

On the Cantor-Bendixson rank of a set that is searchable in Gödel's T

Dag Normann

The University of Oslo, Department of Mathematics
P.O.Box 1053 Blindern, Oslo, Norway
dnormann@math.uio.no
<http://folk.uio.no/dnormann/>

Abstract. We prove that if a closed subset X of the Baire space is searchable in the sense of Escardó [2], and by a functional definable in Gödel's T, then X is countable, and the Cantor-Bendixson rank of X is bounded by the ordinal ε_0 . To this end we introduce *evaluation trees*, well founded decorated trees that induce operators from the full set-theoretical class $\mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}$ to $\mathbb{N}^{\mathbb{N}}$. We prove that when a search operator for a set X is induced from an evaluation tree T , then the Cantor-Bendixson rank of X is bounded by the Kleene-Brouwer order type of T . Further we use a theorem due to Howard [8], estimating the ordinal complexity of the reduction tree of a term in system T, to show that all functionals of type $(\mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}^{\mathbb{N}}$ definable in T can be computed using an evaluation tree of ordinal rank below ε_0 .

Keywords: Searchable, Gödel's T, Cantor-Bendixson rank

1. Introduction

In this paper, we will give partial answers to the following question:

Let X be a set. Assuming that there is a procedure Φ that to a predicate P on X decides if P is empty on X or not, what can we say about X ?

The question is not very precise, and in order to turn it into a mathematical problem we have to specify

1. What kind of sets X will we consider?
2. What do we mean with a "procedure"?
3. Will we consider all predicates or just a natural collection of them?

If X is a finite set $\{x_1, \dots, x_n\}$, the answer will be that as long as we allow for using oracle calls $P(x_i)$ in our model for procedures, whatever this model of procedures might be, we can decide if P is empty on X or not.

If X is infinite, a procedure will essentially have to take a function

$$P : X \rightarrow \{0, 1\}$$

as an input, and then give one of the two values 0 or 1 as the output, so the procedure will itself be an object of type level 2 over X . This suggests that various models for computing with functionals will be of interest when we make our problem precise. When nothing else is said, we will let X be a compact subset of *Baire space* $\mathbb{N}^{\mathbb{N}}$. X is then a topological space, and we will either let P range over all functions from X to $\{0, 1\}$ or, as in Section 4, the continuous ones.

Escardó [2] defined the concepts of *searchable* and *exhaustible* sets originally for continuous models of LCF [15] (or equivalently PCF [14]). We will discuss this in more detail below. Later he also considered these concepts as induced by the computational model of Gödel's T [7]. Our main concern will be with searchable sets modulo T, but we will also consider an intermediate model, Kleene's S1 - S9 [9]. We will let T and S1 - S9 have their standard interpretations over the full set-theoretical type structure.

For a detailed introduction to higher order computability, one may consult the forthcoming [12]. Here we will give brief introductions to PCF, to Kleene's S1 - S9 and to the calculus we are mainly interested in, Gödel's T.

1.1. Background

The Kleene schemes S1 - S9 and the corresponding concept of computability was originally defined for the full type-structure of pure types, and it is a calculus that gives meaning to expressions like

$$\{e\}(\Phi_1, \dots, \Phi_n) = a.$$

Here e may be viewed as a Gödel number of an algorithm, Φ_1, \dots, Φ_n will range over the functionals of some pure, finite types (coded into e) in the full set-theoretical type structure and a is a non-negative integer. S1 - S9 are nine clauses in a grand strictly positive inductive definition of the set of sequences $(e, \Phi_1, \dots, \Phi_n, a)$ such that $\{e\}(\Phi_1, \dots, \Phi_n) = a$. However, Kleene's definition will also make sense in other reasonably closed type structures of total or partial functionals, and the strength of this model of computability will strongly depend on the typed structure it is interpreted in.

The *Kleene-Kreisel continuous functionals* were essentially and independently defined in Kleene [10] and Kreisel [11]. There are numerous characterizations of this typed structure. The one being most useful for our purpose is via the partial continuous functionals. We define the typed structure of partial continuous functionals over the integers within the category of Scott-Ershov domains, as initiated in Scott [15]. We then isolate the hereditarily total objects in this structure with the canonical equivalence relation of extensional equivalence. The equivalence classes will correspond to the Kleene-Kreisel continuous functionals.

In his seminal, but for a long time, unpublished note [15], Scott introduced the formal calculus LCF, which was later transformed into a programming language PCF by Plotkin [14]. PCF is a formal calculus without reference to any particular model, and may be viewed as a typed λ -calculus with least fixed point operators of all types. The partial continuous functionals will serve as a model for PCF, and in this model PCF and S1 - S9 have the same computational power. An element of the corresponding equivalence class will be a *representative* of a functional. We say that a Kleene-Kreisel functional is PCF-definable if there is a representative of it among the partial continuous functionals that is the interpretation of a PCF-term. Normann [13] proved that if a Kleene-Kreisel functional has a representative that can be approximated by a computably enumerable set of finitary elements of the Scott-Ershov domain, then it is PCF-definable. This also means that there is a representative that is S1 - S9 - computable in the model of partially computable functionals. This is seemingly in contrast to the theorem in Tait [17], where it is proved that the *fan functional* (see discussion later) is not Kleene computable, but fortunately, only seemingly.

Martín Escardó [2] introduced *searchable* and *exhaustible* sets in the framework of PCF, the partial continuous functionals and the Kleene-Kreisel continuous functionals. A set X of Kleene-Kreisel functionals is *searchable* in the original sense of Escardó if there is a PCF-definable operator ε such that whenever P is a continuous function mapping X into $\{0, 1\}$ then $\varepsilon(P) \in X$ and

$$\exists x \in X (P(x) = 1) \Leftrightarrow P(\varepsilon(P)) = 1.$$

Escardó proved that all searchable sets are compact and that there are close analogies between being non-empty and compact and being searchable. In particular he proved an effective analogue of the Tychonoff theorem for countable products of searchable sets. He also considered searchable sets of hereditarily total functionals in the Scott-domains, but this will not be of relevance to us.

An important source of inspiration for Escardó's work was a theorem due to Berger [1]:

There is a PCF-definable representative for the fan functional.

The fan functional is a Kleene-Kreisel functional of pure type 3. We will let $C = \{0, 1\}^{\mathbb{N}}$ be our version of *the Cantor set*. C is compact, so every continuous functional F of type 2 will be uniformly continuous on F . This again means that there is a uniform modulus n of continuity, i.e., if $f, g \in C$ agrees on $0, \dots, n-1$, then $F(f) = F(g)$. The fan functional Φ "computes" the least such modulus of continuity. Φ has a computable representative, but Tait [17]

proved that Φ is not computable in Kleene's original calculus based on S1 - S9. The key to the argument is that a Kleene algorithm with an input F of type 2 can only utilize F restricted to a countable set, while $\Phi(F)$ will need to know F on the full Cantor set C .

Berger [1] proved that the fan functional is PCF-definable. According to private information from Martin Hyland, this was known to Robin Gandy, but not published. Simpson [16] used the methods of Berger to prove that there are PCF-programs for Riemann integration and for computing the maximal value of a continuous function on a closed, bounded interval, in the sense of the model for exact computing with reals suggested by Di Gianantonio [4–6].

1.2. Searchability in weaker systems

1.2.1. System T

Our version of T will be like this:

- We consider all finite types over the *base type* 0. We write

$$\sigma = \tau_1, \dots, \tau_n \rightarrow 0$$

for $\sigma = \tau_1 \rightarrow (\dots \rightarrow (\tau_n \rightarrow 0) \dots)$.

- Our term language will be that of typed λ -calculus with extra constants $\hat{0}$ of type 0, successor *suc* of type $0 \rightarrow 0$ and recursors rec_σ of types $0, \sigma, (0, \sigma \rightarrow \sigma) \rightarrow \sigma$ for each type σ . We write N, M etc. for terms.

We will, by recursion, write $n + 1$ for *suc* \hat{n} . The terms \hat{n} are called *numerals*

- The conversion rules will be the conversion rules of typed λ -calculus with α - and β -conversion

$$\alpha: \lambda x^\sigma. M \rightsquigarrow \lambda y. M[x/y]$$

$$\beta: (\lambda x^\sigma. M)N \rightsquigarrow M[x/N]$$

where we also accept conversion of subterms. We of course assume that all substitutions are legal.

We have standard conversion rules for the constants:

$$- \text{rec}_\sigma \hat{0} MN \rightsquigarrow M$$

$$- \text{rec}_\sigma n + 1 MN \rightsquigarrow N \hat{n} (\text{rec}_\sigma \hat{n} MN)$$

We will let \rightsquigarrow be reflexive and transitive. We extend T to T_P by adding a new constant \hat{P} of type $(0 \rightarrow 0) \rightarrow 0$.

We will not add conversion rules for \hat{P} directly, but for each instance P we may consider an infinitary *P-rule*: If M is a term of type $0 \rightarrow 0$ and $M \hat{n} \rightsquigarrow \widehat{f(n)}$ for each $n \in \mathbb{N}$, then $\hat{P}M \rightsquigarrow \widehat{P(f)}$.

We finally say that a term M defines Φ of type $((0 \rightarrow 0) \rightarrow 0) \rightarrow 0$ if for every P and n , $\Phi(P) = n$ if and only if $M \rightsquigarrow \hat{n}$ when we use the *P-rule*. We then say that Φ is *T-definable*. These concepts extend trivially to functionals into $\mathbb{N}^{\mathbb{N}}$.

1.2.2. The conjecture

Escardó [3] found examples of infinite, countable sets that are searchable by operators definable in System T. The least ordinal $> \omega$ closed under addition, multiplication and exponentiation is known as ε_0 . For each ordinal $\alpha < \varepsilon_0$ Escardó found a compact, well ordered subset X_α of the Cantor-space C of order type $\beta \geq \alpha$, such that X is searchable by a functional definable in T. T of course offers a weaker model of computability than PCF, so it is remarkable that for these examples, the search algorithms will even work for all predicates, not just the continuous ones.

Escardó presented these results, later published in [3], at the workshop *Types 2011* in September 2011. There he suggested that if a set X is searchable via an operator definable in T, then X must be countable. He also mentioned the possibility of the Cantor-Bendixson rank of X to be bounded by the ordinal ε_0 and left this as an open problem. At a Dagstuhl workshop in October 2011 he presented his results again, this time actually conjecturing that ε_0 is an

upper bound for the topological complexity of T-searchable sets. The main result of this paper is that this conjecture is true.¹

1.2.3. This paper

In Sections 2 and 3 we prove that if X is searchable in T with respect to all predicates on X , then the Cantor-Bendixson rank of X is bounded by ε_0 . In Section 2 we introduce the concept of an *evaluation tree* and, in the case where there is a search operator for all total predicates P on X computable using an evaluation tree, we show that the ordinal rank of the evaluation tree induces an ordinal bound on the Cantor-Bendixson rank of X .

In Section 3 we use a result by Howard [8] and show that when an operator of type 3 is T-definable, then it is computable via an evaluation tree of rank $< \varepsilon_0$.

In Section 4 we strengthen the result from Section 2 by only assuming that the search algorithm of the evaluation tree works for predicates that are continuous on X . This section actually makes Section 2 obsolete, but we believe that the basic intuition is easier to see from the exposition in Section 2 than it would have been if the sharpest result had been proved directly.

It is also reasonable to ask when a set X may be searchable by a functional ε definable in Kleene's S1 - S9 interpreted in the classical way. If we assume that $X \subseteq \mathbb{N}^{\mathbb{N}}$ is closed and searchable with respect to all predicates that are continuous on X in this way, results of Escardó [2] give us that X is compact.

Employing the method of Tait [17] we can show that while deciding that the empty predicate O is empty on X , the computation tree of $\varepsilon(O)$ must make an oracle call $O(f)?$ for each $f \in X$. Thus X has to be countable, and there will actually be an effective enumeration

$$X = \{f_n \mid n \in \mathbb{N}\}$$

of the elements in X , an enumeration extractable from the computable, infinite and well founded computation tree of $\varepsilon(O)$. We say that $X \subseteq \mathbb{N}^{\mathbb{N}}$ is *computably enumerable* when there is a computable enumeration like this. It is an open problem whether it suffices that a set X is compact and computably enumerable for X to be searchable in the original sense of Kleene computability over the hereditarily total functionals.

For the sake of completeness, we will define the Cantor-Bendixson rank in the next section. Let us consider here the set A of pairs (h, g) from $\mathbb{N}^{\mathbb{N}}$ where g codes a total ordering on a set $\text{dom}(g)$, h maps \mathbb{N} onto $\text{dom}(g)$, and for each $n \in \mathbb{N}$ we have that $h(n) \in \text{dom}(g)$ has a rank in $\text{dom}(g)$ that equals the Cantor-Bendixson rank of f_n in X . A will be an arithmetical set, and when $(h, g) \in A$, g will code an ordinal exceeding the Cantor-Bendixson rank of X . Thus there is a Σ_1^1 -set of codes for well orderings where each well ordering bounds the Cantor-Bendixson rank of X . By standard descriptive set theory, this shows that the Cantor-Bendixson rank of X is a computable ordinal. Since there are no original technical aspects of this argument, we omit further details.

2. Sets that are searchable using evaluation trees

2.1. Preliminaries

It is well known that a countable, compact Hausdorff-space X is homeomorphic to the ordering topology of a countable successor ordinal. A familiar proof will use the *Cantor-Bendixson rank*, CB-rank, of the elements of X defined as follows:

- If x is isolated in X , then $\text{CB-rank}(x) = 0$.
- If $\alpha > 0$, we do not have that $\text{CB-rank}(x) < \alpha$ but there is a neighbourhood B of x such that $\text{CB-rank}(y) < \alpha$ for all $y \in B$ that are different from x , we let $\text{CB-rank}(x) = \alpha$.

¹The author had intense discussions with Escardó on this conjecture during the Dagstuhl meeting. In May-June of 2012 we met again, this time at the Newton Institute in Cambridge, UK in connection with the program SAS, and we continued our discussions with some further progress.

The technical breakthrough came during the autumn of 2012 and was due to the author. We decided, after some discussion, not to write a joint paper, but the influence of Escardó is important, both as a supplier of motivation and as a discussion partner.

If X is a compact, countable Hausdorff space, each $x \in X$ will have a CB-rank. By sequential compactness, there will be an element with maximal CB-rank, and there can only be finitely many of those.

We will make use of the following observation. The proof will use sequential compactness, and the observation may serve as a student exercise in general topology:

Lemma 2.1. *Let X and Y be countable, compact Hausdorff-spaces, $\phi : X \rightarrow Y$ be continuous and surjective. For each $y \in Y$ there is an $x \in X$ such that*

$$\phi(x) = y \text{ and } CB\text{-rank}(x) \geq CB\text{-rank}(y).$$

In this paper, a *tree* will be a set of finite sequences from \mathbb{N} , closed under sub-sequences. The empty sequence e will be the *root* of the tree. The elements of the tree are called *nodes*.

A *subtree* of a tree T will be a subset that is also a tree. If $s \in T$, the *child tree* of s is the set of sequences t such that the concatenation $st \in T$. In the notation for concatenation, we will identify a number with the corresponding sequence of length 1.

A *branch* in a tree will be the set of subsequences of some node in the tree. We will not assume that a branch necessarily ends in a leaf node, but of course it may do so.

Let $<_l$ be the following partial variant of the lexicographical ordering on the set of finite sequences:

$$a_0 \cdots a_{n-1} <_l b_0 \cdots b_{m-1} \text{ if there is an } i < \min\{n, m\} \text{ such that } a_i < b_i \text{ and } a_j = b_j \text{ for all } j < i.$$

Let $s < t$ if s is a proper subsequence of t and let $s \preceq t$ if s is just a subsequence of t . We sometimes write “left of” for $<_l$.

We will use the *Kleene-Brouwer ordering* $<_{KB}$ of the set of finite sequences from \mathbb{N} defined as follows: $s <_{KB} t$ if $t < s$ or if $s <_l t$. We let \leq_{KB} be the reflexive counterpart. The Kleene-Brouwer ordering is a total ordering. The key property is that a tree T is well founded if and only if the Kleene-Brouwer ordering well-orders T . A well founded tree will have ordinal rank $< \varepsilon_0$ if and only if the Kleene-Brouwer ordering of the tree has length $< \varepsilon_0$.

Our trees will for the most be *decorated*, meaning that we attach some mathematical object to each node in the tree.

2.2. Evaluation trees

Our tool for investigating the Cantor-Bendixson rank of sets searchable in weak systems will be *evaluation trees*:

Definition 2.2. An *evaluation tree* will be a well founded, decorated tree T of finite sequences s from \mathbb{N} satisfying

- A node s in T is either a leaf node in T , or a full branching node, i.e. the n 'th successor $sn \in T$ for all $n \in \mathbb{N}$.
- If s is a leaf node, then it is decorated with a function $f_s \in \mathbb{N}^{\mathbb{N}}$.
- If s is a branching node, it will be decorated with a continuous functional $F_s : \mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}^{\mathbb{N}}$.

Definition 2.3. If T is an evaluation tree as above, $P : \mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}$ and $s \in T$, we let $[[s]]_T(P) \in \mathbb{N}^{\mathbb{N}}$ be defined by transfinite recursion as follows:

- If s is a leaf node then $[[s]]_T(P) = f_s$.
- If s is a branching node, we let $[[s]]_T(P) = F_s(\lambda n. P(f_{sn}))$.

We will normally drop the lower index T when it is clear from the context.

2.3. The main theorem

This section is devoted to the proof our main technical result:

Theorem 2.4. *Let $X \subseteq \mathbb{N}^{\mathbb{N}}$ be closed. Let T be an evaluation tree such that*

$$P \mapsto [[e]]_T(P)$$

is a search operator for X (where e still denotes the empty sequence).

Then X is countable, and the Cantor-Bendixson rank of the elements of X are bounded by the rank of the Kleene-Brouwer ordering of T .

We will split the proof into several lemmas, where we assume that X and T are as in the theorem. Our intuition is that in order to define a search operator, T must implicitly search through X for an f satisfying P , and, when at a node in T finds an f satisfying P , this information is brought to the top. In this section we turn this intuition, restricted to one-point predicates, into the definition of being a relevant node. Our construction will be non-constructive, because we cannot tell in an effective way which are the nodes in T actually contributing to $[[e]]_T$ being a search operator and which nodes are just dummy ones.

We will let O denote the everywhere zero function on $\mathbb{N}^{\mathbb{N}}$ and for $f \in X$ we will let

$$P_f(g) = \begin{cases} 1 & \text{if } f = g \\ 0 & \text{if } f \neq g \end{cases}$$

Note that we will have that $[[e]](P_f) = f$ for all $f \in X$.

Definition 2.5. A node $s \in T$ is *relevant* for $f \in X$ if $[[t]](P_f) = f$ for all $t \preceq s$.

A node s in T is *critical* for $f \in X$ if it is \preceq -maximal among the set of nodes relevant for f .

The set of nodes s in T that are relevant for $f \in X$ will form a subtree T_f , and the leaf nodes of T_f will be the nodes that are critical for f .

Since e is relevant for all $f \in X$, there will be at least one critical node for each $f \in X$.

Lemma 2.6. Let $f \in X$ and let s be critical for f .

Then $f = [[s]](O)$.

Proof

If s is a leaf node, we have that $[[s]](O) = f_s = [[s]](P_f) = f$, where the last equation follows from the assumption of the lemma.

If s is a branching node, we have that $[[sn]](P_f) \neq f$ for each $n \in \mathbb{N}$, and thus

$$[[s]](O) = F_s(\lambda n. O([[sn]](O))) = F_s(\lambda n. P_f([[sn]](P_f))) = [[s]](P_f) = f.$$

This ends the proof.

This shows in particular that X must be countable.

For each $s \in T$, we now let $R(s)$ be the set of $f \in X$ such that s is on the *leftmost* branch in T_f .²

We let T^* be the set of nodes s for which $R(s) \neq \emptyset$. T^* will be a subtree of T , and the Kleene-Brouwer ordering on T^* will be a sub-ordering of the Kleene-Brouwer ordering of T . Our aim will be to construct a surjective map ϕ from T^* onto X that is continuous with respect to the order topology of the Kleene-Brouwer ordering of T^* . Since the order topology of T^* will be compact, X is compact. We then use Lemma 2.1 to obtain the theorem.

Observe that if $s \preceq t \in T^*$, then $R(t) \subseteq R(s)$.

Lemma 2.7. If $t \in T^*$, $tn_i \in T^*$ for each $i \in \mathbb{N}$, $\lim_{i \rightarrow \infty} n_i = \infty$ and f_i is in the closure of $R(tn_i)$ for each $i \in \mathbb{N}$, then

- i) $[[t]](O) = \lim_{i \rightarrow \infty} f_i$.
- ii) $[[t]](O)$ is in the closure of $R(t)$.

²In Section 4 we will just assume that X is searchable by T with respect to continuous predicates, We will then modify our proof and also define an alternative to $R(s)$.

Proof

ii) is a consequence of i) and the fact that the closure of $R(tn_i)$ is a subset of the closure of $R(t)$ for each $i \in \mathbb{N}$. In order to prove i) it is sufficient to assume that $f_i \in R(tn_i)$ for each $i \in \mathbb{N}$.

The point of selecting the leftmost path is that if $n < n_i$, then $[[tn]](P_{f_i}) \neq f_i$, so $P_{f_i}([[tn]](P_{f_i})) = 0$ when $n < n_i$. It follows that

$$[[t]](O) = F_t(\lambda n.0) = \lim_{i \rightarrow \infty} F_t(\lambda n. [[tn]](P_{f_i})) = \lim_{i \rightarrow \infty} [[t]](P_{f_i}) = \lim_{i \rightarrow \infty} f_i.$$

This ends the proof of the lemma.

We now define $\phi(t)$ for $t \in T^*$ as follows

- $\phi(t) = [[t]](O)$ if $[[t]](O)$ is in the closure of $R(t)$.
- $\phi(t) \in R(t)$ is arbitrary otherwise.

Lemma 2.8. ϕ is continuous and onto X , when T^* has the order topology of the Kleene-Brouwer ordering on T^* .

Proof

ϕ will be surjective because when t is the leftmost node critical for f we have that $f = \phi(t)$.

Let $t \in T^*$. We will prove that ϕ is continuous in the point t . If t is an isolated point in the order topology, there is nothing to prove, so assume that t is a cluster point. It suffices to prove continuity with respect to sequences $\{t_i\}_{i \in \mathbb{N}}$ where each t_i is a proper extension of t , since any sequence of points different from t , but converging to t , will consist almost everywhere of extensions of t .

Then, for each $i \in \mathbb{N}$ there is an n_i such that t_i extends tn_i , and we will have that $\lim_{i \rightarrow \infty} n_i = \infty$.

Since $\phi(t_i)$ is in the closure of $R(t_i)$ we have, by monotonicity of R , that $\phi(t_i)$ is in the closure of $R(tn_i)$, and also in the closure of $R(t)$.

By Lemma 2.7 we see that $[[t]](O) = \lim_{i \rightarrow \infty} \phi(t_i)$ and thus that $[[t]](O)$ is in the closure of $R(t)$. Then $\phi(t) = [[t]](O)$, and ϕ is continuous with respect to this sequence. Since continuity is equivalent to sequential continuity in this case, this proves that ϕ is continuous.

This ends the proof of Theorem 2.4.

2.4. Pre-evaluation trees

In Section 3 we will prove that whenever a functional Φ of type $((0 \rightarrow 0) \rightarrow 0) \rightarrow (0 \rightarrow 0)$ is definable in Gödel's T, then there is an evaluation tree of rank less than ε_0 that computes Φ . We will use a theorem from Howard [8], where he proves that a specific tree related to a term for Φ has rank $< \varepsilon_0$. The tree we naturally obtain from Howard's analysis will be what we call a *pre-evaluation tree*. In this section we will define this concept and see that each pre-evaluation tree can be transformed to an evaluation tree with the same effect without increasing the rank.

Definition 2.9. a) An *even pre-evaluation tree* will be a well founded decorated tree T of finite sequences of natural numbers such that

- If $t \in T$ has an even length, then $tn \in T$ for all $n \in \mathbb{N}$, and there is no decoration.
- If $t \in T$ has an uneven length, the decoration will be a pair (I_t, F_t) where $I_t \subseteq \mathbb{N}$ and $F_t : (I_t \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is continuous.
- If $t \in T$ has an uneven length, then $tn \in T \Leftrightarrow n \in I_t$.

b) An *uneven pre-evaluation tree* will be like an even one, except that we switch even for uneven and vice versa in the definition

c) If T is an even/uneven pre-evaluation tree, $t \in T$ and F is of pure type 2 we define $[[t]](F) \in \mathbb{N}$ if the length of t is uneven/even and $[[t]](F) \in \mathbb{N}^{\mathbb{N}}$ if the length of t is even/uneven as follows:

- $[[t]](F) = F_t(\lambda n \in I_t. [[tn]](F))$ when the length of t is uneven/even.
- $[[t]](F) = \lambda n. [[tn]](F)$ if the length of t is even/uneven.

- d) Given two trees T and S that either are evaluation or even pre-evaluation trees, we say that they are *equivalent* if $\lambda F. [[e]](F)$ is the same function for the two trees.

Remark 2.1. For even trees, the leaf nodes will be nodes s of uneven length where $I_s = \emptyset$. We may then identify F_s with an integer.

Lemma 2.10. *For every even pre-evaluation tree T of ordinal rank α there is an equivalent evaluation tree T^* of rank $\leq \alpha$.*

Proof

The proof is by transfinite induction over T , but otherwise trivial. Note that we do not require the construction of T^* from T to be effective, and for our applications, it would be irrelevant to do so.

The induction base is when e has no extensions of even length in T .

Then $I_n = \emptyset$ for all n , and F_n is just a number a_n . Then e will be the one and only node in T^* . We let e be decorated with $\lambda n. a_n$.

The induction step is when $I_n \neq \emptyset$ for at least one n .

Let $\{n_i m_i\}_{i \in \mathbb{N}}$ be a surjective enumeration of all sequences nm in T of length 2, and for each $i \in \mathbb{N}$ we let T_i be the child tree of $n_i m_i$.

Bringing the decoration along, T_i will be an even pre-evaluation tree for $\lambda F. [[n_i m_i]](F)$. Let T_i^* be the equivalent evaluation tree for each i , and let T^* consist of e , all sequences i of length one, and all sequences is where $s \in T_i^*$.

We bring with us all decorations from each T_i^* and ad a decoration F_e that we describe informally:

In order to compute $F_e(f)(n)$ let

$$J_n = \{i \mid n_i = n \wedge m_i \in I_n \wedge \forall j < i (n_j \neq n \vee m_j \neq m_i)\}.$$

Then f restricted to J_n will induce a map $g : I_n \rightarrow \mathbb{N}$. We let $F_e(f)(n) = F_n(g)$.

Since each F_n is continuous, F_e will be continuous.

The verifications of the remaining properties are now trivial.

3. Applications to Gödel's T

As mentioned in the introduction, our aim is to show that if a set X is searchable with an algorithm expressible in Gödel's T, then the Cantor-Bendixson rank of X is strictly bounded by the ordinal ε_0 . Using Theorem 2.4 it will be sufficient to prove that if X is searchable in this way, this search can be carried out by an evaluation tree of ordinal rank below ε_0 . We will use results from Howard [8] to establish the ordinal bound, and we will be brief when explaining how we obtain a tree from a term in T.

We will prove that if Φ is T-definable and of type

$$((0 \rightarrow 0) \rightarrow 0) \rightarrow (0 \rightarrow 0)$$

then there is an evaluation tree for Φ of ordinal rank $< \varepsilon_0$.

Definition 3.1. Let M be a term in T, possibly with free variables.

The *Howard tree* of M will be the tree of finite sequences of terms where M is at the top node and branching is generated by the following definition of *successor*. In all cases except 8 the term of which we define the successors will be of type 0, we assume that everything is well typed and that necessary uses of α -conversions are implicit:

1. If M is a numeral, then M is a leaf node.
2. If $M = (\lambda x M_1) N_1 \cdots N_k$ then $M_1(x/N_1) N_2 \cdots N_k$ is the one successor.
3. If $M = \text{rec}_\sigma \hat{0} N_2 N_3 \cdots N_k$ then $N_2 N_4 \cdots N_k$ is the one successor.
4. If $M = \text{rec}_\sigma n + 1 N_2 N_3 \cdots N_k$ then $(N_3 \hat{n} (\text{rec}_\sigma \hat{n} N_2 N_3)) N_4 \cdots N_k$ is the one successor.

5. If $M = \text{rec}_\sigma N_1 \cdots N_k$ where N_1 is not a numeral, then N_1 is a successor (N_1 will be of type 0) and $\text{rec}_\sigma \hat{n} N_2 \cdots N_k$ are successors for each $n \in \mathbb{N}$.
6. If $M = \text{suc } N$ then N is the one successor.
7. If M is of the form $ZN_1 \cdots N_k$ where Z is a variable, then the successors are N_1, \dots, N_k .
8. If M is a term of type $\neq 0$, then the successors will be all $MN_1 \cdots N_k$ where each N_i either is a variable or a numeral.

Theorem 3.2 (Howard [8]). *The Howard tree will always be well founded.*

Moreover, for each $n \in \mathbb{N}$ there is an ordinal $\alpha_n < \varepsilon_0$ such that for each term M with all subterms having types of level $< n$, the Howard tree will have rank $< \alpha_n$.

We refer to the original paper [8] for the proof.

Our situation will not be as general as this. The terms we consider will have one free variable \hat{P} at the most. Then 7. will reduce to the case $\hat{P}N$ where the successor will be N .

The successors of N will be all terms Nx , where x is a variable of type 0, and $N\hat{n}$ for each n . Our *reduced Howard tree* will be the one where we only consider successors $N\hat{n}$ to terms N of type $0 \rightarrow 0$. These reduced trees will of course also satisfy Theorem 3.2.

Theorem 3.3. *Let Φ of type $((0 \rightarrow 0) \rightarrow 0) \rightarrow (0 \rightarrow 0)$ be definable in Gödel's T.*

Then there is an evaluation tree of rank $< \varepsilon_0$ computing Φ .

Proof

By Lemma 2.10 it is sufficient to produce an even pre-evaluation tree.

Let \hat{P} be a variable of type $(0 \rightarrow 0) \rightarrow 0$ and let M_0 with at most the variable \hat{P} free be such that M_0 defines Φ . We will need uneven pre-evaluation trees for the terms $M_0\hat{n}$ for each n in order to produce the even pre-evaluation tree for Φ , and in order to prove their existence we give a general construction.

Let M be an arbitrary term of type 0 with only \hat{P} free and let H be the reduced Howard tree of M . The construction of the uneven pre-evaluation tree will be by the same cases as the definition of H , where some of the cases are treated together:

1. If M is a numeral \hat{n} , we let $I_e = \emptyset$ and F_e be the constant n .
2. If M is of cases 2., 3. or 4. in Definition 3.1, let M_1 be the immediate successor and we use the uneven pre-evaluation tree for M_1 .
3. If $M = \text{suc } M_1$ and H_1 is the reduced Howard tree of M_1 , we let our pre-evaluation tree for M be constructed from H_1 , except that we increase the values of F_e with 1.
4. If M is of case 5, the pre-evaluation tree will be constructed as a kind of amalgamation of the pre-evaluation trees of N_1 and each $\text{rec}_\sigma \hat{n} N_2 \cdots N_k$.

Let J be the set of immediate successors of e and let G be the evaluation function for e in the tree for N_1 . Let I_n, F_n be the corresponding items for $\text{rec}_\sigma \hat{n} N_2, \dots, N_k$.

We imbed J and the I_n 's into disjoint sets J' and I'_n in \mathbb{N} , and let I be the union, where p and q_n are the injective maps.

We let the child tree of $m \in J$ in our tree for N_1 now be the child tree of $p(m)$, and likewise we transport child trees of $m \in I_n$ to child trees of $q_n(m) \in I$. If $f : I \rightarrow \mathbb{N}$ we compute $F(f)$ as follows: Let $n = G(p^{-1}(f))$ and then let $F(f) = G_n(F_n(q_n^{-1}(f)))$. This F will be our new F_e .

5. If $M = \hat{P}N$, the successors two steps down will be all terms of the form $N\hat{n}$. This is where the reduction of the Howard tree comes in. Our pre-evaluation tree will now have a branching to all n and the node n is continued with the pre-evaluation tree constructed from the reduced Howard tree of the term $N\hat{n}$.

It now follows by an easy transfinite induction on the reduced Howard tree that the uneven pre-evaluation tree will compute the number that is the value of M .

4. Search with respect to continuous predicates only

In this section we will still let T be an evaluation tree for a search operator on a closed set $X \subseteq \mathbb{N}^{\mathbb{N}}$, but we will only assume that $[[e]]$ is a search operator on X with respect to predicates P that are continuous on X .

The main result of the section will be that the CB-rank of X is still bounded by the Kleene-Brouwer order type of T . The idea and structure of the proof is like our proof of Theorem 2.4 with an adjusted definition of the predicate R .

We will let s, t range over nodes in T . We let Σ be the set of finite sequences with extensions in X . We let σ , possibly with indices, range over Σ . We will consider σ to be a finite partial function, so an expression like $f = \lim_{n \rightarrow \infty} \sigma_n$ will make sense. Our argument will work for all compact sets $X \subseteq \mathbb{N}^{\mathbb{N}}$, but to ease the use of notation, we let $X \subseteq \{0, 1\}^{\mathbb{N}}$. Violating standard terminology, we let $\tilde{f}(n)$ denote the sequence $(f(0), \dots, f(n-1))$, and not the sequence number of this sequence.

We let

$$B_{\sigma} = \{f \in X \mid \sigma \prec f\}$$

and we let P_{σ} be the characteristic function of B_{σ} restricted to X . As before, we let O denote the constant zero function on X , the characteristic function of the empty predicate. We let P , possibly with an index, denote a continuous predicate on X given as a characteristic function.

We will now redefine some concepts from the proof of Theorem 2.4 and prove the analogue lemmas and theorem.

Definition 4.1. Let P be a non-empty continuous predicate on X , $s \in T$.

s is *relevant* for P if $P([[t]](P)) = 1$ for all $t \preceq s$.

We say that s is *leftmost relevant* for P if s is on the leftmost branch in T consisting of t 's that are relevant for P .

We say that s is *critical* for P if s is relevant for P and there is no $k \in \mathbb{N}$ such that sk is relevant for P .

We observe that if s is relevant for P and s is a leaf node in T , then s is critical for P . We also observe that e is relevant for all non-empty continuous predicates P . This is a consequence of the assumption that $[[e]]$ is a search operator for X . Thus there will be a leftmost path of relevant nodes for P ending in a node that is critical for P .

Lemma 4.2. If s is critical for P , then $P([[s]](O)) = 1$.

The proof is essentially the same as the proof of Lemma 2.6.

We now define the alternative R^+ of R :

Definition 4.3. Let $t \in T$. Let $f \in R^+(t)$ if t is leftmost relevant for $P_{\tilde{f}(n)}$ for infinitely many $n \in \mathbb{N}$.

Let

$$T^+ = \{t \in T \mid R^+(t) \neq \emptyset\}.$$

Lemma 4.4. Let $s \in T$ be a branching node. Let $k_n \in \mathbb{N}$ and $\sigma_n \in \Sigma$ for each $n \in \mathbb{N}$. Assume that sk_n is leftmost relevant for P_{σ_n} for each $n \in \mathbb{N}$.

Assume further that $\lim_{n \rightarrow \infty} k_n = \infty$.

Then

$$[[s]](O) = \lim_{n \rightarrow \infty} \sigma_n.$$

Proof

$$[[s]](O) = F_s(\lambda k.0).$$

By the assumption

$$P_{\sigma_n}([[sk]](P_{\sigma_n})) = 0$$

for $k < k_n$ so since $\lim_{n \rightarrow \infty} k_n = \infty$ we have that

$$\lambda k.0 = \lim_{n \rightarrow \infty} \lambda k.P_{\sigma_n}([[sk]](P_{\sigma_n})).$$

Then, since F_s is continuous,

$$[[s]](O) = \lim_{n \rightarrow \infty} F_s(\lambda k.P_{\sigma_n}([[sk]](P_{\sigma_n}))) = \lim_{n \rightarrow \infty} [[s]](P_{\sigma_n}).$$

Since $s \prec sk_n$ for all n we have that s is relevant for each P_{σ_n} , so $\sigma_n \prec [[s]](P_{\sigma_n})$. Finally we observe that since there are only finitely many σ of a fixed length and each k_n can be repeated at most a finite number of times, the lengths of the σ_n 's will converge to ∞ . The claim of the lemma now follows.

Lemma 4.5. *If $f \in X$, there is a node $s \in T^+$ such that $f \in R^+(s)$ and $[[s]](O) = f$.*

Proof

We know that $f \in R^+(e)$ since e is (leftmost) relevant for all continuous predicates.

Since T^+ is well founded it is sufficient to prove that if $f \in R^+(s)$ then either $f = [[s]](O)$ or $f \in R^+(sk)$ for some k .

So assume that $f \in R^+(s)$ and that $f \neq [[s]](O)$.

Let $\{n_i\}_{i \in \mathbb{N}}$ be an increasing sequence such that s is leftmost relevant for $P_{\bar{f}(n_i)}$ for each $i \in \mathbb{N}$. For each $i \in \mathbb{N}$ we have that either s is critical for $P_{\bar{f}(n_i)}$ (which will be the case when s is a leaf node) or there is some k_i such that sk_i is leftmost relevant for $P_{\bar{f}(n_i)}$.

If s is critical for $P_{\bar{f}(n_i)}$, we use Lemma 4.2 and see that $\bar{f}(n_i) \prec [[s]](O)$. Since we assumed that $f \neq [[s]](O)$ this can only be the case for finitely many $i \in \mathbb{N}$, so for all but finitely many $i \in \mathbb{N}$ there is a leftmost k_i such that sk_i is relevant for $P_{\bar{f}(n_i)}$.

If $\{k_i \mid i \in \mathbb{N}\}$ is unbounded we may find a strictly increasing subsequence $\{k_{i_j}\}_{j \in \mathbb{N}}$ and then, by Lemma 4.2, we have that

$$f = \lim_{j \rightarrow \infty} \bar{f}(n_{i_j}) = [[s]](O),$$

contradicting our assumption.

Thus $\{k_i \mid i \in \mathbb{N}\}$ is bounded, and there will be one number k and a subsequence $\{n_{i_j}\}_{j \in \mathbb{N}}$ of $\{n_i\}_{i \in \mathbb{N}}$ such that $k_{i_j} = k$ for all $j \in \mathbb{N}$.

Then $f \in R^+(sk)$. This ends the proof of the lemma.

We now let $\phi : T^+ \rightarrow X$ be defined as before:

- $\phi(s) = [[s]](O)$ if $[[s]](O)$ is in the closure of $R^+(s)$.
- $\phi(s) \in R^+(s)$ is chosen arbitrarily otherwise.

As before we have that

$$t \preceq s \Rightarrow R^+(s) \subseteq R^+(t)$$

and then, by construction, we have that $\phi(t)$ is in the closure of $R^+(s)$ whenever $s \preceq t$.

By Lemma 4.5 we have that ϕ is surjective. We will prove that ϕ is continuous, and draw our main consequence as before.

Lemma 4.6. *Let $s \in T^+$, let $sk_i \in T^+$ for each $i \in \mathbb{N}$.*

Assume that $\lim_{i \rightarrow \infty} k_i = \infty$ and let f_i be in the closure of $R^+(sk_i)$ for each $i \in \mathbb{N}$.

Then

$$[[s]](O) = \lim_{i \rightarrow \infty} f_i.$$

Proof

It is sufficient to assume that $f_i \in R^+(sk_i)$ for each i .

Let $\{n_i\}_{i \in \mathbb{N}}$ be a strictly increasing sequence from \mathbb{N} such that sk_i is leftmost relevant for $P_{\tilde{f}_i(n_i)}$ for each i . The existence of this sequence follows from the definition of R^+ .

By Lemma 4.2 we have that

$$[[s]](O) = \lim_{i \rightarrow \infty} \tilde{f}(n_i) = \lim_{i \rightarrow \infty} f_i.$$

This ends the proof of the lemma.

Lemma 4.7. *The function ϕ is continuous.*

The proof is as the proof of Lemma 2.8.

We now have everything we need to prove the following

Theorem 4.8. *Let $X \subseteq \{0, 1\}^{\mathbb{N}}$ be closed and assume that there is an evaluation tree T such that for all continuous predicates P on X we have that*

$$\exists f \in X(P(f) = 1) \Leftrightarrow P([[e]](P)) = 1.$$

Then X is countable, and the Cantor-Bendixson rank of X is bounded by the Kleene-Brouwer order type of the tree T .

5. Concluding remarks

5.1. Summary

We have been investigating the topological nature of sets in $\mathbb{N}^{\mathbb{N}}$ that are searchable via an operator definable in Gödel's T. Howard [8] showed how terms in T give rise to computation trees of rank less than ϵ_0 , and we have transformed his trees to what we called *evaluation trees*. The main technical result is that if a set X is searchable for all predicates on X with the aid of an evaluation tree, then the rank of the Kleene-Brouwer ordering bounds the Cantor-Bendixson rank of X . It is sufficient to assume that this holds for continuous predicates on X .

We also discussed the consequence of X being searchable in the sense of Kleene's original S1 - S9.

5.2. Related problems

Our proof is non-constructive, even from the point of view of classical logic, since there are no algorithms for constructing the trees T^* and T^+ used in the proofs. Thus our proofs do not say much about the internal structure of searchable sets beyond the bounds on the Cantor-Bendixson rank. We are nowhere near a characterization of when a closed set may be searchable in T, neither with respect to continuous predicates nor with respect to all predicates.

Problem 5.1. If X is T-searchable with respect to continuous predicates, is X then T-searchable with respect to all predicates?

The problem may not be so interesting in itself, but a positive solution will probably give us more information about the structure of search algorithms. We may ask a similar question for searchability by evaluation trees.

We also discussed briefly what can be said about a set X that is searchable in S1-S9, and concluded that X is computably enumerable and compact. We sketched an argument showing that when X is compact and computably enumerable, then the Cantor-Bendixson rank of X is a computable ordinal.

The natural problem here is if we can characterize all sets $X \subseteq \mathbb{N}^{\mathbb{N}}$ that are S1 - S9 - searchable in Kleene's original sense. It is easy by the methods developed in Escardó [3] to prove that we for every computable ordinal α can find a set X searchable in this sense and with a CB-rank exceeding α . We know that if X is a S1 - S9 - searchable subset of the Cantor space, then X is closed, computably enumerable and the set of neighbourhoods containing elements from X is decidable. It is not likely that these properties are sufficient for proving the converse. The problem will be to find some informative characterization of the compact sets X that are S1 - S9 - searchable.

The following problem is also open

Problem 5.2. Assume that X is S1 - S9 - searchable. Is X then searchable by a computable evaluation tree?

The search operators constructed by Escardó [2, 3] do not just provide an element of each non-empty set, they search through all the elements of X (or a dense subset of X when we only consider continuous predicates) until they find an element satisfying P , and then this element will be the output. This behaviour cannot be described as a denotational property of the search operator, but it is tempting to look for further properties of search operators that might be used to analyse when a set e.g. is *sequentially searchable*. One possible such property is the following monotonicity property:

$$\text{If } e(P) = f \in P \text{ and } f \in Q \subseteq P \text{ then } e(Q) = f.$$

We believe that the concept of searchability must be strengthened if we want to deduce further properties of the operational semantics from the denotational interpretation of a search operator and the mere fact that it is definable in some given calculus.

Acknowledgements

During the research for this note, I had, as explained in footnote 1, important and valuable discussions with Martín Escardó. To a large extent these discussions took place at the Isaac Newton Institute for Mathematical Sciences in Cambridge, England. I am grateful to the Institute for hosting the program *Semantics and Syntax: A Legacy of Alan Turing* and for inviting me to take part in this program. During this stay, I also benefitted from discussions with the proof theorists Arnold Beckmann, Anton Setzer and Monica Seisenberger.

An anonymous referee gave valuable feedback on the exposition.

References

- [1] Berger, U.: Totale Objekte und Mengen in der Bereichstheorie (in German), Thesis, München 1990.
- [2] Escardó, M. H.: Exhaustible sets in higher-type computation, *Logical Methods in Computer Science* **4(3)**, 2008.
- [3] Escardó, M. H.: Infinite sets that satisfy the principle of omniscience in all varieties of constructive mathematics, *Journal of Symbolic Logic* **78(3)**, 2013, 764 - 784.
- [4] Di Gianantonio, P.: A Functional Approach to Computability on Real Numbers, Thesis, Università di Pisa - Genova - Udine 1993.
- [5] Di Gianantonio, P.: Real Number Computability and Domain Theory, *Information and Computation* **127**, 1996, 11 - 25.
- [6] Di Gianantonio, P.: An abstract data type for real numbers, *Theoretical Computer Science* **221**, 1999, 295 - 326.
- [7] Gödel, K.: Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes, *Dialectica* **12**, 1958, 280- 287.
- [8] Howard, W. A.: Ordinal analysis of terms of finite type, *Journal of Symbolic Logic* **45(3)**, 1980, 493 - 504.
- [9] Kleene, S. C.: Recursive functionals and quantifiers of finite types I, *Transactions of the American Mathematical Society*, **91(1)**, 1959, 1 - 52.

- [10] Kleene, S. C.: Countable functionals, in A. Heyting (ed.) *Constructivity in Mathematics*, North-Holland, 1959, 81 - 100.
- [11] Kreisel, G.: Interpretation of analysis by means of functionals of finite type, in A. Heyting (ed.) *Constructivity in Mathematics*, North-Holland, 1959, 101 - 128.
- [12] Longley, J. R. and Normann, D.: *Higher Order Computability*, to appear, Springer Verlag.
- [13] Normann, D.: Computability over the partial continuous functionals, *Journal of Symbolic Logic* **65(3)**, 2000, 1133 - 1142.
- [14] Plotkin, G. D.: LCF considered as a programming language, *Theoretical Computer Science* **5(3)**, 1977, 223 - 255.
- [15] Scott, D. S.: A type-theoretical alternative to ISWIM, CUCH, OWHY, Unpublished notes, Oxford 1969.
- [16] Simpson, A. S.: Lazy Functional Algorithms for Exact Real Functionals, *Mathematical Foundations of Computer Science 1998*, Springer LNCS 1450, 1998, 456 - 464.
- [17] Tait, W. W.: Continuity properties of partial recursive functionals of finite type, unpublished notes 1958.