each polinomial interval function* $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ *where the* a_i*'s are computable intervals.* ■

An example of non computable function is the equality to zero, i.e. the function $eq0 : \mathbb{I}(\mathbb{R}) \longrightarrow \mathbb{I}(\mathbb{R})$ defined by

$$eq0([r,s]) = \begin{cases} [1,1] & , \text{if [r,s]=[0,0]} \\ [0,0] & , \text{otherwise} \end{cases}$$

This is true because, trivially, this function is non monotonic, that is $[-1,1] \sqsubseteq [0,0]$, but $eq0([-1,1]) = [0,0] \not\sqsubseteq eq0([0,0]) = [1,1]$. Therefore $eq0$ is non continuous. So, $eq0$ is non computable. We also could think in $\{0,1\}$, i.e. $\{[0,0],[1,1]\}$ as a domain with two element. But independently of how we order them, this function remains non continuous.

It is a consensus in real computability that each computable function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is continuous w.r.t. the Euclidian topology. So, it is reasonable to ask that our notion of computability also be continuous w.r.t. some natural topology on the real interval. In the interval space endowed with the metric topology given by R. Moore [Moo79] there is computable function which is computable regard to our sense and the other senses above mencioned but which is not continuous. This fact is possible because the computability on the metric topology by Moore and the inclusion monotonicity property are not compatible, i.e. there exists monotonic function which is not continuous [Moo79]. Still, if we consider de Scott topology on the continuous domain $\mathcal{R}$ the interval monotonic function and the computable interval functions are continuous.

[5] Since the real logarithms is a function from $\mathbb{R}^+$ (positive real numbers set) into $\mathbb{R}$, their interval version is a function from $\mathbb{I}(\mathbb{R})^+$ to $\mathbb{I}(\mathbb{R})$, where $\mathbb{I}(\mathbb{R})^+$ is the natural subdomain of $\mathbb{I}(\mathbb{R})$.

Theorem 12 *Each interval computable function is continuous w.r.t. the Scott topology on* $\mathbb{I}(\mathbb{R})$.

PROOF: By definition each interval computable function is continuous with regard to the order. Therefore, by the proposition 3, the interval computable functions are continuous in the Scott topology. ■

5 Conclusion

In domain theory our main contribuition consisted of giving a preponderant role for a countable basis to show the computability of functions. We saw that a function on sets with the cardinality of the continuum is computable if, and only if, there exists a function on their respective countable bases which goes computing finite aproximations of infinite object in such way that in the limit, the function computes the ideal object. This result allows us to introduce a computability notion on real interval based on the computability on the rational interval, i.e. classical computability. We showed that the interval computability has several desired properties that a computability theory on interval analysis must consider. These notions can be extend to the interval space.

We have used the established domain theory in order to introduce a reliable notion of computability on the interval space. But, in the future we will can exclude this theory and to introduce our computability theory on the interval space in a direct way, i.e. like in equation (1) above.

References

[Aci91] Benedito Melo Acióly. *Computational Foundation of Interval Mathematics* (in portuguese). PhD thesis, CPGCC da UFRGS, Porto Alegre, 1991.

[Bed96] Benjamín R. Callejas Bedregal. *Continuous Information Systems: A Computational and Logical Approach to Interval Mathematics* (in portuguese). PhD thesis, UFPE-Depto. de Informática, Recife, 1996.

[Bra95] Vasco Brattka. *Recursive Characterization of Computable Real-valued Functions and Relations.* Preprint submitted to Elsiever Science, 1995.

[BSS89] L. Blum, M. Shub and S. Smale. *On a Theory of Computation and Complexity over Real Number: NP-completness, recursive functions and universal machines.* Bull. of the Amer. Math. Soc. 21, 1989, pg. 1-46.

[Gia93] Pietro Di Gianantonio. *A Functional Approach to Computability on Real Numbers.* PhD thesis, Università di Pisa-Genova-Udine, Italy, march 1993.

[Grz57] A. Grzegorczyk. *On the definitions of computable real continuous functions.* Fundamenta Mathematicae, 44(1), 1957.

[ML70] Per Martin-Löf. *Notes on Constructive Mathematics.* Almqvist & Wiksell, Stockholm, 1970.

[Moo79] Ramon E. Moore. *Methods and Applications of Interval Analysis.* SIAM Studies in Applied Mathematics, philadelphia, 1979

[Rog67] Hartley Rogers Jr. *Theory of Recursive Functions and Effectively Computability.* McGraw-Hill Book Company, New York, 1967.

[Sco70] Dana Scott. *Outline of a Mathematical Theory of Computation.* In $4^{\underline{th}}$ annual Princeton Conference on Inf. Science and Systems, pages 65-106, 1970

[Smy77] Myke Smyth. *Effectively Given Domains.* Theoretical Computer Science, pages 257-274, 1977.

[Smy92] Myke Smyth. *Topology.* In S.Abramsky, D.M. Gabbay and T.S.E. Maimbaum, editors, Handbook of Logic in Computer Science. Vol. 1. Oxford University Press, 1992.

[Wei95] Klaus Weihrauch. *A simple Introduction to Computable Analysis.* Technical Report 171-2/1995, Fern Universitat, 1995.

[WS83] Klaus Weihrauch and G. SchÄfer. *Admissible representations of effective cpo's.* Theoretical Computer Science, 26:131-147, 1983.