Arch. Math. Logic (2009) 48:691–703
DOI 10.1007/s00153-009-0146-2

Infinite time extensions of Kleene's $\mathcal{O}$

Ansten Mørch Klev

Received: 23 October 2008 / Revised: 3 August 2009 / Published online: 12 September 2009

Abstract Using infinite time Turing machines we define two successive extensions of Kleene's $\mathcal{O}$ and characterize both their height and their complexity. Specifically, we first prove that the one extension—which we will call $\mathcal{O}^+$—has height equal to the supremum of the writable ordinals, and that the other extension—which we will call $\mathcal{O}^{++}$—has height equal to the supremum of the eventually writable ordinals. Next we prove that $\mathcal{O}^+$ is Turing computably isomorphic to the halting problem of infinite time Turing computability, and that $\mathcal{O}^{++}$ is Turing computably isomorphic to the halting problem of eventual computability.

Keywords Infinite time Turing machines · Ordinal notation systems

Mathematics Subject Classification (2000) 03D10 · 03D70 · 03F15

1 Introdution

One natural motivation for work on the theory of infinite time Turing machines is the question of how notions and objects from classical computability theory carry over into infinite time. An instance of that question is the motivation for the present paper: we will here study two infinite time analogues of Kleene's $\mathcal{O}$. Our exposition will presuppose some familiarity with these machines and their theory, comparable to what can be got from reading, for instance, Hamkins and Lewis' papers [2] and [3]; material on Kleene's $\mathcal{O}$ can be found in Sacks' book [8].

To distinguish the classical notion of computability from infinite time notions, we will call the former Turing computability; so, for example, instead of saying that A and

A. M. Klev (✉)
Department of Philosophy, McGill University, Montréal, Canada
e-mail: anstenklev@gmail.com

B are computably isomorphic, we will say that they are Turing computably isomorphic. The expression $\{e\}$ denotes the e-th Turing computable function. The following is one way of defining Kleene's $\mathcal{O}$.

Definition 1 (*Kleene [4]*) Let $<_{\mathcal{O}}$ be the least binary transitive relation on $\mathbb{N}$ satisfying the following closure conditions:

1. $1 <_{\mathcal{O}} 2$
2. If $n \in \text{field}(<_{\mathcal{O}})$, then $n <_{\mathcal{O}} 2^n$
3. If $\text{dom}(\{e\}) = \mathbb{N}$, and we have $\{e\}(n) <_{\mathcal{O}} \{e\}(n+1)$ for all $n \in \mathbb{N}$, then $\{e\}(n) <_{\mathcal{O}} 3 \cdot 5^e$ holds for all $n \in \mathbb{N}$

Kleene's $\mathcal{O}$ is the subset of $\mathbb{N}$ coding the relation $<_{\mathcal{O}}$.

We should think of Kleene's $\mathcal{O}$ as a tree constructed from below; at successor stages of its construction each top node of the tree constructed thus far gets another node on top of itself; at limit stages we let the Turing computable functions climb the tree—if a function climbs cofinally in a branch, all natural number codes for that function are put on top of that branch.

In [4] Kleene introduced $\mathcal{O}$ as a system of ordinal notations which is maximal in possessing certain constructive features: it is Turing semi-decidable whether a number is a notation for an ordinal, and in particular whether it is a notation for zero, for a successor, or for a limit; from a notation for a successor, the predecessor can be uniformly computed; there is a Turing computable function $Q\colon \mathbb{N}^2 \rightharpoonup \mathbb{N}$ such that if x is a notation for a limit ordinal α, then there is a sequence of ordinals $\{\alpha_n\}_{n\in\omega}$ whose limit is α and such that $Q(x, n)$ is a notation for α_n.

Now, the only property of the Turing computable functions used in the construction of Kleene's $\mathcal{O}$ is their indexing; thus, in principle, any family $\mathcal{F} = \{F_e\}_{e\in\mathbb{N}}$ of functions can step in for the class of the Turing computable functions in Definition 1.

Definition 2 Suppose $\mathcal{F} = \{F_e\}$ is a family of functions. Let $<_{\mathcal{O}^{\mathcal{F}}}$ be the least binary transitive relation on $\mathbb{N}$ satisfying the following closure conditions:

1. $1 <_{\mathcal{O}^{\mathcal{F}}} 2$
2. If $n \in \text{field}(<_{\mathcal{O}})$, then $n <_{\mathcal{O}^{\mathcal{F}}} 2^n$
3. If $\mathbb{N} \subseteq \text{dom}(F_e)$, and we have $F_e(n) <_{\mathcal{O}^{\mathcal{F}}} F_e(n+1)$ for all $n \in \mathbb{N}$, then $F_e(n) <_{\mathcal{O}^{\mathcal{F}}} 3 \cdot 5^e$ holds for all $n \in \mathbb{N}$

Let $\mathcal{O}^{\mathcal{F}}$ be the subset of $\mathbb{N}$ coding $<_{\mathcal{O}^{\mathcal{F}}}$. We call $\mathcal{O}^{\mathcal{F}}$ the analogue of Kleene's $\mathcal{O}$ for the family $\mathcal{F}$.

The main objects of study in this paper are two such analogues of Kleene's $\mathcal{O}$, one for the family of infinite time computable functions, another for the family of so-called eventually computable functions. These will not in any sense be constructive ordinal notations; our interest in these analogues stems rather from our viewing them as set-theoretical objects.

2 Infinite time computability and extensions of Kleene's $\mathcal{O}$

The infinite time Turing machines of Hamkins and Lewis [2] naturally give rise to two distinct notions of computability, according as to whether we require the machines to halt or to merely stabilize at a constant value.

A partial function on the reals, $F\colon 2^{\mathbb{N}} \rightharpoonup 2^{\mathbb{N}}$, is *infinite time computable* if there is an infinite time Turing machine which on any input $a \in \mathrm{dom}(F)$ halts and outputs $F(a)$ and which does not halt on any input $a \notin \mathrm{dom}(F)$. The function infinite time computed by the Turing machine with code e is denoted by P_e. Not to get lost in symbols, we will also let $P_e(a)$ denote the ω_1-sequence of snapshots corresponding to the infinite time computation of machine e on input a. The expression $P_e(a)\!\downarrow = b$ means that the computation $P_e(a)$ halts with output b.

An infinite time computation may, even if it does not halt, nevertheless stabilize with a constant value on its output tape. Let $\{Q_e\}_{e\in\mathbb{N}}$ be the family of all the infinite time Turing machine programs in which there is no mention of the halting state. The expression $Q_e(a)$ denotes the computation of program Q_e on input a, that is, $Q_e(a)$ is an ω_1-sequence of snapshots according to the program Q_e on input a.

For $\alpha < \omega_1$, let $Q_{e,\alpha}(a)$ denote the content of the output tape of the α-th snapshot in the sequence $Q_e(a)$. If there is an ordinal $\alpha < \omega_1$ and some $b \in 2^{\mathbb{N}}$ such that $Q_{e,\beta}(a) = b$ for all $\beta > \alpha$, then we write $Q_e(a)\!\uparrow = b$; we write $Q_e(a)\!\Uparrow$ if there are no such α and b.

We say that $F\colon 2^{\mathbb{N}} \rightharpoonup 2^{\mathbb{N}}$ is *eventually computable* if there is an e such that $Q_e(a)\!\uparrow = F(a)$ for every $a \in \mathrm{dom}(F)$ and $Q_e(a)\!\Uparrow$ for every $a \notin \mathrm{dom}(F)$. In that case we say that the program Q_e eventually computes F. Abusing notation, we let Q_e also denote the function eventually computed by the program Q_e. It is not hard to see that the class of infinite time computable functions is strictly included in the class of eventually computable functions.

We can now define our two objects of study.

Definition 3 Denote by $\mathcal{O}^+$ the analogue of Kleene's $\mathcal{O}$ for the class of infinite time computable functions.

Definition 4 Denote by $\mathcal{O}^{++}$ the analogue of Kleene's $\mathcal{O}$ for the class of eventually computable functions.

For any family $\mathcal{F}$ and an $n \in \mathrm{field}(\mathcal{O}^{\mathcal{F}})$, let $\mathcal{O}^{\mathcal{F}}\!\restriction n$ code the linear order that results from $<_{\mathcal{O}^{\mathcal{F}}}$ when its field is restricted to $\{k \in \mathrm{field}(\mathcal{O}^{\mathcal{F}}) : k < n\}$. Then $\mathcal{O}^{\mathcal{F}}\!\restriction n$ is a well-order; let $|n|_{\mathcal{O}^{\mathcal{F}}}$ denote its height; define $|\mathcal{O}^{\mathcal{F}}| := \sup\{|n|_{\mathcal{O}^{\mathcal{F}}} : n \in \mathrm{field}(\mathcal{O}^{\mathcal{F}})\}$. Intuitively, $|\mathcal{O}^{\mathcal{F}}|$ is the height of the tree coded by $\mathcal{O}^{\mathcal{F}}$.

Kleene both characterized $|\mathcal{O}|$ as equalling ω_1^{CK}, the supremum of the Turing computable ordinals,[1] and showed that $\mathcal{O}$ is complete for the class of Π^1_1 sets of integers. Here we prove what may be viewed as corresponding results for $\mathcal{O}^+$ and $\mathcal{O}^{++}$: firstly, that $|\mathcal{O}^+|$ equals the supremum of the infinite time computable ordinals, and that $|\mathcal{O}^{++}|$ equals the supremum of the eventually computable ordinals; secondly, that $\mathcal{O}^+$

[1] To be historically correct, Kleene [5] proved that $\omega_1^{\mathrm{CK}} \leq |\mathcal{O}|$, while Markwald [7], and, independently, Spector showed that $|\mathcal{O}| \leq \omega_1^{\mathrm{CK}}$.

is Turing computably isomorphic to the halting problem of infinite time computability and that $\mathcal{O}^{++}$ is Turing computably isomorphic to the halting problem of eventual computability.

In the next section we characterize $|\mathcal{O}^{+}|$ and $|\mathcal{O}^{++}|$, and in Sect. 4 we give complexity measures of $\mathcal{O}^{+}$ and $\mathcal{O}^{++}$.

3 Height

Recall the following notions, introduced in [2].

A real $a \in 2^{\mathbb{N}}$ such that $P_e(0)\downarrow = a$ for some $e \in \mathbb{N}$ is called a *writable real*. An ordinal is a *writable ordinal* if it is coded by a writable real. The supremum of the writable ordinals is called λ. Notice that a real a is writable if and only if it is infinite time computable (as a function on $\mathbb{N}$); the writable ordinals are thus the infinite time computability analogues of the Turing computable ordinals.

A real $a \in 2^{\mathbb{N}}$ is said to be an *eventually writable real* if there is an $e \in \mathbb{N}$ such that $Q_e(0)\uparrow = a$. An ordinal is an *eventually writable ordinal* if it is coded by some eventually writable real. The supremum of the eventually writable ordinals is called ζ. Again, notice that a real a is eventually writable if and only if it is eventually computable (as a function on $\mathbb{N}$); the eventually writable ordinals are thus the eventual computability analogues of the Turing computable ordinals.

A real is said to be an *accidentally writable real* if there is an $e \in \mathbb{N}$ and an ordinal $\alpha < \omega_1$ such that $Q_{e,\alpha}(0) = a$. An ordinal coded by an accidentally writable real is called an *accidentally writable ordinal*; the supremum of the accidentally writable ordinals is named Σ. Hamkins and Lewis proved that $\omega_1^{\mathrm{CK}} < \lambda < \zeta < \Sigma$.

By viewing a set $A \subseteq \mathbb{N}$ and its characteristic function χ_A as the same object, we may refer to sets as being, for instance, eventually writable.

Theorem 1 *1.* $|\mathcal{O}^{+}| = \lambda$
2. $|\mathcal{O}^{++}| = \zeta$

We prove Theorem 1.1 in Sect. 3.1 and Theorem 1.2 in Sect. 3.2.

3.1 The height of $\mathcal{O}^{+}$

In this subsection we prove Theorem 1.1.

Lemma 1 *For any $n \in$ field$(\mathcal{O}^{+})$, the set $\mathcal{O}^{+}\restriction n$ is writable. In consequence, $|\mathcal{O}^{+}| \leq \lambda$.*

Proof The crucial observation is that $\mathcal{O}^{+}$ is eventually writable. This is because an infinite time Turing machine can mimic the inductive definition of $\mathcal{O}^{+}$: while simulating all the computations $\{P_e(k)\}_{e,k\in\mathbb{N}}$ on the scratch tape, it can build $\mathcal{O}^{+}$ on the output tape, writing up limit elements $3 \cdot 5^e$ as soon as it is found that $P_e\restriction\mathbb{N}$ is total and unbounded in a branch of the current approximation to $\mathcal{O}^{+}$. Since we do not know whether $P_e(k)$ will halt for arbitrary $e, k \in \mathbb{N}$, we will not know when to stop this process. Hence, $\mathcal{O}^{+}$ is eventually writable.

Now, when eventually eventually writing $\mathcal{O}^+$ a machine can recognize that $\mathcal{O}^+ \restriction n$ has appeared on the output tape and thus write $\mathcal{O}^+ \restriction n$. □

For the proof of the other direction the idea is, naturally, to embed writable reals coding well-orders into $\mathcal{O}^+$. Our proof takes several cues from Kleene's proof that the computable ordinals embed into the constructive ordinals. The s-m-n Theorem and the Recursion Theorem hold for both infinite time and eventual computability; that fact is used throughout.

Lemma 2 [Kleene] *There is an infinite time computable function* $\dot{+}$ *such that*

1. $m, n \in \text{field}(\mathcal{O}^+)$ *if and only if* $m \dot{+} n \in \text{field}(\mathcal{O}^+)$
2. *for all* $m, n \in \text{field}(\mathcal{O}^+)$ *we have* $|m \dot{+} n|_{\mathcal{O}^+} = |m|_{\mathcal{O}^+} + |n|_{\mathcal{O}^+}$
3. *if* $m, n \in \text{field}(\mathcal{O}^+)$ *and* $n \neq 1$, *then* $m <_{\mathcal{O}^+} m \dot{+} n$

Proof This is just an adaptation of Kleene's definition of $+_{\mathcal{O}}$ in [5]. The function $m \dot{+} n$ can be defined by recursion on n along the well-founded relation $<_{\mathcal{O}^+}$. If n is a successor element of $<_{\mathcal{O}^+}$, then it has the form 2^k for some k, so in that case we want $m \dot{+} n = 2^{m \dot{+} k}$. If n is a limit element of $<_{\mathcal{O}^+}$ then it has the form $3 \cdot 5^e$ for some e, and in that case we want $m \dot{+} n = 3 \cdot 5^d$, where $d \in \mathbb{N}$ codes a program such that $P_d(k) = m \dot{+} P_e(k)$. By use of the Recursion Theorem one can see that there is an infinite time computable function $\dot{+}$ with these required properties. □

This addition operation allows us to bound every writable subset a of $\mathcal{O}^+$ by taking the infinite $\dot{+}$-sum over a. The bounding can be done in a Turing computable manner via a function taking an index for a to an element of $\mathcal{O}^+$ lying above all of a's elements. We use the following nomenclature: say that $n \in \text{field}(\mathcal{O}^{\mathcal{F}})$ *bounds* $a \in 2^{\mathbb{N}}$ *in* $\mathcal{O}^{\mathcal{F}}$ if $a \subseteq \text{field}(\mathcal{O}^{\mathcal{F}})$ and $|m|_{\mathcal{O}^{\mathcal{F}}} < |n|_{\mathcal{O}^{\mathcal{F}}}$ holds for all $m \in a$.

Lemma 3 *There is a Turing computable function* $F' : \mathbb{N} \to \mathbb{N}$ *such that if* $P_e(0)\downarrow = a$ *and* $a \subseteq \text{field}(\mathcal{O}^+)$, *then* $F'(e)$ *bounds* a *in* $\mathcal{O}^+$.

Proof Consider the following procedure. Given $e, n \in \mathbb{N}$, start computing $P_e(0)$; if it is found that $P_e(0)\downarrow = a$, then output the result of applying $\dot{+}$ recursively n times to the string a in the order its elements appear on the tape.

The s-m-n Theorem gives a function $I : \mathbb{N} \to \mathbb{N}$ such that $Q_{I(e)}(n)$ operates according to this procedure; hence we may put $F'(e) := 3 \cdot 5^{I(e)}$. □

Lemma 4 *There is an infinite time computable function* $F : 2^{\mathbb{N}} \rightharpoonup \mathbb{N}$ *with domain the writable reals, such that for all* $a \in \text{dom}(F)$, *if* $a \subseteq \text{field}(\mathcal{O}^+)$, *then* $F(a)$ *bounds* a *in* $\mathcal{O}^+$.

Proof Let F' be the function asserted to exist by the previous lemma. An algorithm which computes F is then: given $a \in 2^{\mathbb{N}}$ search for an $e \in \mathbb{N}$ such that $P_e(0)\downarrow = a$. The first time such an e is found output $F'(e)$ and halt. □

Applying the F of Lemma 4 recursively we can embed every writable real coding a well-order into $\mathcal{O}^+$.

Lemma 5 *There is a Turing computable function $G\colon \mathbb{N} \to \mathbb{N}$ such that if $P_e(0)\downarrow = a$ and $a \in$ WO, then $P_{G(e)}(m) \in \mathrm{field}(\mathcal{O}^+)$ for all $m \in \mathrm{field}(a)$, and*

$$m <_a n \quad \textit{implies} \ |P_{G(e)}(m)|_{\mathcal{O}^+} < |P_{G(e)}(n)|_{\mathcal{O}^+}.$$

Proof Suppose we are given an $e \in \mathbb{N}$ such that $P_e(0)\downarrow = a$ with $a \in$ WO. Uniformly in e we can devise a program which uses the well-order a as a counter for recursive applications of the function F asserted to exist by the previous lemma. For any given $e \in \mathbb{N}$, a general step can be described as follows: let $n \in \mathrm{field}(a)$ be the least element of a not already erased from the counter; erase n and compute $P_{G(e)}(n) = F(\{P_{G(e)}(m) : m <_a n\})$. As all final segments of a are writable, it follows that $\{P_{G(e)}(m) : m <_a n\}$ is writable for each $n \in \mathrm{field}(a)$. □

Proof of Theorem 1.1 Let F and G be the functions of Lemmas 4 and 5, respectively. Suppose $\alpha < \lambda$. There is then an $a \in$ WO coding α and an $e \in \mathbb{N}$ such that $P_e(0)\downarrow = a$. Using P_e and $P_{G(e)}$ we can write a real $c := P_{G(e)}[a]$. We have $c \subseteq \mathrm{field}(\mathcal{O}^+)$ and $\alpha \leq \sup\{|n|_{\mathcal{O}^+} : n \in c\}$, hence also $\alpha \leq |F(c)|_{\mathcal{O}^+} < |\mathcal{O}^+|$. □

3.2 The height of $\mathcal{O}^{++}$

We now turn to the proof of Theorem 1.2, starting with the less messy direction.

Lemma 6 *If $n \in \mathrm{field}(\mathcal{O}^{++})$, then $\mathcal{O}^{++}\restriction n$ is eventually writable. In consequence, $|\mathcal{O}^{++}| \leq \zeta$.*

Proof On one part of the scratch tape we simulate all the computations $\{Q_e(k)\}_{e,k\in\mathbb{N}}$. We stop this simulation at, say, every ω-th step to write an approximation $(\mathcal{O}^{++})^*$ to $\mathcal{O}^{++}$, using the current values $\{Q_{e,\eta}(k)\}_{e,k\in\mathbb{N}}$. Thus, at the η-th step of the simulation of $\{Q_e(k)\}_{e,k}$, this $(\mathcal{O}^{++})^*$ is just the analogue of Kleene's $\mathcal{O}$ for the family $\{Q_{e,\eta}\}_{e\in\mathbb{N}}$. If we find that $n \in \mathrm{field}((\mathcal{O}^{++})^*)$, then we write $(\mathcal{O}^{++})^*\restriction n$ on the output tape, unless this real is already written there.

Now, if $n \in \mathrm{field}(\mathcal{O}^{++})$, then all the computations $\{Q_e(k)\}_{k\in\mathbb{N}}$ for $3\cdot 5^e <_{\mathcal{O}^{++}} n$ will finally stabilize; thus, from some point onwards we will have $(\mathcal{O}^{++})^*\restriction n = \mathcal{O}^{++}\restriction n$ for any approximation $(\mathcal{O}^{++})^*$. Thus, we will be eventually writing $\mathcal{O}^{++}\restriction n$. □

The proof of Lemma 6 does not adopt to show that $\mathcal{O}^{++}$ is eventually writable: in general, it will not be the case that $(\mathcal{O}^{++})^* = \mathcal{O}^{++}$.

To get any further in eventual computability—in particular to prove the other direction of Theorem 1.2—we need the following important results of Welch [12].

Lemma 7 (Welch [12]) *There is an infinite time Turing machine which from some point onwards only outputs ordinals greater than ζ, and which outputs arbitrarily large accidentally writable ordinals.*

For ease of reference we will call this machine the ordinal production machine. Using this machine, Welch proved

Welch's Lemma *The ζ-snapshot of an infinite time Turing machine computation is equal to its Σ-snapshot; moreover, the computation never escapes the loop in which it finds itself at stage ζ.*[2]

Lemma 8 (Welch [12]) *If a computation $Q_e(0)$ stabilizes, then it stabilizes before stage ζ.*

We can now prove the lemmas needed to complete the proof of Theorem 1.

Lemma 9 (Kleene) *There is an eventually computable function $\hat{+}$ such that*

1. $m, n \in \text{field}(\mathcal{O}^{++})$ *if and only if* $m \mathbin{\hat{+}} n \in \text{field}(\mathcal{O}^{++})$
2. *for all* $m, n \in \text{field}(\mathcal{O}^{++})$ *we have* $|m \mathbin{\hat{+}} n|_{\mathcal{O}^{++}} = |m|_{\mathcal{O}^{++}} + |n|_{\mathcal{O}^{++}}$
3. *if* $m, n \in \text{field}(\mathcal{O}^{++})$ *and* $n \neq 1$, *then* $m <_{\mathcal{O}^{++}} m \mathbin{\hat{+}} n$

Proof The function $\hat{+}$ is defined by recursion along $<_{\mathcal{O}^{++}}$. One uses the ordinal production machine and Lemma 8 to show existence of the auxiliary functions needed in this recursive definition. □

Lemma 10 *There is a Turing computable function $F' \colon \mathbb{N} \to \mathbb{N}$ such that if $Q_e(0)\!\uparrow = a$ and $a \subseteq \text{field}(\mathcal{O}^{++})$, then $F'(e)$ bounds a in $\mathcal{O}^{++}$.*

Proof As a preliminary, fix a $d \in \mathbb{N}$ such that the program Q_d eventually computes the function $\hat{+}$. We intend to use the s-m-n Theorem on the following recursive algorithm. Given input $(e, n) \in \mathbb{N}^2$. Let α be given by the ordinal production machine; first compute $a^* := Q_{e,\alpha}(0)$, then apply $Q_{d,\alpha}$ recursively n times to the string a^* in the order its elements appear on the scratch tape and write the result on the output tape (unless it is already written there). Then wait for a new ordinal from the ordinal production machine.

The s-m-n Theorem gives a Turing computable function $I \colon \mathbb{N} \to \mathbb{N}$ such that $Q_{I(e)}(n)$ executes this algorithm. Suppose that $Q_e(0)\!\uparrow = a$. By Lemma 8 the computation $Q_e(0)$ will stabilize before stage ζ, as will any computation $m \mathbin{\hat{+}} n$ for $m, n \in \text{field}(\mathcal{O}^{++})$. Hence, from some point onwards the computation $Q_{I(e)}(n)$ recursively $\hat{+}$-adds n times the elements of a in the order they appear on the scratch tape. Hence, we can let $F'(e) := 3 \cdot 5^{I(e)}$. □

This proof shows how we lift a lemma from infinite time computability to eventual computability: compute approximations along ordinals given by the ordinal production machine; Lemma 8 ensures that from some point onwards the approximation computed will in fact be the true value.

Applying this technique and using the algorithms described in the proofs of Lemmas 4 and 5, the following two lemmas are readily proved.

Lemma 11 *There is an eventually computable function $F \colon 2^{\mathbb{N}} \rightharpoonup \mathbb{N}$ such that for any eventually writable a, if $a \subseteq \text{field}(\mathcal{O}^{++})$, then $F(a)$ bounds a in $\mathcal{O}^{++}$.*

[2] Welch's proof in [12] of this lemma has a gap. For what the author considers to be a gap-free proof, see [6].

Lemma 12 *There is a Turing computable function $G\colon \mathbb{N} \to \mathbb{N}$ such that if $Q_e(0)\uparrow = a$ and $a \in \mathrm{WO}$, then $Q_{G(e)}(m) \in \mathrm{field}(\mathcal{O}^{++})$ for all $m \in \mathrm{field}(a)$, and*

$$m <_a n \quad \textit{implies} \ |Q_{G(e)}(m)|_{\mathcal{O}^{++}} < |Q_{G(e)}(n)|_{\mathcal{O}^{++}}.$$

The proof of Theorem 1.2 is now completed just like the proof of Theorem 1.1.

The finite time notion of eventual computability is captured by that of Turing computable approximations: a Turing computable sequence $\{F_e\}_{e\in\mathbb{N}}$ of partial functions on $\mathbb{N}$ is an approximation to $F\colon \mathbb{N} \rightharpoonup \mathbb{N}$ if for every $n \in \mathrm{dom}(F)$ there is an $m \in \mathbb{N}$ such that $F_e(n) \simeq F(n)$ holds for all $e \geq m$, whereas for all $n \notin \mathrm{dom}(F)$ there is no such m.

According to Shoenfield's Limit Lemma a function has a Turing computable approximation if and only if it is Turing computable with $0'$ as an oracle; consequently, finite time eventual computability is just $0'$-Turing computability. Let $\mathcal{O}^{0'}$ be the analogue of Kleene's $\mathcal{O}$ for the family of $0'$-Turing computable functions. From a theorem of Spector [9] it follows that $|\mathcal{O}^{0'}| = |\mathcal{O}|$. Thus, in contrast to what happens in the infinite time setting, moving to eventual computability in the classical setting does not increase the height of the relevant analogue of Kleene's $\mathcal{O}$.

4 Complexity

In this section we prove that $\mathcal{O}^+$ is Turing computably isomorphic to the halting problem of infinite time computability, and that $\mathcal{O}^{++}$ is Turing computably isomorphic to the halting problem of eventual computability. As a consequence of these results we can situate $\mathcal{O}^+$ and $\mathcal{O}^{++}$ within the infinite time Turing degrees.

As is described in [2], infinite time Turing machines can compute with a set $A \subseteq 2^{\mathbb{N}}$ as an oracle; this gives rise to the partial order of infinite time Turing reducibility $\leq_\infty$ on $2^{\mathbb{N}}$, and derivatively to the infinite time Turing degrees.

An infinite time Turing degree $[A]_{\equiv_\infty}$ is said to be eventually, respectively accidentally, writable if there is an eventually, respectively accidentally, writable real $a \in [A]_{\equiv_\infty}$. Hamkins, Lewis, and Welch proved that the eventually writable degrees ordered by $\leq_\infty$ forms a well-order of height ζ, and that the accidentally writable degrees ordered by $\leq_\infty$ forms a well-order of height $\zeta + 1$.

The halting problem for infinite time computability is the set

$$h := \{e \in \mathbb{N} \ : \ P_e(0)\downarrow\}.$$

It is readily seen that $0 <_\infty h$; Hamkins and Lewis [3] proved that there is no real a such that $0 <_\infty a <_\infty h$; hence, h is of the least non-zero eventually writable degree.

The halting problem for eventual computability—more rightly called the stabilization problem—is the set

$$s := \{e \in \mathbb{N} \ : \ Q_e(0)\uparrow\}.$$

The set s is not eventually writable, but it is accidentally writable; hence, it is of the maximal accidentally writable degree.

Theorem 2 1. $\mathcal{O}^+ \equiv_1 h$
2. $\mathcal{O}^{++} \equiv_1 s$

Proof 2. To see that $h \leq_1 \mathcal{O}^+$ consider, for instance,

$$G(e,n) := \begin{cases} \overbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}^{n\text{ times}} & \text{if } e \in h \\ \text{undefined} & \text{if } e \notin h \end{cases}$$

This G is infinite time computable, so the s-m-n Theorem yields an injective Turing computable $F\colon \mathbb{N} \to \mathbb{N}$ such that $e \in h \Leftrightarrow \langle 1, 3 \cdot 5^{F(e)} \rangle \in \mathcal{O}^+$.

For the other direction, we notice that $\mathcal{O}^+$ is infinite time semi-decidable: this follows from the fact that in the eventual writing of $\mathcal{O}^+$, once something is written on the output tape it is never again erased. It is easy to see that any semi-decidable set is 1-reducible to h.
2. The proof that $s \leq_1 \mathcal{O}^{++}$ is just a rewording of the proof that $h \leq_1 \mathcal{O}^+$, so we head directly for the other direction.

Say that a computation $Q_e(a)$ is locally stable at a limit ordinal α if the sequence $\{Q_{e,\beta}(a)\}_{\beta<\alpha}$ is eventually constant. Now define

$$Q_{e,<\alpha}(a) := \begin{cases} Q_{e,\alpha}(a) & \text{if } Q_e(a) \text{ is locally stable at } \alpha \\ \text{undefined} & \text{otherwise.} \end{cases}$$

For an ordinal $\alpha < \omega_1$, let $(\mathcal{O}^{++})_\alpha$ be the analogue of Kleene's $\mathcal{O}$ for the class of functions $\{Q_{e,<\alpha}\}_{e\in\mathbb{N}}$. Uniformly in $n \in \mathbb{N}$ we devise the following recursive algorithm: given an ordinal α from the ordinal production machine, write $(\mathcal{O}^{++})_\alpha$ on the scratch tape, only using the scratch tape and the input tape. Then flash a flag on the output tape if, and only if, $n \notin (\mathcal{O}^{++})_\alpha$ and wait for a new ordinal.

Let A be the set of ordinals that the ordinal production machine outputs after stage ζ. Then A is an unbounded subset of the interval $[\zeta, \Sigma)$. The s-m-n Theorem applied to the algorithm of the previous paragraph thus yields a function $F\colon \mathbb{N} \to \mathbb{N}$ such that

$$n \in \bigcap_{\beta\in A} (\mathcal{O}^{++})_\beta \Leftrightarrow F(n) \in s.$$

Using Lemma 8 and the fact that A is unbounded in $[\zeta, \Sigma)$ one can see that

$$\bigcap_{\beta\in A} (\mathcal{O}^{++})_\beta = \mathcal{O}^{++}.$$

Hence, F witnesses that $\mathcal{O}^{++} \leq_1 s$. □

Corollary 1 *1.* $\mathcal{O}^+$ *is of the least non-zero accidentally writable degree.*
2. $\mathcal{O}^{++}$ *is of the maximal accidentally writable degree.*

5 A close relationship between $\mathcal{O}$ and $\mathcal{O}^{++}$

We conclude with some observations indicating that objects and notions of classical hyperarithmetic theory are closely related to objects and notions of eventual computability. As a preliminary we recall a few definitions.

An operator $\Gamma\colon \wp(\mathbb{N}) \to \wp(\mathbb{N})$ is said to be inductive if it is either monotone or if $A \subseteq \Gamma(A)$ for all $A \subseteq \mathbb{N}$; the operator Γ is said to be arithmetical if it has an arithmetical definition. Call a set $A \subseteq \mathbb{N}$ arithmetically inductive if it is m-reducible to the least fixed-point of an arithmetical, inductive operator. Kleene proved that a set is arithmetically inductive if and only if it is Π^1_1. Below we shall stick to the latter designation for these sets.

In [1], Burgess introduced the notion of an arithmetically quasi-inductive set. For η a limit ordinal and $\{A_\alpha\}_{\alpha<\eta}$ a sequence of sets, define

$$\liminf\{A_\alpha\}_{\alpha<\eta} := \bigcup_{\alpha<\eta} \bigcap_{\alpha<\beta<\eta} A_\beta.$$

Given an operator $\Gamma\colon \wp(\mathbb{N}) \to \wp(\mathbb{N})$, define

$$\begin{aligned} \Gamma^0 &:= \emptyset \\ \Gamma^{\alpha+1} &:= \Gamma(\Gamma^\alpha) \\ \Gamma^\eta &:= \liminf\{\Gamma^\alpha\}_{\alpha<\eta}, \text{ for limit ordinals } \eta. \end{aligned}$$

Denote by $\Gamma^{<\infty}$ the set $\bigcup_\alpha \bigcap_{\alpha<\beta} \Gamma^\beta$. A set $A \subseteq \mathbb{N}$ is said to be *arithmetically quasi-inductive* if there is an operator Γ with an arithmetical definition such that A is m-reducible to $\Gamma^{<\infty}$.

A set A whose transitive closure is countable can be coded by a set of natural numbers: let $\{v_n\}_{n\in\mathbb{N}}$ be an enumeration of $\mathrm{trcl}(A)$, and let R_A be the unique relation on $\mathbb{N}$ such that

$$mR_An \Leftrightarrow v_m \in v_n \in \mathrm{trcl}(A)$$

holds for all $m, n \in \mathbb{N}$. If $B \subseteq \mathbb{N}$ codes this relation R_A, then B is also a code for A.

For any two sets A, B, say that B is $\Sigma_n(A)$ if there is a Σ_n formula φ in the language of set theory such that

$$v \in B \Leftrightarrow (A, \in) \models \varphi[v].$$

Let $\mathbf{L}_\alpha$ be the α-th level of Gödel's constructible hierarchy. The following Observation 1 is the mother of the rest of our Observations.

Observation 1 (Folklore, Burgess, Welch)

1. $\Sigma_1(\mathbf{L}_{\omega_1^{\mathrm{CK}}})$ *is the set of those sets which have a* Π^1_1 *code.*
2. $\Sigma_2(\mathbf{L}_\zeta)$ *is the set of those sets which have an arithmetically quasi-inductive code.*

Proof 1. is a basic result of admissible set theory; it goes back to the work of Kripke and Platek.

2. is Burgess' Theorem 14.1 in [1] combined with Welch's Theorem 2.1 in [11]. □

Observation 2 (Kleene, Welch)

1. *Kleene's $\mathcal{O}$ is 1-complete with respect to the class of Π^1_1 sets.*
2. *$\mathcal{O}^{++}$ is 1-complete with respect to the class of arithmetically quasi-inductive sets.*

Proof 1. is Theorem 1 in Kleene [5].

2. follows from Observation 1.2 combined with Welch's Theorem 2.6 in [11] and with Theorem 2.2 above. □

Given an operator $\Gamma : \wp(\mathbb{N}) \to \wp(\mathbb{N})$, denote by $\|\Gamma\|$ the least ordinal α such that $\Gamma^\alpha = \Gamma^{<\infty}$; for inductive Γ, $\|\Gamma\|$ is often called the closure ordinal of Γ.

Observation 3 (Spector, Burgess, Welch)

1. $\sup\{\|\Gamma\| : \Gamma$ *is arithmetical and inductive*$\} = \omega_1^{\mathrm{CK}}$.
2. $\sup\{\|\Gamma\| : \Gamma$ *is arithmetical*$\} = \zeta$.

Proof 1. is Theorem 3 in Spector [10].

2. The "Upper bound" part of the proof of Theorem 14.1 in Burgess [1] gives ζ as upper bound for $\sup\{\|\Gamma\| : \Gamma \text{is arithmetical}\}$. There is a set of the form $\Gamma^{<\infty}$ which is complete with respect to the arithmetically quasi-inductive sets. From Observation 1.2 it follows that, for the operator Γ generating this set we must have $\|\Gamma\| \geq \zeta$. □

The class of hyperarithmetic sets is obtained by iterating the Turing jump along $\mathcal{O}$ and closing under Turing reducibility $\leq_{\mathrm{T}}$. Let us look at a similar construction for infinite time computability. First recall that the weak jump of infinite time computabilty $\cdot^\nabla$ is defined by $A^\nabla := A \oplus h^A$, where $\oplus$ is some primitive recursive disjoint union operation. Now define a family $\{E_n\}_{n\in\mathrm{field}(\mathcal{O}^{++})}$ of subsets of $\mathbb{N}$, by iteration of the weak jump along $\mathcal{O}^{++}$, as follows.

$$\begin{array}{lll} E_1 & := \emptyset & \\ E_{2^n} & := (E_n)^\nabla & \text{for } n \in \mathrm{field}(\mathcal{O}^{++}) \\ E_{3\cdot 5^e} & := \{\langle m, Q_e(n)\rangle : m \in E_{Q_e(n)}\} & \text{for } 3\cdot 5^e \in \mathrm{field}(\mathcal{O}^{++}) \end{array}$$

We now observe that if we close the family $\{E_n\}_{n\in\mathrm{field}(\mathcal{O}^{++})}$ under infinite time Turing reducibility, $\leq_\infty$, then we obtain the class of eventually writable sets.

Observation 4 *A set $A \subseteq \mathbb{N}$ is eventually writable if and only if there is some $n \in$* $\mathrm{field}(\mathcal{O}^{++})$ *such that $A \leq_\infty E_n$.*

Proof For the *only if* direction it suffices to prove that every set E_n is eventually writable; we prove that by induction on $<_{\mathcal{O}^{++}}$. The weak jump of any eventually writable real is again eventually writable, a fact which takes care of the successor step. For the limit step, we recall that Hamkins and Lewis [2] proved that if z is an eventually writable real coding an ordinal, and a is any eventually writable real, then $a^{\nabla_z^{(\alpha)}}$—the α-th $\cdot^{\nabla}$-iterate of a, where the iteration is done along the well-order coded by z—is also eventually writable. Using their algorithm with some extra elaboration on how to deal with limit levels, one can show that $E_{3.5^e}$ is eventually writable.

For the *if* direction, note that if b is a path through $\mathcal{O}^{++}$ of height ζ, then the set $\{[E_n]_{\equiv_\infty} : n \in \text{field}(b)\}$ ordered by $\leq_\infty$ is a well-order of height ζ. Hence, the existence of an eventually writable real not reducible to any E_n would contradict the fact that the eventually writable degrees ordered by $\leq_\infty$ is a well-order of height ζ. □

The relationship between hyperarithmetic and eventually writable is further strengthened by

Observation 5 (Folklore, Burgess, Welch)

1. $\mathbf{L}_{\omega_1^{\text{CK}}}$ *is the set of those sets which have a hyperarithmetical code.*
2. $\mathbf{L}_\zeta$ *is the set of those sets which have an eventually writable code.*

Proof 1. is another basic result of admissible set theory.
2. is Corollary 3.1 in [12]. □

We can sum up Observations 1–5 in the following diagram.

$\Sigma_1(\mathbf{L}_{\omega_1^{\text{CK}}})$	Π_1^1	Hyper-arithmetic	$\mathcal{O}$	ω_1^{CK}
$\Sigma_2(\mathbf{L}_\zeta)$	Arithmetically Quasi-Inductive	Eventually Writable	$\mathcal{O}^{++}$	ζ

Acknowledgments The results presented in this paper are taken from the author's MSc thesis [6]. The author wishes to express his sincerest thanks to Joel David Hamkins for very kind and helpful supervision.

References

1. Burgess, J.P.: The truth is never simple. J. Symb. Log. **51**, 663–681 (1986)
2. Hamkins, J.D., Lewis, A.: Infinite time Turing machines. J. Symb. Log. **65**, 567–604 (2000)
3. Hamkins, J.D., Lewis, A.: Post's problem for supertasks has both positive and negative solutions. Arch. Math. Log. **41**, 507–523 (2002)
4. Kleene, S.C.: On notation for ordinal numbers. J. Symb. Log. **3**, 150–155 (1938)
5. Kleene, S.C.: On the forms of predicates in the theory of constructive ordinals (second paper). Am. J. Math. **77**, 405–428 (1955)
6. Klev, A.M.: Extending Kleene's O using infinite time Turing machines. Master's thesis, Universiteit van Amsterdam (2007)
7. Markwald, W.: Zur Theorie der konstruktive Wohlordnungen. Math. Ann. **127**, 135–140 (1954)
8. Sacks, G.E.: Higher Recursion Theory. Springer, Heidelberg (1990)
9. Spector, C.: Constructive well-orderings. J. Symb. Log. **20**, 151–163 (1955)

10. Spector, C.: Inductively defined sets of natural numbers. In: Infinitistic Methods, pp. 97–102. Pergamon Press, New York (1961)
11. Welch, P.D.: Eventually infinite time degrees: infinite time decidable reals. J. Symb. Log. **65**, 1193–1203 (2000)
12. Welch, P.D.: The length of infinite time Turing machine computations. Bull. London Math. Soc. **32**, 129–136 (2000)