Computability 0 (2020) 1–0
IOS Press

Measure-theoretic Uniformity and the Suslin Functional

Dag Normann
Department of Mathematics, The University of Oslo, Norway
dnormann@math.uio.no

Abstract. Given a set A in the unit interval and the associated Lebesgue measure λ, it is a natural question whether we may (in some sense) compute the measure $\lambda(A)$ in terms of the set A. Under the moniker *measure theoretic uniformity*, Tanaka and Sacks have (independently) provided a positive answer for the well-known class of *hyperarithmetical* sets of reals, and provided a basis theorem for such sets of positive measure. The hyperarithmetical sets are exactly the sets computable in terms of the functional ${}^2E$, in the sense of Kleene's S1-S9. In turn, Kleene's ${}^2E$ essentially corresponds to *arithmetical comprehension* as in ACA_0. In this paper, we generalise the aforementioned results to the 'next level', namely Π^1_1-CA_0, in the form of the *Suslin functional*, or the equivalent *hyperjump*. We also generalise the Tanaka-Sacks basis theorem to sets of positive measure that are semi-computability relative to the Suslin functional. Finally, we discuss similar generalisations for infinite time Turing machines.

Keywords: Measurability, Suslin operator, C-set, infinite time Turing machine

1. Introduction

1.1. Measure, uniformity, and computability

Infamous as they may be to the highschool students required to memorise them, the formulas for determining the length, surface, or volume of basic objects in three-dimensional Euclidean space are *highly elementary* in the sense of computational complexity. Now, the Lebesgue measure λ assigns a real number $\lambda(A)$ to general classes of sets A conforming to the intuitive rules governing length, surface, and volume (as much as possible). In its full generality, measure theory is a complicated affair as even the countable additivity of the Lebesgue measure requires a fragment of the Axiom of Choice not provable in ZF [8]. In light of this, it is a natural question whether we may (somehow) compute the measure $\lambda(A)$ in terms of the set A, for certain classes of sets. In a nutshell, the aim of this paper is to provide a substantial generalisation of established results, known as *measure-theoretic uniformity*, by Tanaka [23] and Sacks [17], on the computational properties of the Lebesgue measure. We should note that 'computable' in general refers to Kleene's well-known schemes S1-S9, see [5] or [10, p. 169], *unless explicitly stated otherwise*. We give our equivalent, modified version of Kleene computability in Definition 1.1.

First of all, the aforementioned results due to Tanaka and Sacks (see Theorem 1.3 for the exact formulation), deal with the well-known *hyperarithmetical* sets. The latter are exactly the sets Kleene-computable in the functional ${}^2E$ (see Section 1.2), and computing with ${}^2E$ represents a natural transfinite extension of the *arithmetical comprehension principle*, expressed by the axiom ACA_0 from [21, III], up to the first non-computable ordinal. In this paper, we generalise the Tanaka-Sacks results to the 'next level', namely Π^1_1-CA_0, in the form of the *Suslin functional* $\mathbf{S}$, or the equivalent *hyperjump* (see Section 1.2 for definitions). We also generalise the basis theorem for Π^1_1-sets of positive measure to sets computable and semi-computable relative to $\mathbf{S}$. Note that computing with $\mathbf{S}$ represents a natural transfinite extension of Π^1_1-CA_0, up to the first recursively inaccessible ordinal. In Section 1.3 we give reference to a non-trivial application of the basis theorem, and in Section 3 we provide a new application, i.e. the basis theorem is not mere theory. Finally, we discuss similar generalisations for *infinite time Turing machines*, originally due to Carl and Schlicht [2].

Secondly, hyperarithmetical sets come in two forms, as sets of integers and as sets of sets (or sets of functions), i.e. as objects of order 2 and as objects of order 3. An important aspect is that sets of the second kind can be coded by objects of the first kind, and one aspect of measure theoretic uniformity is that the measure theory of objects of the second kind can be reduced to properties of the codes, staying within the hyperarithmetical fragment of second-order arithmetic. Another aspect is that while there may be a mismatch between the complexity of a hyperarithmetical set and that of its elements, this mismatch disappears for hyperarithmetical sets of positive measure.

2211-3568/20/$35.00

The main result of the paper will broaden the above picture to the class of *C-sets* as defined by Selivanovskij in [19] and the related class of sets computable in the Suslin functional. [**?**] is written in Russian, with a *résumé* in French, and we give the French reference.

Thirdly, as to the technical framework, some of our arguments are well-known from the classical literature on *Descriptive Set Theory*, but the intention is to make the note reasonably self-contained. For arguments and results on measure-theoretic uniformity, we refer to Sacks [18], and not to the original papers. We refer to Kechris [4] for some results in Descriptive Set Theory. Finally, there are several sources for the definitions of admissible ordinals, e.g. Sacks [18, Chapter VII] and the recent Normann [12, §2]

Finally, as to content, in Section 1.2 we state the known results on measure theoretic uniformity and the proposed generalisation. In Section 1.3 we place our results in a wider context, and discuss why Kleene computability is an appropriate notion of higher-order computability (in this wider context). In Remark 1.5 we provide a brief account of research related to the contents of Sections 2 and 3. In Section 2 we introduce the C-sets of Selivanovskij and show how the measure theory of those sets is captured by $\mathbf{S}$-computability. The key step will be to give a natural coding of the C-sets and then, given a code for a C-set A, use $\mathbf{S}$ to compute codes for Borel-sets $B_1 \subseteq A \subseteq B_2$ approximating A in measure. In Section 3 we show how the termination predicate for computing relative to $\mathbf{S}$ can be approximated by C-sets, and we prove our main results for computability relative to $\mathbf{S}$ in Proposition 1.4. As an application, we also provide a selection theorem for C-sets. In Section 4 we place our results in a broader picture, namely in the framework of infinite time Turing machines. In Section 5 we round off with discussions and acknowledgements.

1.2. Definitions and main results

1.2.1. Some definitions from computability theory

First of all, Kleene's functional ${}^2E$ is defined as follows for any $f : \mathbb{N} \to \mathbb{N}$:

$$ {}^2E(f) := \begin{cases} 0 & \text{if } (\forall n \in \mathbb{N})(f(n) = 0) \\ 1 & \text{if } (\exists n \in \mathbb{N})(f(n) > 0) \end{cases} . $$

For now, we just point out that hyperarithmetic theory can be formulated via computations relative to ${}^2E$.

Secondly, the *Suslin functional* $\mathbf{S}$ is the functional of pure type two defined as follows for any $g : \mathbb{N} \to \mathbb{N}$:

$$ \mathbf{S}(g) := \begin{cases} 0 & \text{if } (\exists f \in \mathbb{N}^{\mathbb{N}})(\forall n \in \mathbb{N})(g(\bar{f}(n)) = 0) \\ 1 & \text{if } (\forall f \in \mathbb{N}^{\mathbb{N}})(\exists n \in \mathbb{N})(g(\bar{f}(n)) > 0) \end{cases} . $$

The Suslin functional is computationally equivalent to the *hyperjump*, and is closely related to the Π^1_1-comprehension axiom from [21]. At the same time, the Suslin functional is the characteristic function of a complete Π^1_1-set on $\mathbb{N}^{\mathbb{N}}$ and the functional version of the Suslin operator, see Subsection 2.1.

Thirdly, Kleene has introduced in [5] a notion of higher-order computation via his schemes S1-S9, see e.g. [10, Section 5.1] or [12, §3] for recent formulations. The Kleene schemes define a relation $\{e\}(\vec{\Phi}) = n$ by transfinite recursion, where Φ is a finite sequence of functionals of *pure type* at any level. In this note, we will only consider input sequences $\vec{\Phi}$ where all but at most one Φ_i are of type 0 or 1, and where any higher-order input F will be of type two, mainly $F = {}^2E$ or $F = \mathbf{S}$. For the sake of notational simplicity, we base ourselves on this simplified definition, where we also omit the redundant scheme S5 for primitive recursion. We will refer to this definition in Section 3.

Definition 1.1. Using transfinite recursion, we define the relation $\{e\}(F, \vec{f}, \vec{a}) = c$, where F is of type $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, $\vec{f} = (f_1, \dots, f_n)$ is a sequence from $\mathbb{N}^{\mathbb{N}}$, $\vec{a} = (a_1, \dots, a_k)$ is a sequence from $\mathbb{N}$ and $c \in \mathbb{N}$, as follows.

S1 If $e = \langle 1 \rangle$, then $\{e\}(F, \vec{f}, \vec{a}) = a_1 + 1$.

S2 If $e = \langle 2, q \rangle$, then $\{e\}(F, \vec{f}, \vec{a}) = q$.

S3 If $e = \langle 3\rangle$, then $\{e\}(F, \vec{f}, \vec{a}) = a_1$.
S4 If $e = \langle 4, e_1, e_2\rangle$, $\{e_2\}(F, \vec{f}, \vec{a}) = b$ and $\{e_1\}(F, \vec{f}, b, \vec{a}) = c$, then $\{e\}(F, \vec{f}, \vec{a}) = c$.
S6 If $e = \langle e_1, \tau_1, \tau_2\rangle$, where τ_1 and τ_2 are permutations of (the index sets for) the input sequences $\vec{f}$ and $\vec{a}$, then $\{e\}(F, \vec{f}, \vec{a}) = \{e_1\}(F, \vec{f}_{\tau_1}, \vec{a}_{\tau_2})$.
S7 If $e = \langle 7\rangle$, then $\{e\}(F, \vec{f}, \vec{a}) = f_1(a_1)$.
S8 If $e = \langle 8, e_1\rangle$, then $\{e\}(F, \vec{f}, \vec{a}) = F(\lambda b.\{e_1\}(F, \vec{f}, b, \vec{a}))$ when defined.
S9 If $e = \langle 9\rangle$ then $\{e\}(F, \vec{f}, d, \vec{a}) = c$ if $\{d\}(F, \vec{f}, \vec{a}) = c$.

On purpose, we omitted S5, the scheme for primitive recursion. For S8, we require that $\{e_1\}(F, \vec{f}, b, \vec{a})$ has a value for all $b \in \mathbb{N}$, which means that an entry $\{e\}(F, \vec{f}, \vec{a}) = c$ in the inductively defined set of *computations* may have infinitely many predecessors. We introduce *non-terminating* computations with S9, since we do not provide a value if d does not code a scheme. The scheme S9 suffices to prove the recursion theorem for Kleene-computations, and this is why both primitive recursion and the μ-operator are redundant. Now, each computation will have a *rank*, the countable ordinal where the computation is added in this inductive process.

Fourth, the computability theory of ^2E is well-behaved and well-treated in the literature. We survey what we need below, and use Sacks [18] as our reference text for this brief survey. For all results but one, Part A of [18] will do. The monotone inductive definition of $\{(e, \vec{a}, c) : \{e\}(^2E, \vec{f}, \vec{a}) = c\}$ is *arithmetical* in $\vec{f}$, and thus the relation $\{e\}(^2E, \vec{f}, \vec{a}) = c$ will be a Π^1_1-relation by [18, III.8]. Thus, every set $A \subset \mathbb{N}^\mathbb{N}$ computable in ^2E is Δ^1_1, which is equivalent to being hyperarithmetical, and will have a hyperarithmetical code as a Borel set by [18, II].

Fifth, we use the standard definition of ω_1^{CK} as the first non-computable ordinal and ω_1^f as the first ordinal that is not computable in the function f. The ordinal ω_1^{CK} is the least ordinal $\alpha > \omega$ that is *admissible* in the sense that L_α satisfies the Kripke-Platek axioms, and this characterisation relativises to any function f.

Finally, we let $\mathbf{C} = \{0,1\}^\mathbb{N}$ be *Cantor space* and we let $\mathbf{m}$ be the standard product measure on $\mathbf{C}$.

Remark 1.2. We may view the elements of $\mathbf{C}$ as binary expansions of reals in $[0,1]$, a representation that is one-to-one except at countably many points. With this view, $\mathbf{m}$ is equivalent to the Lebesgue-measure λ on $[0,1]$, and we may freely transfer results from one measure to the other.

1.2.2. Measure-theoretic uniformity and its generalisation

First of all, we consider the following results independently due to Tanaka [23] and Sacks [17]. As explained in the previous section, the aim of this paper is to establish the generalisation of these results to 'the next level' as embodied by Π^1_1-CA_0, i.e. we establish measure-theoretic uniformity relative to $\mathbf{S}$. Note that a Borel set is the same as a set hyperarithmetical in some function f, which is the same as a set computable in ^2E and some function f.

Proposition 1.3 (Measure-theoretic uniformity relative to ^2E)**.**

a) If $A \subseteq \mathbf{C}$ is a Borel set, then $\mathbf{m}(A)$ is hyperarithmetical uniformly in any Borel-code for A. This is a corollary of [18, Theorem IV.1.3].
b) We have $\mathbf{m}(\{f \in \mathbf{C} : \omega_1^f = \omega_1^{\mathrm{CK}}\}) = 1$ [18, Corollary IV.1.6].
c) If $A \subseteq \mathbf{C}$ is hyperarithmetical and $\mathbf{m}(A) > 0$*, then there is a hyperarithmetical $f \in A$* [18, Theorem IV.2.2].

An important consequence of Proposition 1.3.c), proved via Gandy selection, is that if $A \subseteq \mathbf{C}$ is hyperarithmetical with $\mathbf{m}(A) > 0$, then we can compute an element $f \in A$ *uniformly* from a Kleene-index for A and ^2E. By contrast, there are arithmetical sets with no hyperarithmetical elements. Gandy selection is proved in a more general form as in [18, Theorem X.4.1] and was originally proved in [3]. We will discuss an application of this in Section 1.3, an application that was the original motivation for the aforementioned generalisation.

Secondly, the technical results to be proved in Section 3 are as follows.

Proposition 1.4 (Measure-theoretic uniformity relative to **S**).

a) If $A \subset \mathbf{C}$ is computable in **S***, then $\mathbf{m}(A)$ is computable in* **S***. This relativises to all functions of type 1.*
b) The set of $f \in \mathbf{C}$ such that all ordinals computable in f and **S** *are computable just in* **S** *has measure 1. This set is the set of f such that the least recursively inaccessible ordinal is recursively inaccessible relative to f. We denote this result as $\mathbf{m}(\{f \in \mathbb{N}^{\mathbb{N}} : \omega_1^{f,\mathbf{S}} = \omega_1^{\mathbf{S}}\}) = 1$.*
c) If $A \subseteq \mathbf{C}$ is computable in **S** *and $\mathbf{m}(A) > 0$, then A contains an element computable in* **S**.

The conditions of the theorem are really necessary: there is in fact a subset of **C** computable in **S** with one single element, and this element is not computable in **S**. Thus the assumption that the measure be positive is essential for the basis theorem. The set in question is defined as the set of least fixed points of the monotone inductive definition of the set of computations relative to **S**. Moreover, it is possible to construct a nonempty Π^1_1-set $X \subseteq \mathbf{C}$ which is computable in **S** but has no element even computable by an infinite time Turing machine, the set of codes for executions of the universal machine until the full behaviour of all machines is shown through termination or cyclic computations, see e.g. the discussion in [25, p. 502].

Finally, it is well-known that all subsets of $\mathbb{N}^{\mathbb{N}}$ (semi-)computable in **S**, are Δ^1_2, and since this is provable in ZFC, all such sets are measurable for any completed Borel-measure on $\mathbb{N}^{\mathbb{N}}$ or on closed subsets thereof. Thus, the measurability of the sets we define will not be an issue. This neat result is due to R. Solovay (unpublished) based on a forcing argument from [22].

1.3. The Bigger picture

In this section we place our results in a wider context by providing a brief survey of the project this research is a part of. Readers only interested in the new results of the paper may skip this part.

This paper is a spin-off from a joint project with Sam Sanders. In this project, we combine methods from proof theory and higher computability theory to analyse the logical and computational complexity of classical theorems in analysis concerning the uncountable. One aspect is to investigate the relative computational complexity of realisers for (higher-order) theorems of ordinary mathematics, like e.g. Heine-Borel compactness for uncountable covers.

For instance, the input for a realiser of the *Heine-Borel theorem* is a map sending each $x \in [0,1]$ to a neighbourhood of itself and the output will be a finite sub-covering. An input for a realiser of the *Vitali covering theorem* is a map sending $x \in \mathbb{R}$ and $k \in \mathbb{N}$ to a neighbourhood of x of radius bounded by 2^{-k}. The output will be a set of pairwise disjoint neighbourhoods in the range of the input covering almost all reals. In [14, Theorem 3.29], it is (indirectly) proved that there is a realiser Λ of this kind for the Vitali Covering Theorem such that no realiser Θ for the Heine-Borel theorem is Kleene-computable in Λ and ${}^2E$. As it happens, Proposition 1.3 of this paper is used in the proof, using both the fact that one can compute the measure of a Borel set from a code for it and ${}^2E$ and that one can effectively, in the sense of ${}^2E$-computability, select an element when the Borel-set has positive measure.

Later, ${}^2E$ was replaced by **S**, yielding [16, Theorem 4.17]. The latter result provides computational evidence for the claim that Π^1_1-comprehension does not suffice to obtain 'full' compactness from compactness 'up to measure'. In contrast, the combination of Π^1_1-comprehension and (a realiser for) the Heine-Borel theorem yields the full strength of non-monotone inductive definitions of subsets of $\mathbb{N}$ by [11, Theorem 1]. We note that Theorem 3.9 of this paper, and its corollaries, are essential for the proof of the theorem from [16]. We refer to the Introduction in [15] for a discussion of the foundational interest of such realisers.

As to alternative approaches, there is no *Turing thesis* for higher-order computability, and Kleene's definition is just one possible generalisation of classical (Turing) computability to computability relative to higher-order objects. Alternatively, one could consider the weaker typed λ-calculus that can be relativized by the use of constants, or the stronger concept of infinite time Turing machines, ITTMs for short. We will take a closer look at ITTMs in Section 4. For the purpose of our foundational project, the ITTM-approach is too strong as many of the type 3 realisers studied in our project can be obtained using non-monotone induction, and thus computed by ITTMs. Indeed, Welch describes in [26] how to compute functionals of type 3 using ITTMs. Thus, Kleene-computability relative to some or all functionals of type 2 is adequate for proving negative results as in

Objects of kind **A** can not be uniformly computed from objects of kind **B**,

since this is the strongest natural concept for computing with higher-order objects for which we have hitherto managed to prove most negative results. Positive results on relative computability in this project require for the most part only typed λ-calculus or a small fragment of Gödel's system T of higher-order primitive recursion.

Remark 1.5 (Related research)**.** First of all, the concept of C-set has been studied for its own sake, in the original paper [19], and in e.g. Shreve [20] and Burgess [1]. What is new in this paper is the connection between the measure theory of C-sets and computability relative to **S**.

Secondly, Hinman describes in [9], via a hierarchy for the set of subsets of $\mathbb{N}$ computable in **S**, an effective version of the C-sets, in much the same way that the hyperarithmetical subsets of $\mathbb{N}$ form an effective version of the Borel sets. We will see that there are subsets of **C** that are computable in **S** but that are not C-sets. In this way, the C-sets form a proper subclass of the sets computable in **S** and some function, in contrast to how computability in ${}^2E$ corresponds to the Borel-sets.

Thirdly, Burgess [1, §3] introduces the C-sets in a similar way as we do in Section 2, and proves a number of selection theorems for C-sets based on topological properties ([1, §8]). Our basis theorem, Proposition 1.4, item c), in conjunction with the Gandy selection theorem, leads to a similar selection theorem based on measure theoretic properties, namely Theorem 3.13. Our methods are quite different from those used by Burgess in [1].

Fourth, Carl and Schlicht proved the analogue of Proposition 1.4 for infinite time Turing machine computability in [2, Theorem 4.5]. In Section 4 we will look closer at how their results connect to ours.

2. The hierarchy of C-sets

We introduce the hierarchy of C-sets and prove some measure and computability theoretic facts.

2.1. Basic facts about C-sets

In this section, we define the notion of 'C-set', first introduced by Selivanovskij in [19], and prove elementary facts.

First of all, let SEQ be the set of finite sequences s of integers $\langle s_0, \ldots, s_{\mathrm{lh}(s)-1}\rangle$ where $\mathrm{lh}(s)$ is the *length* of the sequence. We use s, t for elements of SEQ, and we identify them with the corresponding sequence numbers $\langle s\rangle$ and $\langle t\rangle$. We use the standard notation $\bar{f}(n) = \langle f(0), \ldots, f(n-1)\rangle$. We let $s \preceq t$ if $\mathrm{lh}(s) \leqslant \mathrm{lh}(t)$ and $s_i = t_i$ for $i < \mathrm{lh}(s)$. We let P, Q range over subsets of X, where X may be the Cantor space **C**, the Baire space $\mathbb{N}^{\mathbb{N}}$, or any product of such spaces, possibly with $\mathbb{N}$ as extra factors.

Secondly, a *Suslin scheme* **P** on X is a map $s \mapsto P_s$ from SEQ to the powerset of X. Given a Suslin scheme **P**, we define $\mathbf{A}(\mathbf{P}) = \bigcup_{f\in\mathbb{N}^{\mathbb{N}}} \bigcap_{n\in\mathbb{N}} P_{\bar{f}(n)}$. The functional **A** is known as the *Suslin operator*.

Thirdly, by simultaneous recursion on the countable ordinal α, we now define two classes Σ_α and Π_α of subsets of X. The C-sets inhabit the 'limit' of this recursion process.

Definition 2.1 (C-sets)**.**

i) Σ_0 consists of all sets that are both closed and open.
ii) For $\alpha > 0$, Σ_α is the set of $\mathbf{A}(\mathbf{P})$ where **P** is a Suslin scheme with all P_s in

$$\bigcup\{\Pi_\beta : \beta < \alpha\}.$$

iii) $\Pi_\alpha = \{X \setminus P : P \in \Sigma_\alpha\}$.

Let Σ_{ω_1} be the union of all Σ_α for countable α. The *C-sets* are exactly the elements of Σ_{ω_1}.

It is easy to see that $\Sigma_\alpha \subseteq \Sigma_\gamma$ and $\Pi_\alpha \subseteq \Pi_\gamma$ when $\alpha \leqslant \gamma$, and that $\Pi_\alpha \subseteq \Sigma_{\alpha+1}$ for all α. Thus the complement of a set in Σ_{ω_1} will itself be in Σ_{ω_1}.

We shall need the following lemmas pertaining to the previous definitions.

Lemma 2.2. *For each $\alpha > 0$, Σ_α is closed under the application of* **A** *to Suslin schemes with sets from Σ_α, and consequently the class of C-sets is closed under the application of* **A**.

Proof. The proof is classical, and not too difficult. See e.g. Proposition (25.6) in [4]. □

In the same way we use well-founded trees to code Borel sets, we can use well-founded trees, or rather function codes for them, to code elements in Σ_{ω_1}. We do not provide the tedious details: it is easy to distinguish between codes for Σ_0, codes for Π-sets, and codes for Σ-sets; when a tree codes a set, then we can extract codes for all the building blocks. The ordinal rank of the tree coded by a function f will determine if f is a Σ_α-code or a Π_α-code. The set of these codes will be a complete Π^1_1-set.

Remark 2.3. If we consider the Borel sets as inductively defined from the clopen sets through the processes of countable unions and taking complements, each code for a Borel set can be viewed as a code for a C-set as well. Instead of interpreting a countable branching as representing a union, we interpret it via the use of the aforementioned Suslin operator **A**. However, we must distinguish between the two interpretations of a single code.

Lemma 2.4. *For each ordinal α, the sets Σ_α and Π_α are closed under finite unions and intersections. Moreover, given a finite list of Σ_α-codes or of Π_α-codes, we can compute the codes for the respective unions and intersections in the given classes uniformly in ${}^2E$.*

Proof. By De Morgan's laws, it suffices to prove the lemma for Σ_α. Closure under finite unions is trivial. Closure under finite intersections requires a combinatorial effort, but essentially is proved translating the quantifier change

$$\forall k \leqslant m \exists f \forall n R(k, f, n) \mapsto \exists f \forall k \leqslant m \forall n R(k, (f)_k, n)$$

to a manipulation of Suslin schemes. □

Definition 2.5. A Suslin scheme **P** is *regular* if $s \preceq t \Rightarrow P_t \subseteq P_s$ for all s and t in SEQ.

Every Suslin scheme is equivalent to a regular one in the sense that it defines the same set, see e.g. [4, Exercise 25.5]. By Lemma 2.4 we can 'reorganise' a Suslin scheme to a regular one without increasing the rank. Using the recursion theorem for ${}^2E$, we may even transform a code for a well-founded tree of Suslin schemes to a code for an equivalent tree of regular Suslin schemes, using a ${}^2E$-algorithm. In the sequal, we therefore assume that all our Suslin schemes are regular. Moreover, for $\alpha > 0$, we can use the arguments of Lemma 2.4 to show that the classes of Σ and Π-sets are closed under countable unions and intersections, possibly at the cost of increasing the rank by 1. We can also find codes for the intersection and the union from a sequence of codes for such sets uniformly computable in ${}^2E$. However, it is only for *finite* unions and intersections that we can preserve the rank.

We end this section with an application of the recursion theorem for **S**:

Lemma 2.6. *Each C-set A is uniformly computable in* **S** *and any of its codes. Moreover, if A has code g, then the ordinal ranks of the computations deciding if $f \in A$ from* **S** *and g will be strictly bounded below $\omega_1^{\mathbf{S},g}$*

Since the set of codes is computable in **S** as well, we see, by a standard diagonal argument, that there will be sets computable in **S** that are not C-sets.

2.2. Measurability

In this section, we restrict our attention to $X = \mathbf{C}$, the *Cantor space*. As stated in the introduction, all sets semi-computable in $\mathbf{S}$ will be measurable. In this section, we show that if f is a code for the C-set $P[f]$, then we can compute $\mathbf{m}(P[f])$ from f and $\mathbf{S}$ in the sense of computing its Dedekind cut.

For $f, g \in \mathbb{N}^{\mathbb{N}}$, we let '$f \leqslant g$' mean that f is bounded by g in the pointwise ordering. Likewise, for $s \in \mathrm{SEQ}$ and $f \in \mathbb{N}^{\mathbb{N}}$, '$s \leqslant f$' means that $s_i \leqslant f(i)$ whenever for $i < \mathrm{lh}(s)$.

Definition 2.7. Let $\mathbf{P}$ be a (regular) Suslin scheme and let $f \in \mathbb{N}^{\mathbb{N}}$. Define $\mathbf{A}_f(\mathbf{P}) := \bigcup_{g \leqslant f} \bigcap_{n \in \mathbb{N}} P_{\bar{g}(n)}$.

Lemma 2.8. *With the notation from Definition 2.7, we have* $\mathbf{A}_f(\mathbf{P}) = \bigcap_{n \in \mathbb{N}} \bigcup_{\mathrm{lh}(s) = n, s \leqslant f} P_s$.

Proof. The proof is by an application of the *Weak König's Lemma*, also called WKL in [21], and is left for the reader. We do need that the Suslin scheme is regular. □

In the next classical lemma we use that we are only dealing with sets that are measurable:

Lemma 2.9. *For a Suslin scheme* $\mathbf{P}$*, we have* $\mathbf{m}(\mathbf{A}(\mathbf{P})) = \sup\{\mathbf{m}(\mathbf{A}_f(\mathbf{P})) : f \in \mathbb{N}^{\mathbb{N}}\}$.

Proof. Let $\mathbf{m}(\mathbf{A}(\mathbf{P})) = a$. Let $\epsilon > 0$. By recursion on k, we find $f(k)$ such that the union of all $\bigcap_{n \in \mathbb{N}} P_{\bar{g}(n)}$, where $g(i) \leqslant f(i)$ for $i \leqslant k$, has measure larger than $a - \epsilon(1 - 2^{k+1})$. Then $\mathbf{m}(\mathbf{A}_f(\mathbf{P})) \geqslant a - \epsilon$, which proves the lemma.□

Theorem 2.10. *For a code f of the Σ-set or Π-set $P[f]$, the measure $\mathbf{m}(P[f])$ is computable in $\mathbf{S}$ and f, uniformly in f.*

Proof. We use the recursion theorem and argue by induction on the ordinal complexity of the code. The only non-trivial case is when $P[f] = \mathbf{A}(\mathbf{P})$ where each P_s is of the form $P[f_s]$. As an induction hypothesis we assume, with reference to Lemma 2.4, that for any finite Boolean combination of the sets $P[f_s]$ we can compute the measure using $\mathbf{S}$. We can effectively enumerate all finite Boolean combinations of the sets $P[f_s]$ indexed by the finite sequences s. Thus there will be a function $h = \langle h_a \rangle_{a \in \mathbb{N}}$, computable in $\mathbf{S}$ and f, such that h_a gives us (the code for) the measure of the corresponding Boolean combination of the sets $P[f_s]$. Using lemmas 2.8 and 2.9 we see that the measure of $\mathbf{A}(P[f])$, still seen as a Dedekind cut, will be Σ^1_1 in h, and thus computable in $\mathbf{S}$ and f itself. □

2.3. A basis theorem

It is well-known that any measurable set can be approximated from the inside and from the outside by Borel sets with the same measure. In this section, we show that we can find such sets for $P[f]$ *uniformly* in f and $\mathbf{S}$. This is a crucial step in showing that a C-set A of positive measure contains an element computable in $\mathbf{S}$ and a code for A.

First of all, we establish the following lemma and theorem.

Lemma 2.11. *Let $\mathbf{P}$ and $\mathbf{Q}$ be two Suslin schemes of measurable sets such that $\mathbf{m}(P_s \Delta Q_s) = 0$ for all finite sequences s, where Δ is the symmetric difference. Then $\mathbf{m}(\mathbf{A}(\mathbf{P})) = \mathbf{m}(\mathbf{A}(\mathbf{Q}))$.*

Proof. Since the measure of a Boolean manipulation of countably many sets does not change through perturbations of the sets of measure 0, we can use Lemmas 2.8 and 2.9. □

Theorem 2.12. *Let f be a code for a Σ-set or Π-set $P[f]$. Then uniformly computable in f and $\mathbf{S}$ we can find Borel codes for sets $B_{\subseteq}[f]$ and $B_{\supseteq}[f]$ such that $B_{\subseteq}[f] \subseteq P[f] \subseteq B_{\supseteq}[f]$ and $\mathbf{m}(B_{\subseteq}[f]) = \mathbf{m}(B_{\supseteq}[f])$.*

Proof. Like before, we use the recursion theorem and induction on the rank; like before, the only non-trivial case is an application of the Suslin operator $\mathbf{A}$.

In this way, let $P[f] = \mathbf{A}(\mathbf{P})$ where $P_s = P[f_s]$ for each $s \in$ SEQ, and let $B_{\subseteq}[f_s]$ and $B_{\supseteq}[f_s]$ satisfy the induction hypothesis for each s. Then we can find a g computable in f and $\mathbf{S}$ such that g is the join of the Borel-codes for all $B_{\subseteq}[f_s]$ and $B_{\supseteq}[f_s]$. Now consider the Suslin schemes $\mathbf{B}_{\subseteq}$ and $\mathbf{B}_{\supseteq}$ given by the sets $B_{\subseteq}[f_s]$ and $B_{\supseteq}[f_s]$ respectively. Then both $\mathbf{A}(\mathbf{B}_{\subseteq})$ and $\mathbf{A}(\mathbf{B}_{\supseteq})$ are Σ^1_1-sets relative to g, they have the same measure as $\mathbf{A}(\mathbf{P})$ by Lemma 2.11, and they are a subset, resp. superset of the same.

For $a = \mathbf{m}(\mathbf{A}(\mathbf{P}))$ and $n \in \mathbb{N}$, we can use the Kleene basis theorem (see e.g. [18, Theorem III 1.3]) relative to g to find f_n such that $\mathbf{m}(\mathbf{A}_{f_n}(\mathbf{B}_{\subseteq})) > a - 2^n$. The sequence $\{f_n\}_{n\in\mathbb{N}}$ is computable in g and $\mathbf{S}$, and from this sequence and g we can compute a Borel-code for $\bigcup_{n\in\mathbb{N}} \mathbf{A}_{f_n}(\mathbf{B}_{\subseteq})$, a set with measure a. We use this set as $B_{\subseteq}[f]$.

In order to find $B_{\supseteq}[f]$, we use the relativised version of the theorem saying that if a Π^1_1-set D has a hyperarithmetical measure, there will be a Δ^1_1-subset with the same measure (see [18, Chapter IV]). Thus, there will be a Borel-code hyperarithmetical in g for a subset E of the set $\mathbf{C} \setminus \mathbf{A}(\mathbf{B}_{\supseteq})$ with the same measure as $\mathbf{C} \setminus \mathbf{A}(\mathbf{B}_{\supseteq})$. We use the complement of E as our $B_{\supseteq}[f]$. □

Corollary 2.13. *Let f be a code for a Σ-set or Π-set $P[f]$, and assume that $\mathbf{m}(P[f]) > 0$. Then $P[f]$ contains an element computable in f and* $\mathbf{S}$.

Proof. This follows from Theorem 2.12 and the Tanaka-Sacks basis theorem, see Proposition 1.3, item c. □

In the sequel we will need the following uniform version of Theorem 2.12. The proof is left for the reader.

Theorem 2.14. *Let $A \subset \mathbf{C} \times \mathbf{C}$ be a C-set with code g. Then there are $\mathbf{S}, g$ computable functions F and G from $\mathbf{C}$ to $\mathbb{N}^{\mathbb{N}}$ computing, for each $f \in \mathbf{C}$, Borel-codes for inner and outer approximations-in-measure to $\{h \in \mathbf{C} : (f,h) \in A\}$ such that the ordinal ranks of the computations of $F(f)$ and $G(f)$ are uniformly and strictly bounded by $\omega_1^{\mathbf{S},g}$.*

3. Computing with the Suslin-functional

3.1. Terminating computations and C-sets

We establish that computations in $\mathbf{S}$ terminating at a countable level give rise to C-sets. We refer to Definition 1.1 for our inerpretation of the Kleene schemes S1 - S9.

As to context, it is well-known that if F is a functional of type 2 computable in ${}^2E$, then the ordinal ranks of the computations of $F(f)$, when f varies over $\mathbb{N}^{\mathbb{N}}$, will be bounded by a computable ordinal. However, $\mathbf{S}$ does not share this nice behaviour: the set WO of codes for well-orderings is computable in $\mathbf{S}$, and for each $f \in$ WO we can (uniformly and easily) devise a terminating computation that lasts at least as long as the ordinal rank of f. However, we will see that the C-sets capture computations in $\mathbf{S}$ at a countable level much in the same way the Borel sets capture computations relative to ${}^2E$.

Definition 3.1. Let $\Omega_{\mathbf{S}}$ be the set of sequences $\langle e, \vec{f}, \vec{a}, b\rangle$ such that $\{e\}(\mathbf{S}, \vec{f}, \vec{a}) = b$ and let $||\cdot||_{\Omega_{\mathbf{S}}}$ be the corresponding ordinal rank. We let all arguments come in the order Suslin, functions, integers, as in Definition 1.1.

Theorem 3.2. *For a countable ordinal α and, e, k, $\vec{a}$ and b from $\mathbb{N}$, $\{\vec{f} \in (\mathbb{N}^{\mathbb{N}})^k : ||\langle e, \vec{f}, \vec{a}, b\rangle||_{\Omega_{\mathbf{S}}} < \alpha\}$ is a C-set.*

Proof. We prove this by induction on α. In the case $\alpha = 0$, all sets in question are empty, so this case is trivial. When α is a limit ordinal, all sets in question are countable unions of C-sets by the induction hypothesis, so again, this case is trivial. We are left with the *simple* case when $\alpha = \beta + 1$, where we assume that the theorem holds for β.

There are now *nine* sub-cases: one for each of S1 - S4, S6 - S9 and one for the *otherwise* case. For the cases S1 - S3, S7, the initial computations, we do not even need the induction hypothesis, and for the *otherwise*-case, all sets in question are empty. Case S9, the scheme of enumeration, and case S6, the scheme of permutation, follow by direct applications of the induction hypothesis, since there is only one immediate subcomputation in each case. This leaves us with S4 (composition) and S8 (application of **S**), as follows. Recall that '$\simeq$' means that both sides are undefined or both sides are defined and equal.

For the case S4 (composition), let $e = \langle 4, e_1, e_2\rangle$ be such that $\{e\}(\mathbf{S}, \vec{f}, \vec{a}) \simeq \{e_1\}(\mathbf{S}, \{e_2\}(\mathbf{S}, \vec{f}, \vec{a}), \vec{f}, \vec{a})$. We obtain the required results from the induction hypothesis *and* that C-sets are closed under countable unions, since

$$||\langle e, \vec{f}, \vec{a}, b\rangle||_{\Omega_{\mathbf{S}}} < \alpha \leftrightarrow \exists c \in \mathbb{N}[||\langle e_2, \vec{f}, \vec{a}, c\rangle||_{\Omega_{\mathbf{S}}} < \beta \ \wedge \ ||\langle e_1, \vec{f}, c, \vec{a}, b\rangle||_{\Omega_{\mathbf{S}}} < \beta].$$

For the case S8 (application of **S**), let $\perp$ mean 'undefined' and let

$$\{e\}(\mathbf{S}, \vec{f}, \vec{a}) = \begin{cases} 0 & \text{if } (\exists f \in \mathbb{N}^{\mathbb{N}})(\forall n \in \mathbb{N})(\{e_1\}(\mathbf{S}, \vec{f}, \bar{f}(n), \vec{a}) = 0) \\ 1 & \text{if } (\forall f \in \mathbb{N}^{\mathbb{N}})(\exists n \in \mathbb{N})(\{e_1\}(\mathbf{S}, \vec{f}, \bar{f}(n), \vec{a}) > 0) \\ \perp & \text{otherwise} \end{cases}.$$

Define R as the C-set $R = \bigcap_{s \in \mathrm{SEQ}} \bigcup_{c \in \mathbb{N}} \{\vec{f} : ||\langle e, \vec{f}, s, \vec{a}, c\rangle||_{\Omega_{\mathbf{S}}} < \beta\}$, i.e. the set of $\vec{f}$ for which we apply **S** to a total function at level β. We will consider two Suslin schemes **P** and **Q** where

$$P_s := R \cap \{\vec{f} : ||\langle e_1, \vec{f}, s, \vec{a}, 0\rangle||_{\Omega_{\mathbf{S}}} < \beta\} \text{ and } Q_s := R \setminus \bigcup_{c>0} \{\vec{f} : ||\langle e_1, \vec{f}, s, \vec{f}, c\rangle|| < \beta\}.$$

Then we have that

$$\mathbf{A}(\mathbf{P}) = \{\vec{f} : ||\langle e, \vec{f}, \vec{a}, 0\rangle||_{\Omega_{\mathbf{S}}} < \alpha\} \text{ and } \big[R \setminus \mathbf{A}(\mathbf{Q})\big] = \{\vec{f} : ||\langle e, \vec{f}, \vec{a}, 1\rangle||_{\Omega_{\mathbf{S}}} < \alpha\}.$$

In this case, for $b \neq 0, 1$, there will be no $\vec{f}$ such that $||\langle e, \vec{f}, \vec{a}, b\rangle||_{\Omega_{\mathbf{S}}} < \alpha$ since **S** is $\{0,1\}$-valued, so the set in question will be empty, and we are through. □

Remark 3.3. If we have an ordinal code h for α in the above proof, then a code for the C-set constructed will be computable in h and ${}^2E$, uniformly in h.

3.2. Measure-theoretical uniformity for the Suslin functional

In this section, we adjust arguments from [18] in order to 'relativise to **S** the result that almost all reals are hyperarithmetically low. This will be done via a series of lemmas leading up to Theorem 3.9.

As to representation, the elements of the structure $L_{\omega_1^{\mathbf{S}}}[f]$ can be described as the interpretations t^f of terms $t = t(\mathbf{f})$ where $\mathbf{f}$ is a constant that in each case is interpreted as f. The terms t will be coded by elements in $L_{\omega_1^{\mathbf{S}}}$, and the interpretation t^f will be in $L_{\omega_1^{\mathbf{S}}}[f]$.

Remark 3.4. In our coding, we can replace $\omega_1^{\mathbf{S}}$ with any ordinal α that is admissible, or a limit of admissibles. Indeed, employing a standard projection of the ordinals onto $L[f]$, using a Gödel-numbering with an extra number for $\mathbf{f}$ will do. See e.g. [6, Chapter V, §1] for a precise way to do this. There is no need to introduce forcing-related concepts for our applications.

With the above coding, any Δ_0-formula of $L_{\omega_1^{\mathbf{S}}}$, with free variables x and X for numbers and sets, will then be of the form $\phi(x, X, t_1, \ldots, t_k)$, where ϕ has bounded quantifiers only. If $\phi(t_1, \ldots, t_k)$ is a Δ_0-formula and $t_1, \ldots, t_k \in L_\beta$ for $\beta < \omega_1^{\mathbf{S}}$, then $\{f : L_\beta[f] \models \phi\}$ will be a Borel-set with a code uniformly ${}^2E$-computable in codes for β and $t_1, \ldots, t_k$, and thus with a measure uniformly ${}^2E$-computable in these codes. Since the codes can be chosen to be **S**-computable, the measure will be **S**-computable.

Lemma 3.5 (Push-down lemma)**.** *Let $0 \leqslant r \leqslant 1$, and let $\phi(x, X, t_1, \ldots, t_k)$ be a Δ_0-formula where x and X are free variables and assume that*

$$\mathbf{m}(\{f \in \mathbb{N}^{\mathbb{N}} : (\forall x \in \mathbb{N})(\exists X \in L_{\omega_1^{\mathbf{S}}}[f])\phi(x, X, t_1^f, \ldots, t_k^f)\}) \geqslant r.$$

Then

$$\mathbf{m}(\{f \in \mathbb{N}^{\mathbb{N}} : (\exists \beta < \omega_1^{\mathbf{S}})(\forall x \in \mathbb{N} \exists X \in L_\beta[f])\phi(x, X, t_1^f, \ldots, t_k^f)\}) \geqslant r.$$

Proof. It suffices to prove this for rational numbers r. We have that

$$\{f : (\forall x \in \mathbb{N})(\exists X \in L_{\omega_1^{\mathbf{S}}}[f])\phi(x, X, t_1^f, \ldots, t_k^f)\} = \bigcap_{n \in \mathbb{N}} \{f : (\forall x \leqslant n)(\exists X \in L_{\omega_1^{\mathbf{S}}}[f])\phi(x, X, t_1^f, \ldots, t_k^f)\}.$$

For each $n \in \mathbb{N}$ we then have that $\mathbf{m}(\{f : \forall x \leqslant n \exists X \in L_{\omega_1^{\mathbf{S}}}[f]\phi(x, X, t_1^f, \ldots, t_k^f)\}) \geqslant r$, and for each rational $r' < r$ we can find $\beta_{n,r'} < \omega_1^{\mathbf{S}}$ such that $\mathbf{m}(\{f : \forall x \leqslant n \exists X \in L_{\beta_{n,r'}}[f]\phi(x, X, t_1^f, \ldots, t_k^f)\}) > r'$. We then use Gandy selection for $\mathbf{S}$ to find such $\beta_{n,r'}$. Since r is rational, we can use the bounding principle in $L_{\omega_1^{\mathbf{S}}}$ and conclude that there is one $\beta < \omega_1^{\mathbf{S}}$ larger than all the $\beta_{n,r'}$. Then $\mathbf{m}(\{f : \forall x \in \mathbb{N} \exists X \in L_\beta[f]\phi(x, X, t_1^t, \ldots, t_k^t)\}) \geqslant r$ as required. □

Remark 3.6. Lemma 3.7 and Theorem 3.9 below are consequences of Lemma 2.11 in Carl and Schlicht [2]. They work in the context of random reals and forcing. We give proofs without reference to this context.

Lemma 3.7. *The set of $f \in \mathbf{C}$ such that $\omega_1^{\mathbf{S}}$ is f-admissible has measure 1.*

Proof. We only need to check Δ_0-replacement, since the other Kripke-Platek axioms are trivially satisfied for all f. Since there are only countably many instances of replacement, we only need to verify the lemma for each one of them. By Lemma 3.5, the set of f such that a particular instance of Δ_0-replacement fails, will have measure 0. Thus, the set of f for which this axiom holds will have measure 1. □

Lemma 3.8. *There is a well-ordering $(A, \prec)$ of a subset of $\mathbb{N}$ of order type $\omega_1^{\mathbf{S}}$, semicomputable in $\mathbf{S}$, such that for each $a \in A$, $\{\langle b, c\rangle : b \prec c \prec a\}$ is computable in $\mathbf{S}$, uniformly in a.*

Proof. We let A be the set of computation tuples $a = \langle e, \vec{a}, b\rangle$ such that $\{e\}(\mathbf{S}, \vec{a}) = b$ with the norm $|| \cdot ||$, and we let $a \prec a'$ if $||a|| < ||a'||$ or if $||a|| = ||a'||$ and $a < a'$. This ordering has the desired properties. □

Theorem 3.9. *The set $\{f \in \mathbf{C} : \omega_1^{\mathbf{S}} = \omega_1^{\mathbf{S},f}\}$ has measure 1.*

Proof. Let $(A, \prec)$ be as in Lemma 3.8, let $a \in A$ and let $(A_a, \prec_a)$ be the corresponding initial segment of $(A, \prec)$, coded as the function f_a. Then applying a relativized version of Proposition 1.3, b) we have that

$$\mathbf{m}(\{f : \omega_1^{f_a, f} = \omega_1^{f_a}\}) = 1.$$

Taking the intersection of all these sets and the set from Lemma 3.7, we see that the set of f such that $\omega_1^{\mathbf{S}}$ is both f-admissible and a limit of f-admissibles will have measure 1. □

Corollary 3.10. *Let $F : \mathbf{C} \to \mathbb{N}$ be computable in $\mathbf{S}$. Then there is an ordinal $\alpha < \omega_1^{\mathbf{S}}$ such that the algorithm for F terminates at an ordinal rank below α on a set of measure 1.*

Proof. Define $F(f) = \{e\}(\mathbf{S}, f)$. By Theorem 3.9, the set of f such that $\{e\}(\mathbf{S}, f)$ terminates before $\omega_1^{\mathbf{S}}$ has measure 1. Then for all $n \in \mathbb{N}$ there is an $\alpha_n < \omega_1^{\mathbf{S}}$ such that the set of f where $\{e\}(\mathbf{S}, f)$ terminates before α_n has measure $> 1 - 2^{-n}$. Combining the $\mathbf{S}$-effective version of Theorem 3.2 and Theorem 2.10 we can use Gandy selection for $\mathbf{S}$ to compute α_n from n. Thus $\{\alpha_n\}_{n\in\mathbb{N}}$ is bounded below $\omega_1^{\mathbf{S}}$, and we are through. □

Corollary 3.11. *Let $A \subseteq \mathbf{C}$ be semicomputable in $\mathbf{S}$ and with a positive measure. Then A contains an element computable in $\mathbf{S}$.*

Proof. Let A be such that $f \in A \leftrightarrow \{e\}(\mathbf{S}, f)\downarrow$. By Theorem 3.9, almost all of these computations terminate at an ordinal level below $\omega_1^{\mathbf{S}}$. Thus, there is $\alpha < \omega_1^{\mathbf{S}}$ such that the set of f such that $\{e\}(\mathbf{S}, f)$ terminates before α have positive measure. This is a C-set of positive measure and it contains an element computable in $\mathbf{S}$ by Corollary 2.13.□

Remark 3.12. If $A \subseteq \mathbf{C}$ is computable in $\mathbf{S}$, we can combine Corollary 3.10, Theorem 2.12, and Theorem 3.2 (with Remark 3.3) to search over $L_{\omega_1^{\mathbf{S}}}$ to find both the measure of A computably in $\mathbf{S}$ and the codes for Borel sets $A_{\subseteq} \subseteq A \subseteq A_{\supseteq}$ with the same measure as A.

In [1, §8], Burgess lists a number of selection theorems for C-sets where the selecting function claimed to exist is always C-measurable. As promised in Section 1.1, we obtain a similar result as applications of the above.

Theorem 3.13. *Define the set $E_f := \{g \in \mathbf{C} : (f, g) \in E\}$ where $E \subset \mathbf{C} \times \mathbf{C} = \mathbf{C}^2$.*

- *a) Let $A \subset \mathbf{C}^2$ be a C-set such that for all $f \in \mathbf{C}$, we have $\mathbf{m}(A_f) > 0$. Then there is a C-measurable function $F : \mathbf{C} \to \mathbf{C}$ such that $(f, F(f)) \in A$ for all $f \in C$.*
- *b) Let $B \subset \mathbf{C}^2$ be computable in $\mathbf{S}$ and such that $\mathbf{m}(B_f) > 0$ for all $f \in \mathbf{C}$. Then there is an $\mathbf{S}$-computable function $G : \mathbf{C} \to \mathbf{C}$ such that $(f, G(f)) \in B$ for all $f \in \mathbf{C}$.*

Proof. Item b) is a direct consequence of Corollary 3.11 and Gandy selection. To prove item a), we argue uniformly in the C-code for A. For the sake of notational simplicity, we assume that this code is computable in $\mathbf{S}$. By Theorem 2.14 there is a function $H : \mathbf{C} \to \mathbb{N}^{\mathbb{N}}$ computable in $\mathbf{S}$ such that for each $f \in \mathbf{C}$, $H(f)$ is a Borel-code for a set $H_f \subseteq A_f$ with the same measure. We now consider the following two easy computational facts about H.

- Each value $H(f)$ is computed with a rank uniformly below $\omega_1^{\mathbf{S}}$.
- The functional $\mathbf{S}$ can compute complete Π_1^1-subsets of $\mathbb{N}$ relative to each $H(f)$ in finitely many steps.

Thus, we define $F(f)$ as an element from H_f and this procedure can be done (i) computably in $\mathbf{S}$ and (ii) uniformly at a bounded level. By Theorem 3.2, $F^{-1}(X)$ is a C-set whenever X is a basic neighbourhood. Since the Suslin operator commutes with inverse images, the inverse $F^{-1}(X)$ is a C-set whenever X is a C-set, and F is C-measurable. □

Remark 3.14. In contrast to Borel-sets, it does not suffice to show that the graph of F is a C-set in order to prove that it is C-measurable.

4. Measure-theoretic uniformity and infinite time Turing machines

Using the above results, we show that the computability theory induced by infinite time Turing machines (ITTMs) satisfies similar properties of measure-theoretic uniformity as for the hyperarithmetical sets (Tanaka-Sacks; see Proposition 1.3) and computability relative to $\mathbf{S}$ (Section 3). The key result of this section is already known, namely as [2, Theorem 4.5], where it is proved in the context of random reals and forcing. In this section, we adjust Theorem 3.9 to obtain a generalisation to ITTMs. While not new, our results provide an alternative and more elementary proof that in particular does not make use of forcing.

4.1. Infinite time Turing machines

The notion of ITTM was introduced by Hamkins and Lewis in [7]. An ITTM is like an ordinary Turing machine, with the extra option of continuing working after passing limit ordinals. The notion of ITTM-computability is a strong concept, much stronger than computations relative to **S**. We refer to [7] or recent surveys like [25, §3] for an introduction to ITTMs and their theory.

What is of interest to us is that an ITTM has three tapes: one input tape where the input may be any binary function $f : \mathbb{N} \to \{0,1\}$, a work tape, and an output tape where at each stage ν of the computation there is a function $g_\nu : \mathbb{N} \to \{0,1\}$. An empty input tape also represents the input 0. There are various classes of characteristic functions (or sets) of interest associated with the ITTMs.

Definition 4.1 (ITTM output classes)**.**

i) A function g is *writable* if there is an ITTM that halts on input 0 with g on the output tape when halting.
ii) A function g is *eventually writable* if there is an ITTM with input 0 that after some stage ν will always have g on the output tape. This machine does not have to halt.
iii) A function g is *accidentally writable* if there is an ITTM with input 0 that have g on its output tape at least once.
iv) An ordinal is *clockable* if it is the exact length of an ITTM computation with input 0 that halts

Ordinals will, by definition, belong to these classes if they have codes belonging to them. The least ordinal that is not writable is denoted λ, the least one that is not eventually writable is denoted ζ, and the least one that is not even accidentally writable is denoted Σ. They are all countable ordinals and λ is also the least upper bound of the clockable ordinals. The relativized version of this was proved in [24, Theorem 1.1], and will be applied in the proof of Corollary 4.5. We will also need the following results.

Proposition 4.2 (Hamkins-Lewis, Welch)**.**

a) The above ordinals satisfy $\lambda < \zeta < \Sigma$ and all three are closed under "the next admissible"-operator.
b) The triple (ζ, Σ, λ) is the least triple with $\lambda < \zeta < \Sigma$ in the lexicographical ordering of $\aleph_1^3$ satisfying $L_\lambda \prec_{\Sigma_1} \zeta \prec_{\Sigma_2} \Sigma$. Moreover, this result can be relativised to any input $f \in \mathbf{C}$ on the input tape.

Proof. For the first item, we refer to [7, §8] for λ and ζ, and to [24, Corollary 3.4] for Σ. The second item can be found in [24]. We refer to [25, Theorem 3] for further details. □

The key result on measure-theoretic uniformity for ITTMs is now as follows, to be proved in Section 4.2.

Theorem 4.3 ([2], Theorem 4.5)**.** *The set $\{f \in \mathbf{C} : L_\lambda[f] \prec_{\Sigma_1} L_\zeta[f] \prec_{\Sigma_2} L_\Sigma[f]\}$ has measure one.*

We have the following corollaries, also originally due to Carl-Schlicht in [2].

Corollary 4.4. *For almost all $f \in \mathbf{C}$ we have that all ordinals writable relative to f will be writable, all ordinals eventually writable relative to f will be eventually writable and all ordinals accidentally writable relative to f will be accidentally writable.*

Proof. This follows from the relativized version of [25, Theorem 3]. □

Corollary 4.5. *Let $X \subseteq \mathbf{C}$ be ITTM-semicomputable such that $\mathbf{m}(X) > 0$. Then X contains an element that is ITTM-computable.*

4.2. The proof of Theorem 4.3

The proof of Theorem 4.3, to be found below, requires a few concepts and lemmas, as follows.

First of all, we recall the terms t $(= t^{\mathbf{f}})$ from Section 3.2 and Remark 3.4. When α is sufficiently closed, which our crucial ordinals λ, ζ and Σ are, each element in $L_\alpha[f]$ will be the interpretation t^f of a term t in L_α, and the map $t \mapsto t^f$ is defined by a Δ_1 transfinite recursion uniformly in f. A full proof of the next lemma requires a tediously detailed definition of the terms t and the map $t \mapsto t^f$, but the essence of the proof is trivial.

Lemma 4.6. *Let $\phi(x_0, \dots x_{k-1})$ be a Δ_0-formula and let $t_1, \dots, t_{k-1}$ be countable terms with codes $g_1, \dots, g_{k-1}$ in $\mathbb{N}^{\mathbb{N}}$. Then $\{f \in \mathbf{C} : \phi(t_1^f, \dots, t_{k-1}^f)\}$ is a Borel set with a Borel code uniformly hyperarithmetical in $g_1, \dots, g_{k-1}$.*

Lemma 4.7. *Let $\alpha \in \{\lambda, \zeta, \Sigma\}$ and let $\beta < \alpha$. Let $\vec{t}$ be terms for elements in $L_\beta[f]$ and let $\Phi(\vec{x})$ be any formula where $\vec{x}$ and $\vec{t}$ are of the same length. For $r \in \mathbf{Q}$, the statement $\mathbf{m}(\{f \in \mathbf{C} : L_\beta[f] \models \Phi(\vec{t}^f)\}) \geqslant r$ is Δ_1 over L_α in parameters r, $\vec{t}$ and β.*

Proof. By the choice of α, there will be a term t for $L_\beta[\mathbf{f}]$ in L_α and a code g for this term, also in L_α. By the closure properties of α we see that the relation between sets and codes will be Δ_1 over L_α. We then use Lemma 4.6 and Proposition 1.3, a), and our lemma follows. □

Lemma 4.8. *Let $\vec{t} = (t_0, \dots, t_{k-1})$ be terms in L_λ, let $\phi(\vec{x}, y)$ be a Δ_0-formula and let $0 \leqslant r \leqslant 1$. For each ordinal α, let $E_\alpha := \{f \in \mathbf{C} : (\exists y \in L_\alpha[f])\phi(\vec{t}^f, y)\}$. We then have that $\mathbf{m}(E_\zeta) \geqslant r$ implies $\mathbf{m}(E_\lambda) \geqslant r$.*

Proof. Assume that $\mathbf{m}(E_\zeta) \geqslant r$. It suffices to show that $\mathbf{m}(E_\lambda) \geqslant r - \epsilon$ for each rational r and each $\epsilon > 0$. By our assumptions there will be an ordinal $\beta < \zeta$ such that $\mathbf{m}(E_\beta) \geqslant r - \epsilon$ and by Lemma 4.7 the existence of this β is a Σ_1 statement with parameters from L_λ. Since $L_\lambda \prec_{\Sigma_1} L_\zeta$ we have a $\beta < \lambda$ such that $\mathbf{m}(E_\beta) \geqslant r - \epsilon$ and this proves the lemma. □

We can replace (λ, ζ) with (ζ, Σ) in this lemma, and by direct application of the two versions we obtain the following.

Lemma 4.9. *The set $\{f \in \mathbf{C} : L_\lambda[f] \prec_{\Sigma_1} L_\zeta[f] \prec_{\Sigma_1} L_\Sigma[f]\}$ has measure 1.*

In order to prove Theorem 4.3 it remains to prove that $\{f \in \mathbf{C} : L_\zeta[f] \prec_{\Sigma_2} L_\Sigma[f]\})$ has measure 1, and this follows from our final lemma.

Lemma 4.10 (Relativisation)**.** *Let ϕ be Δ_0, $\vec{t}$ be a list of terms in L_ζ and assume that*

$$\mathbf{m}(\{f \in \mathbf{C} : L_\Sigma[f] \models (\exists x)(\forall y)\phi(\vec{t}^f, x, y)\}) \geqslant r.$$

Then the same holds 'relativised' as follows:

$$\mathbf{m}(\{f \in \mathbf{C} : (\exists x \in L_\zeta[f])(\forall y \in L_\Sigma[f])\phi(\vec{t}^f, x, y)\}) \geqslant r.$$

Proof. We may restrict to rational r. Let $\epsilon > 0$ be rational. Let T be the set of terms for elements in $L_\Sigma[f]$. Then

$$\{f \in \mathbf{C} : L_\Sigma[f] \models \exists x \forall y \phi(\vec{t}^f, x, y)\} = \bigcup_{t \in T} \{f \in \mathbf{C} : L_\Sigma[f] \models \forall y \phi(\vec{t}^f, t^f, y)\}.$$

Since T is countable, there will be a finite set $\{s_1, \dots, s_n\}$ from T such that

$$\mathbf{m}(\{f \in \mathbf{C} : \textstyle\bigvee_{i=1}^{n} (\forall y \in L_\Sigma[f])\phi(\vec{t}^f, s_i^f, y)\}) > r - \epsilon.$$

Then L_Σ will satisfy the following Σ_2-formula:

$$(\exists s_1,\ldots,s_n)(\forall\beta<\Sigma)\big(\mathbf{m}(\{f\in\mathbf{C}:\bigvee_{i=1}^n(L_\beta[f]\models\forall y\phi(\vec{t^f},s_i^f,y))\})>r-\epsilon\big),\tag{1}$$

and thus L_ζ will satisfy (1) as well. We now fix $s_1,\ldots,s_n$ in L_ζ satisfying (1) and note that the following set $\{f\in\mathbf{C}:\bigvee_{i=1}^n(L_\beta[f]\models\forall y\phi(\vec{t^f},s_i^f,y))\}$ is clearly decreasing with increasing β. Thus, we obtain

$$\mathbf{m}(\bigcap_{\beta<\zeta}\{f\in\mathbf{C}:\bigvee_{i=1}^n(L_\beta[f]\models\forall y\phi(\vec{t^f},s_i^f,y))\})\geqslant r-\epsilon.$$

For each $f\in\mathbf{C}$ we have the following equivalence (we only need that ζ is a limit ordinal for this):

$$(\forall\beta<\zeta)\Big[\bigvee_{i=1}^n(L_\beta[f]\models\forall y\phi(\vec{t^f},s_i^f,y))\Big]\qquad\leftrightarrow\qquad\bigvee_{i=1}^n(\forall y\in L_\zeta[f])\phi(\vec{t^f},s_i^f,y),$$

Hence, we have $\mathbf{m}(\{f\in\mathbf{C}:\exists x\in L_\zeta[f]\forall y\in L_\zeta[f]\phi(\vec{t^f},x,y)\})\geqslant r-\epsilon$. Now, if f is such that

$$L_\zeta[f]\prec_{\Sigma_1}L_\Sigma[f]\text{ and }(\exists x\in L_\zeta[f])(\forall y\in L_\zeta[f])\phi(\vec{t^f},x,y),$$

then $\exists x\in L_\zeta[f]\forall y\in L_\Sigma[f]\phi(\vec{t^f},x,y)$, and our claim holds. Since $\epsilon>0$ was arbitrary, the lemma follows. □

5. Conclusions and discussion

The results of this paper illustrate a more general phenomenon, which we formulate as follows.

Almost all subsets of $\mathbb{N}$ are random, and random sets do not have any particular qualities.

We have seen that for almost all $f\in\mathbf{C}$ we have that $\omega_1^{CK}=\omega_1^f$, $\omega_1^{\mathbf{S}}=\omega_1^{\mathbf{S},f}$, $\lambda=\lambda^f$, $\zeta=\zeta^f$ and $\Sigma=\Sigma^f$ using quite elementary methods. By the forcing argument in [22], we even have that if α is a countable ordinal such that $L_\alpha\models\mathsf{ZFC}$ then $L_\alpha[f]\models\mathsf{ZFC}$ for almost all f. The paper [2] by Carl and Schlicht broadens this picture in a precise and more general way, using methods that fall outside the scope of this paper.

Acknowledgements

I am grateful to Sam Sanders for involving me in the project that led to this research. I am also grateful to Alexander S. Kechris and Gerald E. Sacks for their brief comments on a preliminary note on the subject, and to the *Seminar on Mathematical Logic* at the University of Oslo for comments during my presentation there. Sadly, Gerald E. Sacks passed away before the completion of this work.

Furthermore, I am grateful to the two anonymous referees that made valuable comments on the original exposition, including the suggestion by one of them to extend the original results to ITTMs as in Section 4. During the revision of the paper, Sam Sanders made me aware of the paper by Burgess [1], and Philip Welch directed me to the paper by Carl and Schlicht [2].

Finally, I am grateful to Sam Sanders for providing useful feedback on the exposition of this paper.

References

[1] J.P. Burgess, Classical hierarchies from a modern standpoint. Part I. *C*-sets, *Fundam. Math.* **115** (1983), 81 - 95.

[2] M. Carl and P. Schlicht, Randomness via infinite computation and effective descriptive set theory, *J. Symb. Log.* **83** (2) (2018), 766 - 789.

[3] R.O. Gandy, General recursive functionals of finite type and hierarchies of functionals, *Ann. Fac. Sci. Univ. Clermond-Ferrand* **35** (1967), 202 - 242.

[4] A.S. Kechris, *Classical Descriptive Set Theory*, Graduate Texts in Mathematics **156**, Springer-Verlag, New York 1995.
[5] S.C. Kleene, Recursive functionals and quantifiers of finite types I, *Trans. Amer. Math. Soc.* **91** (1959), 1-52.
[6] K. Kunen, *Set theory - An introduction to independence proofs*, Studies in Logic and the Foundations of Mathematics **102**, North-Holland, 1980.
[7] J.D. Hamkins and A. Lewis, Infinite time Turing machines, *J. Symb. Log.* **65** (2) (2000), 567 - 604.
[8] H. Herrlich, *Axiom of Choice*, Lecture Notes in Mathematics **1876**, Springer, 2006.
[9] P.G. Hinman, Hierarchies of effective descriptive set theory, *Trans. Amer. Math. Soc.***142** (1969), 111-140.
[10] J.R. Longley and D. Normann, *Higher-Order Computability*, Springer Verlag 2015.
[11] D. Normann, Functionals of Type 3 as Realisers of Classical Theorems in Analysis, in *Sailing Routes in the World of Computation*, F. Manea, R.G. Miller and D. Nowotka, eds., Springer Lecture Notes in Computer Science **10936** (2018), pp. 318 - 327.
[12] D. Normann, Higher generalizations of the Turing model, in *Turing's Legacy, Developments from Turing's Ideas in Logic*, R. Downey, ed., Lecture Notes in Logic **42**, Cambridge University Press (2014), pp. 397 - 433.
[13] D. Normann and S. Sanders, Nonstandard Analysis, Computability Theory, and their connections, *J. Symb. Log.* **84** (4), (2019) 1422 - 1465, doi: 10.1017/jsl.2019.69.
[14] D. Normann and S. Sanders, The strength of Compactness in Computability Theory and Nonstandard Analysis, *Ann. P. Appl. Log.* **170** (11) (2019), doi: 10.1016/j.apal.2019.05.007.
[15] D. Normann and S. Sanders, On the mathematical and foundational significance of the uncountable, *Jour. Math. Log.* **19** (1), (2019), doi: 10.1142/S0219061319500016.
[16] D. Normann and S. Sanders, Representations in measure theory: between a non-computable rock and hard to prove place, submitted, arXiv-1902.02756.
[17] G.E. Sacks, Measure theoretic uniformity in recursion theory and set theory, *Trans. Amer. Math. Soc.* **124** (1969), 381-420.
[18] G.E. Sacks, *Higher Recursion Theory*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin-Heidelberg 1990.
[19] M.E. Sélivanowski, Sur une class d'ensembles effectifs (ensembles C), *Mat. Sb.* **35 (3-4)** (1928), 379-413.
[20] S.E. Shreve, Probability measures and the *C*-sets of Selivanovskij, *Pac. J. Math.* **79** (1) (1978), 189 - 196.
[21] S.G. Simpson, *Subsystems of second-order arithmetic*, Cambridge University Press (2009).
[22] R. Solovay, A model of set theory in which every set of reals is Lebesgue-measurable, *Ann. of Math.* **92** (1970), 1-56.
[23] H. Tanaka, A basis result for Π^1_1-sets of positive measure, *Comment. Math.Univ. St. Paul* **16** (1968), 115-127.
[24] P.D. Welch, The length of infinite time Turing machine computations, *Bull. London Math. Soc.* **32** (2000), 129-136.
[25] P.D. Welch, Transfinite machine models, in *Turing's Legacy, Developments from Turing's Ideas in Logic*, R. Downey, ed., Lecture Notes in Logic **42**, Cambridge University Press (2014), pp. 493- 529.
[26] P.D. Welch, Higher Type Recursion for Transfinite Machine Theory, in *Computing with Foresight and Industry*, F. Manea, B. Martin B, D. Paulusma and G. Primeiro, eds, Springer Lecture Notes in Computer Science **11558** (2019), 72-83.