Applications of the Kleene-Kreisel Density Theorem to Theoretical Computer Science

Dag Normann *

March 20, 2006

Abstract

The Kleene-Kreisel density theorem is one of the tools used to investigate the denotational semantics of programs involving higher types. We give a brief introduction to the classical density theorem, then show how this may be generalized to set theoretical models for algorithms accepting real numbers as inputs and finally survey some recent applications of this generalization.

1 Introduction

Classical Computability Theory is the study of what may actually be computed, when the objects used for inputs are finite entities like integers or words in a finite alphabet. Of course, relativized computability, complexity issues and other aspects of genuine computations will be considered to be in the realm of classical computability theory as well.

In *Generalized Computability Theory* we analyze mathematical structures that support alternative forms of computations, or computation-like phenomena. Normally, it is the mathematical structures that are important, the concepts of computability are adjusted to these structures. We will use "CT" for "Computability Theory" and "GCT" for "Generalized Computability Theory".

*Department of Mathematics, The University of Oslo, P.O. Box 1053, Blindern N-0316 Oslo, Norway. e-mail: dnormann@math.uio.no

In Computer Science the strategy i often different. There the actual programs and computations are what is important, and the mathematical models used for denotational semantics are of importance only to the extent they help us analyze the programming languages and programs. The split in attitude between CT and TCS (Theoretical Computer Science) is not absolute, and the same concepts of computability over the same mathematical structures will occasionally be studied both in CT and in TCS.

In this paper we will give a survey of the theory for continuous functionals of higher types, and we will focus on recent nontrivial applications of the Kleene-Kreisel density theorem and its generalizations. The paper will be semi-technical, with some formal definitions, but only indications of proofs. For a general overview of the interplay between the CT-approaches and the TCS-approaches to computations in higher types in general, see Normann [20].

In 1959 Kleene [13] and Kreisel [14] introduced what is now known as the *Kleene-Kreisel continuous functionals*. The motivation behind the two papers were different, the two approaches were different, and in fact, the two concepts of *countable functional* due to Kleene and of *continuous functional* due to Kreisel are not equivalent. Still, both authors claimed that the two approaches were essentially equivalent, without offering proofs. In Section 2 we will give a definition based on domain theory. We will let $Ct(\sigma)$ be the set of total continuous functionals of finite type σ.

Kleene's aim was to find a natural notion of computations relative to higher type objects. He observed that there is a natural sub-hierarchy, the *countable functionals*, of the full type structure that is closed under his notion of computability. Kreisel's motivation behind introducing the continuous functionals of finite type was to give an interpretation of the constructive content of a statement in second order number theory. This should enable him to decide in an absolute way wether a statement in analysis is constructively true or not. We will not go into this analysis here.

In both approaches, the aim was to construct a hierarchy of total functionals of finite type, where the action of one functional Ψ on an input F is locally determined via finite "approximations" to Ψ and F. Kleene used *associates*, i.e. functions in $\mathbb{N}^{\mathbb{N}}$, to represent the functionals, while Kreisel used certain "total" ideals of formal neighborhoods. The Kleene-Kreisel density theorem was formulated in two ways:

Kleene: The set of finite sequences that may be extended to an associate for an object of type σ is computable.

Kreisel: Each formal neighborhood can be extended to a total ideal.

The density theorem was used to prove this now classical result:

Theorem 1 [Kreisel]
Let $A \subseteq \mathbb{N}^{\mathbb{N}}$ be Π^1_k where $k \geq 1$.
Then there is a primitive recursive relation R in $\mathbb{N}^{\mathbb{N}} \times Ct(k) \times \mathbb{N}$ (where k denotes the pure type at level k) such that

$$f \in A \Leftrightarrow \forall \Phi \in Ct(k) \exists n \in \mathbb{N} R(f, \Phi, n).$$

The density theorem enables us to systematically replace a quantifier $\exists F \in Ct(k-1)S(F)$ by $\exists n \in \mathbb{N}S(F_n)$ where $\{F_n\}_{n\in\mathbb{N}}$ is an effectively enumerated dense subset, and S is a predicate that is computable in some parameters.

2 A modern view of the Kleene-Kreisel functionals

The two approaches to the continuous functionals discussed in Section 1 both belong to the CT-tradition. There is, however, an alternative approach more natural from the point of view of TCS, the approach via domains. In reality, this approach is not far from Kreisel's, but in spirit there is a certain gap. In CS it is important to model partiality, so the main structure is a hierarchy of partial continuous functionals. In order to define this hierarchy, we use domain theory, initiated by Scott [31] and by Ershov [9]. We will start with a brief introduction to domain theory. See Stoltenberg-Hansen, Lindström and Griffor [33], Abramsky and Jung [1] or Amadiou and Curien [2] for detailed introductions.

2.1 Scott-domains

A *complete partial ordering*, a *cpo* for short, is a partial ordering $(X, \sqsubseteq)$ such that each directed subset has a least upper bound $\sqcup A$. A *cpo* is *bounded complete* if each bounded set will have a least upper bound. Since the empty set is directed, a *cpo* X will have a least element, named $\bot_X$ or just $\bot$. $x \in X$ is *compact* or *finitary* if for each nonempty directed set A with $x \sqsubseteq \sqcup A$ there

will be an $a \in A$ such that $x \sqsubseteq a$.
A *cpo* $(X, \sqsubseteq)$ is an *algebraic domain* if for each $x \in X$, the set

$$C_x = \{x_0 \in X \mid x_0 \text{ is compact and } x_0 \sqsubseteq x\}$$

is directed and $x = \sqcup C_x$.

If X is bounded complete, C_x as defined above will always be directed. In this paper we will restrict ourselves to *Scott domains*, i.e. bounded complete algebraic domains where the set of compacts is countable.

If $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ are two *cpo*'s, we define a function $f : X \to Y$ to be *continuous* if f is monotone, and for each directed set $A \subseteq X$ we have that

$$f(\sqcup_X A) = \sqcup_Y \{f(a) \mid a \in A\}.$$

If we use the pointwise ordering of the continuous functions from X to Y, we obtain a new *cpo*, and if we in addition are dealing with Scott domains, the function space will be a Scott domain. We are not going to prove this here. The key to the argument for Scott domains is to characterize the compacts in $X \to Y$ as the least upper bounds of finite bounded sets of step functions, where whenever p is a compact in X and q is a compact in Y, we define the step function $f_{p,q}$ by

$$f_{p,q}(x) = q \text{ if } p \sqsubseteq_X x.$$

$$f_{p,q}(x) = \bot_Y \text{ if } p \not\sqsubseteq_X x.$$

If we use the continuous functions as morphisms, the Scott domains form a category which is cartesian closed.

There is a natural, non-Hausdorff topology on a Scott domain, known as the Scott topology:

Definition 2 The Scott topology on a Scott domain $(X, \sqsubseteq)$ is the topology generated from the base consisting of all

$$\{x \in X \mid x_0 \sqsubseteq x\}$$

where x_0 varies over all compacts in X.

A subset $H \subseteq X$ will be *dense* if it is dense with respect to the Scott topology, which means that every compact will have an extension in H. We justify our definition of continuous function from X to Y by observing that this means continuous with respect to the Scott topologies on X and Y.

In a bounded complete *cpo*, a set A is bounded if and only if each finite subset A_0 is bounded. If in addition, a set A is bounded whenever all subsets A_0 with at most two elements are bounded, the *cpo* is a *coherence space*. It is a basic and easy fact of domain theory that $X \to Y$ is a coherence space whenever X is an algebraic domain and Y is a coherence space.

2.2 Partial continuous functionals of finite type

Scott [31] introduced a formal logic LCF. In the language of LCF we have terms expressing functionals of higher types and in the formal theory we can reason about the relationship between various terms. His language is based on typed combinators, and he introduced a set of typed constants to be interpreted as the least fixed point operators at each type.

In this (for a long time) unpublished paper, Scott also gave a set-theoretical model for LCF. One important motivation at the time was that untyped λ-calculus lacked a set-theoretical model, and Scott suggested that LCF could replace λ-calculus for any practical purpose. We are not going to discuss the motivation of Scott further. To us, what is of interest is that Scott gave an interpretation $D(\sigma)$ for each finite type σ and by that laid the foundation of domain theory. E.g. the natural numbers will be interpreted as the set $D(\iota) = \{\bot, 0, 1, \ldots\}$ where $\bot$ signifies *the undefined* and where $a \sqsubseteq b \Leftrightarrow a = b \wedge a = \bot$. This domain is known as the *flat domain* of natural numbers and is one of the base domains in the semantics for LCF. Another base domain will be the similar flat domain $D(o)$ of boolean values $\{\bot, tt, ff\}$. Then each finite type σ over the base type ι for the natural numbers and o for the booleans is interpreted as $D(\sigma)$ in the cartesian closed category of Scott domains. $D(\sigma)$ will also be a coherence space.

It is in order to make the interpretation of the constants for the fixed point operators possible that we need *cpo*'s, if $f : X \to X$ is continuous, then $\sqcup\{f^n(\bot) \mid n \in \mathbb{N}\}$ will actually be a least fixed point of f.

Developing the theory of enumerations, Ershov came up with concepts quite equivalent to Scott domains. Since it is irrelevant to our story, we will not give any details. Ershov's characterization of the Kleene-Kreisel continuous functionals can be found in [9].

Though being methodologically inspired from CT, LCF is a contribution to TCS, and [31] turned out to be an influential paper beyond its CT-content. Plotkin [28] reformulated LCF into the typed λ-calculus PCF. Special PCF is typed λ-calculus with constants for all natural numbers and boolean values, constants for the successor function, the predecessor function, the boolean test of zero-hood and the conditionals over the two base types ι and o. In addition there are constants for the fixed point operator at every type. There are conversion rules from λ-calculus together with special rules for all the special constants.

Later we will discuss an application of the density theorem to a problem concerning PCF, see Section 3. For now, let us mention three important results:

1. To the extent it makes sense, Kleene's $S1 - S9$ - computability interpreted over the partial computable functionals is equivalent to PCF.
2. There is a compact object of mixed type 1 that is not the interpretation of any PCF - term.
3. If a closed PCF - term t of base type is interpreted as an element $\neq \bot$, then we may rewrite t to the constant for the interpretation using the conversion rules.

The proof of 1. goes back to Platek's thesis [27], where it is proved in the discontinuous case. A proof can also be found in Moldestad [18]. 2. is also known since Platek [27], and is observed in Scott [31] and Plotkin [28]. It is relevant to us to observe that this compact, an interpretation of *parallel or*, cannot be extended to a PCF - definable object. Thus the PCF - definable objects do not form a dense subset of the underlying domains seen as topological spaces. 3. is one of the main results of Plotkin [28] and is known as the *Plotkin adequacy theorem*.

2.3 Hereditarily total functionals

We may construct the Kleene-Kreisel continuous functionals from the partial continuous functionals via the so called *hereditarily total functionals*.

Definition 3 a) For each finite type σ over the base types ι and o we are going to define the set $H(\sigma) \subset D(\sigma)$ of hereditarily total elements:

1. $H(\iota) = \mathbb{N}$ and $H(o) = \mathbb{B} = \{tt, ff\}$.
2. Let $\sigma = \tau \to \delta$ and assume that $H(\tau)$ and $H(\delta)$ are defined. Let $f \in D(\tau \to \delta)$.
   Then
$$f \in H(\sigma) \Leftrightarrow \forall x \in H(\tau)\ (f(x) \in H(\delta)).$$

b) By recursion on the type σ we define an equivalence relation $\approx_\sigma$ on $H(\sigma)$ as follows:

   1. $\approx_\iota$ and $\approx_o$ are the identity relations on $\mathbb{N}$ and $\mathbb{B}$ respectively.
   2. If $\sigma = \tau \to \delta$ and f and g are in $H(\sigma)$, we let

$$f \approx_\sigma g \Leftrightarrow \forall x \in H(\tau) \forall y \in H(\tau)(x \approx_\tau y \Rightarrow f(x) \approx g(y)).$$

Longo and Moggi [16] observed that we with ease may prove that

$$x \approx_\sigma y \Leftrightarrow x \sqcap y \in H(\sigma)$$

when x and y are in $H(\sigma)$. It then follows that $\approx_\sigma$ is an equivalence relation, something that had been seen as a consequence of the density theorem until then.

The best we can do within PCF with respect to the density theorem is to prove

Proposition 4 *Let σ be a finite type, and let $x_0 \in D(\sigma)$ be compact. Then there is a hereditarily total PCF-definable $x \in H(\sigma)$ such that x and x_0 are consistent.*

Though we have dropped every detail that could verify our claims, we are now in the position of giving a precise definition of the continuous functionals of finite type:

Definition 5 By recursion on the finite type σ we define the set $Ct(\sigma)$ of *continuous functionals of type* σ together with the surjective map $\rho_\sigma : H(\sigma) \to Ct(\sigma)$ as follows:

1. $Ct(\iota) = \mathbb{N}$ and $Ct(o) = \mathbb{B}$. ρ_ι and ρ_o are the respective identity maps.

2. Let $\sigma = \tau \to \delta$ and assume that $Ct(\tau)$, ρ_τ, $Ct(\delta)$ and ρ_δ are defined. If $F : Ct(\tau) \to Ct(\delta)$ and $f \in H(\sigma)$, we let $F = \rho_\sigma(f)$ if for all $x \in H(\tau)$ we have that

$$F(\rho_\tau(x)) = \rho_\delta(f(x)).$$

We then define $Ct(\sigma)$ as the image of ρ_σ.

We need to establish a few facts before we can claim that this definition makes sense for all types. The following lemma is easy to prove by induction on the type:

Lemma 6 *For every type σ we have that $\rho_\sigma(f)$ is defined for each $f \in H(\sigma)$ and that when f and g are in $H(\sigma)$ we have that*

$$f \approx_\sigma g \Leftrightarrow \rho_\sigma(f) = \rho_\sigma(g).$$

Each set $H(\sigma)$ will have a topology inherited from the Scott topology on $D(\sigma)$, and thus induce a quotient topology on $Ct(\sigma)$ via the identification map ρ_σ.

There are numerous characterizations of the Kleene-Kreisel functionals. The characterizations are mainly of two kinds

1. We choose a way to model partial computable functionals and then we extract the hereditarily total objects.
2. We construct the hierarchy directly by imposing a superstructure, like a limspace structure or a topology, at each step.

The interesting fact is that, in particular for the first category, most conceptually well based approaches lead to the same hierarchy of total functionals, even though the philosophy behind the superstructure (partial objects, limstructure etc) may differ.

2.4 The density theorem

Definition 7 By recursion on the type σ we will define the n'th approximation $(a)_n$ to any element a of $D(\sigma)$:

1. If $\sigma = \iota$ and $m \in \mathbb{N}$, we let $(m)_n = min\{n, m\}$. $\bot_n = \bot$.
2. If $\sigma = o$ and $a \in \mathbb{B}_\bot$, we let $(a)_n = a$.
3. If $\sigma = \tau \to \delta$, $f \in D(\sigma)$ and $a \in D(\tau)$ we let $(f)_n(a) = (f((a)_n))_n$.

The following lemmas are trivial:

Lemma 8 *For each type σ and each $n \in \mathbb{N}$, $\{(a)_n \mid a \in D(\sigma)\}$ is finite.*

Lemma 9 *For each type σ, each $a \in H(\sigma)$ and $n \in \mathbb{N}$, $(a)_n \in H(\sigma)$.*

Lemma 10 *Let σ be a type and let $a \in D(\sigma)$ be compact. Then there is a number n such that for all $m \geq n$ we have*

$$a \sqsubseteq (a)_m.$$

We will then have

Theorem 11 [Density Theorem] *Let σ be a finite type.*

a) Consistency (boundedness) is an equivalence relation on pairs from $H(\sigma)$

b) Each compact $a \in D(\sigma)$ has an extension to an element of $H(\sigma)$.

The density theorem is proved over and over again in the literature, [13, 14, 9, 4] and in surveys in general. In the proof, a) and b) are proved by simultaneous induction on the type. In order to prove b) one proves that for each compact a of type σ there is a total Φ of type σ such that a is consistent with $(\Phi)_m$, where $m \geq n$ is as in Lemma 10.

2.5 Kleene-schemes

The Kleene schemes S1 - S9 were introduced in [12]. They can be seen as nine clauses in a grand inductive definition defining the relation

$$\{e\}(\phi_1, \ldots, \phi_k\} \simeq a$$

where each ϕ_i will be a functional of pure type and $a \in \mathbb{N}$. The interpretation of these schemes will depend on the typed structure at hand. To us, three such structures are of interest, the full type structure $\{Tp(n)\}_{n\in\mathbb{N}}$ of total functionals, the Kleene-Kreisel functionals $\{Ct(n)\}_{n\in\mathbb{N}}$ and the Scott hierarchy $\{D(n)\}_{n\in\mathbb{N}}$. We will not give a detailed introduction to S1 - S9, see the original paper [12] or any later survey. The three first schemes introduce basic arithmetical functions like identity, successor and constants. S4 is composition, S5 primitive recursion, S6 permutation of variables and S7 represents the application operator on $(\mathbb{N} \to \mathbb{N}) \times \mathbb{N}$. S8 is a combination of application and composition at higher types:

$$\{e\}(\Phi^{k+2}, \phi_1, \ldots, \phi_n) \simeq \Phi(\lambda\xi^k.\{d\}(\xi, \Phi, \phi_1, \ldots, \phi_n)),$$

where e depends on d.

If we interpret this scheme over the full type structure or over the Kleene-Kreisel type structure, we require that $\lambda\xi^k.\{d\}(\xi, \Phi, \phi_1, \ldots, \phi_n)$ is total, while this is relaxed in the Scott hierarchy since non-total functionals are present there. As we will see, this has a dramatical effect on the computational power of S1 - S9.
Tait [34] observed that the *fan functional* is not S1 - S9 - computable over the Kleene-Kreisel functionals. Later, Martin Hyland [10, 11] showed that the functional Γ, defined by Gandy, is not S1 - S9 - computable relative to the fan functional. Finally, Normann [19] showed that for any type $k \geq 3$ and any continuous Φ of type k there is a continuous Ψ of type k that is not S1 - S9 - computable relative to Φ and any continuous functional of lower type.

However, if we move to type $k + 1$ we can find a functional Φ with a computable associate such that every $\psi \in Ct(k)$ is uniformly μ-computable relative to Φ and any associate for ψ. This was also proved in [19].

3 The Cook-Berger Problem

Though S1 - S9 interpreted over the Kleene-Kreisel continuous functionals is an interesting example of a computation theory, the relevance to TCS is rather meager. In modeling real computations, partiality is an important aspect. Moreover, the requirement in the traditional interpretation of S8:

$$\{e\}(\Phi, \vec{\phi}) \simeq \Phi(\lambda\xi\{d\}(\xi, \Phi, \vec{\phi}))$$

that $\lambda\xi\{d\}(\xi, \Phi, \vec{\phi})$ must be total, is rather unnatural. On the contrary, it is natural to imagine that we may have an algorithm for Φ that now and then consults $\lambda\xi\{d\}(\xi, \Phi, \vec{\phi})$ and after finitely many steps provides the value of $\Phi(\lambda\xi\{d\}(\xi, \Phi, \vec{\phi}))$.

PCF, with its operational semantics, is actually making this idea precise, and then, in a denotational semantics for PCF we have to accept a more liberal interpretation of what corresponds to S8. Again, this is obtained by interpreting S8 over the Scott-hierarchy of partial continuous functionals.

What is interesting from a foundational point of view is that when we customize S8 to TCS, we increase the computational power of S1 - S9 considerably. One simple example is the functional Γ defined by the equation

$$\Gamma(F) = F_0(\lambda n.\Gamma(F_{n+1})).$$

Using the fixed point operator in PCF, we see that Γ is a well defined functional, and that actually $\Gamma \in H(3)$. If we interpret the index obtained by the recursion theorem in the Kleene-Kreisel hierarchy, we just get the nowhere defined functional. As remarked above, Γ is not S1 - S9 - computable in this latter sense. This shows that relaxing the requirements on S8 increases the computational power.

Apparently, Robin Gandy (unpublished) had made a similar observation for the fan functional. The fan functional essentially evaluates the modulus of uniform continuity of a continuous $F \in Ct(2)$ over a compact $C_f = \{g \in \mathbb{N}^{\mathbb{N}} \mid g \leq f\}$ in $\mathbb{N}^{\mathbb{N}}$. Independently, Berger [4] showed that there is a representative $\Phi \in H(3)$ for the fan functional that is S1 - S9 - computable in the Scott - hierarchy sense. Simpson [32] used this to show that we may write a PCF-program for Riemann integration if the reals are represented by the elements of a given σ-compact subset of $\mathbb{N}^{\mathbb{N}}$. We will not go into detail, but the point is that we may write programs for functions of interest in mathematics using PCF or S1 - S9 and suitable representations of the data as domain elements. Thus it is of foundational interest to see to which extent we are able to write programs for functionals. Cook asked if -, and Berger suggested that all Kleene-Kreisel functionals that have computable associates also have hereditarily total representatives that are PCF-computable. It turned out that this is true in a rather strong sense:

Theorem 12 (Normann [21])
Let σ be a finite type. There is a PCF-definable function $EVAL$ of type $(\iota \to \iota) \to \sigma$ such that whenever $a \in H(\sigma)$ and f enumerates (the Gödel numbers of) the set of compacts in $D(\sigma)$ bounded by a, then $EVAL(f) \in H(\sigma)$ and $EVAL(f) \sqsubseteq a$.

It will lead too far to give the details of the construction, see [21] or [22]. In [22] we only prove the theorem for type 3. There are some obstacles that are not visible at this level. Here we will explain how the density theorem is used.

For types ≤ 2 the theorem is quite easy, and that the standard way of computing a total functional of type 2 from an enumeration of its approximations actually will give a $G \sqsubseteq F$ but not F itself in all cases. If we know that $\{f_{p_n,b_n}\}_{n\in\mathbb{N}}$ are step functions approximating F and f is given, we may for each n ask if f is consistent with p_n or not. If it is, we let $G(f) = b_n$ while if f is not consistent with p_n we go to the next step function approximating F for help.

If ϕ is of type 3, we may try a similar strategy. The problem is that consistency of an $F \in H(2)$ and a compact $p \in D(2)$ is not decidable; we just have that inconsistency is semi-decidable. In order to semi-decide inconsistency between $F \in H(2)$ and a compact $p \in D(2)$ we search through an effectively enumerated dense subset of the domain of p for an argument ξ such that $p(\xi)$ and $F(\xi)$ differ.

Now we let $\{f_{p_n,b_n}\}_{n\in\mathbb{N}}$ be an enumeration of all step functions bounded by $\phi \in H(3)$, and we let $F \in H(2)$. One ingredient in the algorithm for $\psi(F)$ is that we for each n search for a witness to the fact that p_n is inconsistent with F. If we find a witness, we just go on to $n+1$. If we do not find a witness, and $p_n \sqsubseteq F$ the result of the search will be a partial object q for which $p_n(q) = F(q) \in \mathbb{N}$ and we conclude that $\psi(F) = b_n$. The problematic case is when neither of these occur, and then we have in parallel to search for a later stage actually verifying that $\psi(F) = b_n$ in any case, and then it does not matter that we found no witness either way. The functional ψ constructed this way will then be total, and $\psi \sqsubseteq \phi$. This argument explains where density is used, but is of course not complete.

At type 3 we only use that there is an effectively enumerated dense set of objects of type 1. At higher types we have to use the full density theorem to the same effect. The obstacle is that not every functional constructed in order to verify the density theorem will be PCF-definable. Thus we have to use induction on the type, applying our construction to the effective enumerations of the approximations to the witnesses to the density theorem. We will not discuss the details here.

The Scott model is not fully abstract for PCF. This means that there are compacts in the Scott model that are not PCF-definable. Milner [17] showed that there exist one, and up to isomorphism, only one fully abstract model for PCF consisting of algebraic domains. Of course we may define the class of hereditarily total objects in Milner's model as well, consistency will be an equivalence relation on these objects, so we may even form the extensional collapse of the hereditarily total elements of Milner's model. Plotkin [29] showed as a corollary of Theorem 12 that this construction leads us to the Kleene-Kreisel continuous functionals. Thus Milner's construction is an alternative characterization. Recently Longley [15] showed that under reasonable general assumptions constructions of an extensional hierarchy of total continuous functionals containing $\mathbb{N}^{\mathbb{N}}$ at type 1 tends to end up with the Kleene-Kreisel functionals. Longley's proof is an elaboration on the argument from [21], where showing that we can simulate the effect of the density theorem is a nontrivial ingredient.

4 Replacing the natural numbers with the real numbers

Up to now, we have considered various models for computing relative to functionals of finite types over the natural numbers. Although we accept inputs to our algorithms that are infinite, the sets of natural numbers and boolean values are at the bottom, and these are discrete sets of data easily represented as digital data in a computer. As long as we are not involved in complexity issues, the nature of the representation is not important.

If we want to use other data-types, such as the real numbers, the set of differentiable functions on the complex numbers, certain Polish spaces or even structured Polish spaces like Banach spaces, the situation is quite different. Then in addition to discuss which principles for forming valid algorithms we will accept and which superstructures we will find useful for the denotational semantics, we also have to discuss how to represent the basic data in digital form. Given a representation of basic data, there will be a conflict of interest:

1. If we write programs utilizing the representation, we may get more efficient programs and we may be able to write programs solving more problems that we otherwise might.

2. If we hide the representation and write programs in a language based on the algebra of the data-type itself, it may be easier to analyze each program and to verify correctness.

It is well established that the decimal representation of real numbers is not suitable for modeling computability. Moreover, traditional constructions of $\mathbb{R}$ within set theory like the set of Dedekind cuts or the set of equivalence classes of Cauchy sequences, do not lead to useful computational models either. As is customary in constructive analysis, Cauchy-sequences with a prefixed rate of convergency works much better.

In this paper we will consider two ways of representing reals, one improvement of the Dedekind cut representation and one improvement of the Cauchy sequence representation. Though both approaches are good for analyzing what we mean by computable reals and computable functions on the reals, we will see that they may make a difference for the objects of higher types. In both cases there will be domains of partial computable functionals, and we will extract the hierarchies of total ones as substructures of particular interest.

4.1 The extensional hierarchy

Given the Dedekind cut of a real x, we may approximate x by half open rational intervals $(p, q]$ by first looking for a pair p_0, q_0 with distance ≤ 1 where p_0 is in the cut and q_0 is not, and then we recursively test if $\frac{p_n+q_n}{2}$ is in the cut or not in order to decide if this is p_{n+1} or q_{n+1}. The problem with cuts is that they are asymmetric, and it has turned out to be better to work with closed rational intervals as approximations. We let $R_0(0)$ be the set of closed rational intervals ordered by reverse inclusion (0 will denote the one base type now), and we let $R(0)$ be the set of ideals in this ordered set. An ideal $\mathcal{I}$ will determine a closed interval I of reals, the intersection of the ideal. We let an ideal be *total* if it determines an interval $[x, x]$ of length zero, and then we say that it *represents* x. Rational numbers can be represented in three ways, while irrational numbers are only represented by one ideal each.

Remark 13 We have chosen to use the algebraic domain version of the closed interval way of approximating reals. Even more common is the *continuous domain* of all closed intervals ordered by reverse inclusion. The point with both these domains is that the ordering of the domain elements only reflects the set of reals they approximate. Thus we call this an *extensional* way of representing reals via finite approximations.

We are now ready to define the full hierarchy of domains interpreting functionals of finite types over the reals. Replacing $\mathbb{N}_\perp$ with $R(0)$ and $\mathbb{N}$ with $\mathbb{R}$ we may copy the definition of the $Ct(\sigma)$-hierarchy defining $R(\sigma)$ in the category of algebraic domains for each finite type σ, the set $H_R(\sigma)$ of hereditarily total elements of type σ, the equivalence $\approx^R_\sigma$ on $H_R(\sigma)$ and the extensional collapse hierarchy $\{Ct_{\mathbb{R}}(\sigma)\}_{\sigma \text{ type}}$ with the corresponding identification functions ρ^R_σ. Details may be found e.g. in Normann [23].

It is easy to see that each $Ct_{\mathbb{R}}(\sigma)$ is organized into a topological vector-space in a natural way. By induction on the type, we also see that each compact p in $R(\sigma)$ will determine a closed, convex subset V^p_σ of $Ct_R(\sigma)$, the set of objects with a representative in $H_R(\sigma)$ extending p. We will use the convexity of V^p_σ in the following way:

Lemma 14 *Let $a_1, \ldots, a_n$ be elements in V_σ^p and let μ be a probability distribution on $\{1, \ldots, n\}$. Then*

$$\sum_{i=1}^{n} \mu(i) a_i \in V_\sigma^p.$$

We are now going to discuss the density theorem, originally due to Normann [23] but with an alternative proof in DeJaeger [5]:

Theorem 15 *Let p be a compact in $R(\sigma)$. Then there is an element $a \in H_{\mathbb{R}}(\sigma)$ extending p.*

Since we have not given any detailed definitions, we will not give a detailed proof. We will however indicate how a proof goes, and our approach here is closer to DeJaeger's proof than to our original argument.

Definition 16 By recursion on the type σ we define a finitary type-structure $\{X_n(\sigma)\}_{\sigma \text{ type}}$ for each $n \in \mathbb{N}$ as follows:

$$X_n(0) = \{\tfrac{k}{n!} \mid -(n+1)! \leq k \leq (n+1)!\}.$$

$$X_n(\tau \to \delta) = \{h \mid h : X_n(\tau) \to X_n(\delta)\}.$$

We are now, by simultaneous recursion for each n, going to define an embedding $\nu_{n,\sigma} : X_n(\sigma) \to H_R(\sigma)$ and for each $a \in H_R(\sigma)$ a probability distribution $\mu_{n,\sigma}(a)$ on $X_n(\sigma)$. The prime objects are the embeddings $\nu_{n,\sigma}$ and the probability distributions will replace projections that are used for similar purposes in the discrete case ($\mathbb{N}$ instead of $\mathbb{R}$).

Definition 17

1. Let $\sigma = 0$.

   If $x \notin (-(n+1), n+1)$, we let $\mu(\frac{k}{n!}) = 1$ if $\frac{k}{n!}$ is the object in $X_n(0)$ closest to x, otherwise we let $\mu(\frac{k}{n!}) = 0$.

   If $x \in (-(n+1), n+1)$ there are unique $k \in \mathbb{Z}$ and $y \in [0,1)$ such that $x = \frac{k}{n!} + \frac{1-y}{n!}$.
   Let $\mu_{n,0}(\frac{k}{n!}) = y$, $\mu_{n,0}(\frac{k+1}{n!}) = 1 - y$ and $\mu_{n,0}(\frac{l}{n!}) = 0$ for all $l \neq k, k+1$.

2. Let $\sigma = \tau \to \delta$

Let $h \in X_n(\tau \to \delta)$, $a \in H(\tau)$.
Let
$$\nu_{n,\sigma}(h)(a) = \sum_{c \in X_n(\tau)} \mu_{n,\tau}(a)(c) \cdot \nu_{n,\delta}(h(c)).$$
Now let $f \in H(\tau \to \delta)$ and $h \in X_n(\tau \to \delta)$.
Let
$$\mu_{n,\sigma}(f)(h) = \prod_{c \in X_n(\tau)} \mu_{n,\delta}(f(\nu_{n,\tau}(c)))(h(c)).$$

We observe that at base type we are constructing explicit probability distributions, while at higher types we are using finite products of probability distributions, which again will be probability distributions. Thus $\mu_{n,\sigma}(a)$ is a probability distribution for all n, σ and $a \in H(\sigma)$. We also observe that since all types are topological vector spaces, we may view products with scalars and sums as partial continuous operations on the underlying domains, and thus our constructions are sound and continuous over the underlying domains.

In an embedding-projection pair it is important that

$$\mathit{projection} \circ \mathit{embedding} = \mathit{identity}.$$

In this setting, we formulate the similar phenomenon as

Lemma 18 *For all types σ and each $h \in X_n(\sigma)$ we have that*

$$\mu_{n,\sigma}(\nu_{n,\sigma}(h))(h) = 1.$$

The proof is by induction on σ using a tedious but in principle simple calculation at the induction step.

The density theorem will follow from

Lemma 19 *Let $p \in R(\sigma)$ be compact. Then there is a number n_p such that*

a) For all $n \geq n_p$ there is a $b \in X_n(\sigma)$ such that p is consistent with $\nu_{n,\sigma}(b)$.

b) If $a \in H_R(\sigma)$ and p is consistent with a then

$$\mu_{n,\sigma}(a)(\{b \in X_n(\sigma) \mid p \text{ is consistent with } \nu_{n,\sigma}(b)\}) = 1.$$

c) If $a \in H_R(\sigma)$ then a is pre-maximal *in the sense that a has a unique maximal extension.*

Proof
We use induction on σ.
Let $\sigma = 0$
Let $p = [q, r]$ where q and r are rational numbers. Choose n_p such that $|q|$, $|r|$ and the denominators of q and r are bounded by n_p. This will do the trick for a) and b). c) is trivial.
Let $\sigma = \delta \to \tau$.
c) follows by the density theorem for τ and c for δ.
Let p be the least upper bound of the step functions $\{f_{q_i,r_i}\}_{i \in I}$ where I is finite. Let

$$n_p = max\{n_{\bigsqcup_{i \in K} q_i}, n_{\bigsqcup_{i \in K} r_i} \mid K \subseteq I \wedge \{q_i \mid i \in K\} \text{ is consistent}\}.$$

We now prove a) and b), assuming both a) and b) for τ and δ.
Proof of a):
By c) for τ we have that q_i is consistent with $a \in H_R(\tau)$ if and only if q_i is in the maximal extension of a, so the set of q_i's consistent with a is itself consistent. Let $n \geq n_p$ and by the induction hypothesis let $h \in X_n(\sigma)$ be such that for all $b \in X_n(\tau)$, r_i is consistent with $\nu_{n,\delta}(h(b))$ whenever q_i is consistent with $\nu_{n,\tau}(b)$.
So let $a \in H_R(\tau)$ and let $K = \{i \in I \mid p_i \text{ is consistent with } a\}$. By definition,

$$\nu_{n,\sigma}(h)(a) = \sum_{c \in X_n(\tau)} \mu_{n,\tau}(a)(c) \cdot \nu_{n,\delta}(h(c)).$$

By the induction hypothesis, we only have to consider those $c \in X_n(\tau)$ where $\nu_{n,\tau}(c)$ is consistent with $\bigcup\{q_i \mid i \in K\}$, and for those c, $\nu_{n,\delta}(h(c))$ will be in $V_\delta^{\bigsqcup_{i \in K} r_i}$. By the convexity of this set we see that $\nu_{n,\sigma}(h)(a) \in V_\delta^{\bigsqcup_{i \in K} r_i}$. This verifies a.
Proof of b):
Let $n \geq n_p$, $h \in X_n(\sigma)$, $f \in H_R(\sigma)$ such that f is consistent with p, but $\nu_{n,\sigma}(h)$ is not consistent with p. We verify b) by showing that $\mu_{n,\sigma}(f)(h) = 0$. $\mu_{n,\sigma}(f)(h)$ is defined to be a product, and it is sufficient to prove that one of the factors is zero.

Since $\nu_{n,\sigma}(h)$ is not consistent with p, there must be an $i \in I$ and $a \in V_\tau^{q_i}$ such that $\nu_{n,\sigma}(h)(a)$ is inconsistent with r_i.
By the argument of a), we can find $b \in X_n(\tau)$ such that $\nu_{n,\tau}(b)$ is consistent with q_i while $\nu_{n,\delta}(h(b))$ is inconsistent with r_i.
Since f is consistent with p, we have that $f(\nu_{n,\tau}(b))$ is consistent with r_i and

$f(\nu_{n,\tau}(b)) \in H_R(\delta)$. Then it follows by b) for δ that $\mu_{n,\delta}(f(\nu_{n,\tau}(b)))(h(b)) = 0$.
Since this is one of the factors in $\mu_{n,\sigma}(f)(h)$, we are through.

If we consider a typed hierarchy with both connected and discrete base types, e.g. domain representations of $\mathbb{R}$ and $\mathbb{N}$, the density-theorem fails for trivial reasons. We may find a compact approximation to a partial continuous function $f : \mathbb{R} \to \mathbb{N}$ that is not constant, but all total such functions will be constant. In Normann [23] we define an alternative hierarchy of domains with totality satisfying the density theorem where the quotient spaces of hereditarily total functionals will be the same as with the traditional Scott domain approach. The drawback is that we will leave some interesting partial functionals out of this model.

4.2 The intensional hierarchy

In our first construction, we started with the closed interval domain where the elements represent approximations to the reals. This is called the *extensional approach* and may be viewed as improvements of Dedekind Cuts. Alternative ways of representing the reals are via data-streams, or more mathematically, via infinite words in some alphabet. We will consider one example, the so called *negative binary digit representation*.

Definition 20 Let $\alpha \in \mathbb{Z} \times \{-1, 0, 1\}^{\mathbb{N}^+}$ where $\mathbb{N}^+ = \{1, 2, 3, \cdots\}$. We consider α as a function defined on $\mathbb{N}$.
Let the real $r(\alpha)$ *represented by* α be

$$r(\alpha) = \alpha(0) + \sum_{n \in \mathbb{N}^+} \alpha(n) \cdot 2^{-n}.$$

Let REP_0 be the set of such α's

We may consider REP_0 as the total objects in the domain $R^I(0)$ (I for intensional) of finite or infinite sequences $ab_1b_2\cdots$ where $a \in \mathbb{Z}$ and each $b_i \in \{-1, 0, 1\}$. This domain is again homeomorphic to a sub-domain of $D(\iota \to \iota)$, the domain of partial continuous functions from $\mathbb{N}_\perp$ to $\mathbb{N}_\perp$ such that both the embedding and the projection are total. This further suggests that based on this model, we may use this representation, its higher type semantics and a PCF-like language to give a semantics to typed computations involving reals. This idea is discussed further in DiGianantonio [6, 7, 8] and Simpson [32].

Let us define the hereditarily total functionals REP_σ with the equivalence relation $\sim_\sigma$ for each type σ:

Definition 21 We let $R^I(0)$ be as above, and define $R^I(\sigma)$ for all finite types σ over the base type 0 in the category of algebraic domains. By recursion on σ we define the set $REP_\sigma \subseteq R(\sigma)$ and the binary relation $\sim_\sigma$ on REP_σ

$\sigma = 0$:
If f and g are in REP_0 we let $f \sim_0 g$ if they represent the same real.

$\sigma = \tau \to \delta$:
Let $\phi \in R^I(\sigma)$. We let $\phi \in REP_\sigma$ if

i) $a \in REP_\tau \Rightarrow \phi(a) \in REP_\delta$ for all $a \in R(\tau)$

ii) $a \sim_\tau b \Rightarrow \phi(a) \sim_\delta \phi(b)$ whenever $a, b \in REP_\delta$.

If ϕ and ψ are in REP_σ we let $\phi \sim_\sigma \psi$ if $\phi(a) \sim_\delta \psi(b)$ whenever $a \sim_\tau b$.

Except for the base type, REP_σ will not be dense in $R^I(\sigma)$. There will be inconsistent compacts in $R^I(0)$ that can be extended to inconsistent, but equivalent total objects. This can be used to construct two consistent step-functions in $R^I(0 \to 0)$ that cannot be extended to any element in $REP_{0\to 0}$. It has actually been left open if the set of compacts that can be extended to a total object is decidable, but this is probably because no one tried hard to prove it. In Normann [24] an alternative hierarchy of effective domains with totality, satisfying density and leading to the same coefficient spaces of total objects, was constructed. The idea is that we introduce an extra relation representing 'can be extended to equivalent total objects' on the compacts, and we only accept the compacts in a function space respecting this relation.

4.3 The coincidence problem

We have considered two hierarchies of total continuous functionals of finite types over the reals, both constructed as quotients of hereditarily total objects in a hierarchy of algebraic domains, one based on the closed interval domain and one on the negative binary representation of reals. There is a third approach, originating from Weihrauch's TTE [35]. In this approach one uses admissible representations at each type. The *coincidence problem* is if these approaches coincide, i.e. if the typed structures defined via these approaches are the same.

Due to characterizations of the extensional hierarchy (Normann [23]) and the TTE-based hierarchy (Schröder [30]) as the one obtained from the reals in the category of limit spaces, these two approaches are known to be equivalent. Thus the real coincidence problem is if the domain theoretical approaches based on extensional and intensional representations of the reals coincide. This problem is of course more of a foundational than of a practical nature. In its nontechnical form the question is if the choice of representation of reals as data will have any influence on what is considered to be a continuous, total functional of higher type.

The problem was first addressed by Bauer, Escardó and Simpson [3], and they obtained some partial results. Normann [26] showed that the coincidence problem is equivalent to a topological problem about the Kleene-Kreisel functionals:

> *Is the topology generated by all continuous functions $\phi : Ct(n) \to \mathbb{R}$ zero dimensional for all n?*

Theorem 22 (see below) is used to establish this equivalence.

5 Density and probability

In $Ct(\sigma)$, two distinct objects of the same type can be separated by a clopen set, i.e. a set that i both closed and open. This is of course useful in algorithm design, definitions by cases are handy. Each space $Ct_{\mathbb{R}}(\sigma)$ is path connected [23], so definitions by cases have to fail for some inputs. It has turned out that in some cases, elementary methods from probability theory can be combined with the density theorem to overcome this lack of separation. The proof of the density theorem itself is one such example. We will mention two more results here.

5.1 The embedding theorem

Theorem 22 *Let $Ct_{\mathbb{R}}(\sigma)$ be the quotient space of the hereditarily total functionals over $\mathbb{R}$ in the extensional hierarchy.*
For each type σ there is a continuous map $\pi_\sigma : Ct(\sigma) \to Ct_{\mathbb{R}}(\sigma)$ such that π_0 is the standard embedding of $\mathbb{N}$ into $\mathbb{R}$ and for each type $\sigma = \tau \to \delta$, each $\phi \in Ct(\sigma)$ and $a \in Ct(\tau)$ we have that

$$\pi_\sigma(\phi)(\pi_\tau(a)) = \pi_\delta(\phi).$$

This theorem, and an analogue result for the hierarchy based on the intensional representation, is proved in [24]

We will not give the proof here, but discuss where the density theorem is used. Ideally we would like to replace the (possible) use of a (nonexistent) continuous projection from $Ct_{\mathbb{R}}(\sigma)$ to $Ct(\sigma)$ with a continuous function μ_σ defined on $Ct_{\mathbb{R}}(\sigma)$ where $\mu(\phi)$ then would be a probability measure on $Ct(\sigma)$. Then, if $\phi \in Ct(\tau \to \delta)$ and $b \in Ct_{\mathbb{R}}(\tau)$ we could define

$$\pi_{\tau\to\delta}(\phi)(b) = \int_{a \in Ct(\tau)} \phi(a) d\mu_\tau(b).$$

Unfortunately, the complexity of these spaces makes it impossible to define decent probability measures on them. The alternative is to use approximations via probability distributions on the sets $Z_n(\sigma) = \{(a)_n \mid a \in H(\sigma)\}$ and to use that every total object in $Ct(\tau)$ can be approximated in a uniform way by these finitary elements.

5.2 Representation theorems

It is well known that topological spaces can be represented as quotient spaces over domains, old and contemporary literature is too vast for us to be specific about it. It is also well known that Polish spaces will be homeomorphic to G_δ-subsets of $[0,1]^\omega$. In Normann [25] we consider finite types where arbitrary Polish spaces may be used as base types, interpreted in the category of limit spaces. We then show that the hierarchy

$$\{Ct_{\mathbb{R}}(\sigma)\}_{\sigma \text{ type}}$$

is adequate for interpreting all these spaces of objects of higher types, in the sense that each such space will be homeomorphic to a subspace of some $Ct_{\mathbb{R}}(\sigma)$.

In finding these subspaces we use a concept of *representation* developed from an intermediate step in the proof of Theorem 1:

Definition 23 Let $A \subseteq Ct_R(\sigma)$. A *representation of A in $Ct_R(\pi)$* is a continuous map

$$\phi : R(\sigma) \to R(\pi)$$

such that for all $a \in REP_\sigma$ we have that

$$\phi(a) \in REP_\pi \Leftrightarrow \rho^R_\sigma(a) \in A.$$

We show that each space of finite type over Polish spaces is homeomorphic to a subspace of some $Ct_R(\sigma)$ with a representation. This is proved by induction on the type. We give a highly incomplete sketch of the induction step, omitting that the set we construct will be representable:

- By the density theorem for $Ct_R(\pi)$ there is a total map $h : R(\pi) \to (\mathbb{N}_\perp \to R(0))$ such that h identifies exactly the consistent elements.

- Let $A \subseteq Ct_R(\tau)$ and $B \subseteq_R (\delta)$ and let $h : R(\tau) \to R(\pi)$ be a representation of A. Let $f : A \to B$ be continuous. Uniformly (but non-trivial) in f there is a sequence $\{f_n\}_{n \in \mathbb{N}}$ of continuous functions from $Ct_R(\tau)$ to $Ct(\delta)$ such that f will be the limit of the restrictions of f_n to A. By linear interpolation, we define f_x for each non-negative real x.

- If $f : A \to B$, $a \in Ct(\tau)$ and $b \in Ct(\pi)$, we let $F(f)(a, b) = f(a)$ if $h(\phi(a)) = h(b)$ while it is $f_x(a)$ for some x continuously witnessing that $h(\phi(a)) \neq h(b)$ otherwise.

- We may recover f from $F(f)$ by $f(a) = F(a, \phi(a))$. Thus the set $\{F(f) \mid f \in A \to B\}$ will be homeomorphic to $A \to B$.

References

[1] Abramsky, S. and Jung, A. , *Domain Theory*, in S. Abramsky, D.M. Gabbay and T.S.E. Maibaum (eds.) *Handbook of Logic in Computer Science*, volume 3, Clarendon Press (1994).

[2] Amadio, R. M. and Curien, P. -L. , *Domains and Lambda-Calculi*, Cambridge University Press (1998).

[3] Bauer, A., Escardó, M.H. and Simpson, A., *Comparing Functional Paradigms for Exact Real-number Computation*, in Proceedings ICALP 2002, Springer LNCS 2380, pp. 488-500 (2002).

[4] Berger, U., *Totale Objekte und Mengen in der Bereichtheorie* (in German), Thesis, München (1990).

[5] DeJaeger, F., *Calculabilité sur les réels*, Thesis, Paris VII (2003).

[6] DiGianantonio, P., *A Functional Approach to Computability on Real Numbers*, Thesis, Università di Pisa - Genova - Udine, (1993).

[7] DiGianantonio, P., *Real Number Computability and Domain Theory*, Information and Computation, Vol. 127, 11 - 25 (1996).

[8] DiGianantonio, P., *An abstract data type for real numbers*, Theoretical Computer Science Vol. 221, 295 - 326 (1999).

[9] Ershov, Yu. L.,*Maximal and everywhere defined functionals*, Algebra and Logic 13 , 210-225 (1974).

[10] Gandy, R.O. and Hyland, J.M.E., *Computable and recursively countable functions of higher type*, Logic Colloquium '76, North-Holland (1977).

[11] Hyland, J.M.E., *Recursion on the countable functionals*, Dissertation, Oxford (1975).

[12] Kleene, S. C. , *Recursive functionals and quantifiers of finite types I*, Trans. Amer. Math. Soc. 91, 1 - 52 (1959).

[13] Kleene, S. C., *Countable functionals*, in A. Heyting (ed.) Constructivity in Mathematics, North-Holland , 81-100 (1959).

[14] Kreisel, G. , *Interpretation of analysis by means of functionals of finite type*, in A. Heyting (ed.) Constructivity in Mathematics, North-Holland, 101-128 (1959).

[15] Longley, J.R., *On the ubiquity of certain total type structures*, In preparation.

[16] Longo, G. and Moggi, E., *The hereditary partial effective functionals and recursion theory in higher types*, Jour. Symb. Logic 49 , 1319-1332 (1984).

[17] Milner, R., *Fully abstract models for typed λ-calculi*, Theoretical Computer Science 4, 1 - 22 (1977).

[18] Moldestad, J., *Computations in Higher Types*, Springer Lecture Notes in Mathematics No. 574, Springer Verlag (1977).

[19] Normann, D., *The continuous functionals; computations, recursions and degrees*, Annals of Mathematical Logic 21, 1-26 (1981).

[20] Normann, D., *Computing with functionals - computability theory or computer science?*, to appear in Bulletin of Symbolic Logic (2006).

[21] Normann, D., *Computability over the partial continuous functionals*, The Journal of Symbolic Logic 65 , 1133-1142 (2000).

[22] Normann, D., *The Cook-Berger Problem - A Guide to the Solution*, Electronical Notes in Theoretical Computer Science 35 (2000)

[23] Normann, D., *The continuous functionals of finite types over the reals*, in K. Keimel, G.Q. Zhang, Y. Liu and Y. Chen (eds.) Domains and processes, 103 - 124, Kluwer Academic Publishers, (2001).

[24] Normann, D., *Hierarchies of Total Functionals over the Reals*, Theoretical Computer Science 316 pp. 137 - 151 (2004).

[25] Normann, D., *Definability and reducibility in higher types over the reals*, to appear in the proceedings og Logic Colloquium '03.

[26] Normann, D., *Comparing hierarchies of total functionals*, Logical Methods in Computer Science 1(2) (2005).

[27] Platek, R. A., *Foundations of Recursion Theory*, Thesis, Stanford University (1966).

[28] Plotkin, G., *LCF considered as a programming language*, Theoretical Computer Science 5, 223 - 255 (1977).

[29] Plotkin, G., *Full abstraction, totality and PCF*, Math. Struct. in Comp. Science , vol. 11, pp. 1-20 (1999).

[30] Schröder, M., *Admissible representations of limit spaces*, in J. Blanck, V. Brattka, P. Hertling and K. Weihrauch (eds.), *Computability and complexity in Analysis*, volume 237, Informatik Berichte, pp. 369 - 388, (2000).

[31] Scott, D., *A type-theoretical alternative to ISWIM, CUCH, OWHY*, Unpublished notes, Oxford (1969).
Printed with suplementary comments in Theoretical Computer Science 121, 411 - 440 (1993).

[32] Simpson, A., *Lazy Functional Algorithms for Exact Real Functionals*, Mathematical Foundations of Computer Science 1998, Springer LNCS 1450, 456-464, (1998).

[33] Stoltenberg-Hansen, V. , Lindström, I. and Griffor, E.R. , *Mathematical Theory of Domains*, Cambridge Tracts in Theor. Comp. Sci. 22, Cambridge University Press (1994).

[34] Tait, W.W., *Continuity properties of partial recursive functionals of finite type*, unpublished notes (1958).

[35] Weihrauch, K., *Computable analysis*, Texts in Theoretical Computer Science, Springer Verlag, Berlin, (2000).