UNIVERSITY OF BRISTOL

University of Bristol - Explore Bristol Research
General rights

REACHABILITY FOR INFINITE TIME TURING MACHINES WITH LONG TAPES

MERLIN CARL, BENJAMIN RIN, AND PHILIPP SCHLICHT

Fachbereich Mathematik und Statistik, University of Konstanz, 78457 Konstanz, Germany, and Europa-Universität Flensburg, Institut für mathematische, naturwissenschaftliche und technische Bildung, Abteilung für Mathematik und ihre Didaktik, Auf dem Campus 1b, 24943 Flensburg, Germany
e-mail address: merlin.carl@uni-konstanz.de

Departement Filosofie en Religiewetenschap, Utrecht University, Janskerkhof 13, 3512 BL, Utrecht, The Netherlands
e-mail address: b.g.rin@uu.nl

Department of Computer Science, The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand, and School of Mathematics, University of Bristol, Fry Building. Woodland Road, Bristol, BS8 1UG, UK
e-mail address: philipp.schlicht@bristol.ac.uk

ABSTRACT. Infinite time Turing machine models with tape length α, denoted T_α, strengthen the machines of Hamkins and Kidder with tape length ω. A new phenomenon is that for some countable ordinals α, some cells cannot be halting positions of T_α given trivial input. The main open question in a paper of Rin from 2014 asks about the size of the least such ordinal δ.

We answer this by providing various characterizations. For instance, δ is the least ordinal with any of the following properties:
- For some $\xi < \alpha$, there is a T_ξ-writable but not T_α-writable subset of ω.
- There is a gap in the T_α-writable ordinals.
- α is uncountable in L_{λ_α}.

Here λ_α denotes the supremum of T_α-writable ordinals, i.e. those with a T_α-writable code of length α.

We further use the above characterizations, and an analogue to Welch's submodel characterization of the ordinals λ, ζ and Σ, to show that δ is large in the sense that it is a closure point of the function $\alpha \mapsto \Sigma_\alpha$, where Σ_α denotes the supremum of the T_α-accidentally writable ordinals.

1. Introduction

1.1. **Motivation.** The infinite time Turing machines introduced by Hamkins and Kidder (see [HL00]) are, roughly, Turing machines with a standard tape that run for transfinite ordinal time. One of the main motivations for studying these machines is the fact that they model a class of functions that is closely related to Σ_1^1 and Π_1^1 sets in descriptive set theory. Soon after they were introduced, several variations were proposed, for instance with an arbitrary ordinal as tape length [Koe09], or an exponentially closed ordinal as tape length and time bound [Koe05, KS09].

More recently, the second author studied machines with an arbitrary ordinal as tape length, but no ordinal bound on the running time [Rin14]. These machines are natural generalizations of infinite time Turing machines for tapes of length α, and are thus called α-ITTM*s*. They do not include ordinal parameters, which are present in most other models [Koe05, KS09, COW18]. For a given ordinal $\alpha \in \mathrm{On}$ and program $e \in \mathbb{N}$ there exists a unique machine, $T_\alpha[e]$. We will frequently identify a program with the corresponding machine. The set $\{T_\alpha[e] \mid e \in \mathbb{N}\}$ of all machines with tape length α is called the *device* or *machine model* T_α. It is known that the computability strength[1] of T_α can increase with α, though it remains equal when the increase in α is small. When α itself is not too large, increasing its size necessarily makes the computational strength greater or equal. However, it turns out that for sufficiently large tapes, the machine models' computability strengths are not always commensurable: there exist pairs of countable ordinals such that two devices with these tape lengths can each compute functions that the other one can't [Rin14, Proposition 2.9]. Thus α-ITTMs fail to be linearly ordered by computational strength. What is responsible for this phenomenon is that, in spite of the lack of ordinal parameters, a machine can use its tape length α to perform computations that rely on the exact size of α—an ability which, *because* of the lack of parameters, can permit two differently sized machines to exploit their tape lengths in ways the other cannot. This phenomenon clearly does not occur for models that include ordinal parameters (as in [COW18]), since one can then always simulate a shorter tape on a longer one (cf. [Rin14, Proposition 2.1]). This is because one can easily move the head up to cell ξ and halt there whenever one is allowed to mark the ξth cell, as is possible when computing with parameter ξ. Indeed, it is straightforward to see that an ordinal parameter $\xi < \alpha$ is equivalent to an oracle that allows a machine T_α to emulate the computational behavior of smaller machine T_ξ.

In the present article, we are interested in the writability strength of α-ITTMs without parameters, i.e., in the set of possible outputs of such a machine at the time when it halts. One of the tools in [Rin14] to help classify the machines in question is the connection between computability strength and ordinals α such that T_α cannot reach[2] all of its cells. In particular, let δ denote the least such ordinal. Rin already showed that δ equals the least ordinal γ such that the computability strength of T_γ (as defined above) is incomparable with that of some machine with a shorter tape [Rin14, Proposition 2.1]. The main question left open was to identify δ.[3] We answer this by giving various characterizations of δ in the

---

[1]Computability strength in the present sense is a relative notion: given $\alpha, \beta \in \mathrm{On}$, we write $T_\alpha \preceq T_\beta$ when the set of functions $f \colon 2^{\min(\alpha,\beta)} \to 2^{\min(\alpha,\beta)}$ computable by T_α (that is to say, computable by $T_\alpha[e]$ for some $e \in \mathbb{N}$) is a subset of the set of such functions computable by T_β.

[2]An ordinal μ is defined to be *reachable* by T_α when there exists a program P such that T_α running P on trivial input (input $\vec{0}$) halts with the final head position located at cell μ.

[3]See the discussion after [Rin14, Proposition 2.9].

next theorem. Some of them are formulated via α-ITTMs. The remaining ones are stated in term of constructible set theory and resemble fine-structural properties of the constructible universe L, where first-order definability is replaced with variants of infinite time writability.

Theorem 1.1. *The next conditions for $\alpha > \omega$ hold for the first time at the same ordinal:*

(a) *Not every cell is T_α-reachable (T_α-eventually reachable).*
(b) *There is a gap in the T_α-writable (T_α-eventually writable) ordinals.*
(c) *For some μ, ν with $\omega \leq \mu \leq \nu < \alpha$, there is a T_ν-writable but not T_α-writable subset of μ.*
(d) $\lambda_\alpha < \hat{\lambda}_\alpha$.
(e) $\zeta_\alpha < \hat{\zeta}_\alpha$.
(f) α *is uncountable in L_{λ_α}.*
(g) α *is regular in L_{λ_α}.*
(h) α *is a cardinal in L_{λ_α}.*
(i) *As (f), (g) or (h), but for $\hat{\lambda}_\alpha$, ζ_α, $\hat{\zeta}_\alpha$ or $\hat{\Sigma}_\alpha = \Sigma_\alpha$.*

In (d)–(i), λ_α and ζ_α denote the versions of λ and ζ for T_α without ordinal parameters, while $\hat{\lambda}_\alpha$ and $\hat{\zeta}_\alpha$ denote those with parameters.

Let λ, ζ and Σ denote the suprema of writable, eventually writable and accidentally writable ordinals for ITTMs. The previous characterizations imply that $\Sigma < \delta$. By (f), we can obtain triples (μ, ν, ξ) with $\mu < \nu < \xi < \delta$ and $L_\mu \prec L_\nu \prec L_\xi$ by forming countable elementary substructures of L_δ in $L_{\hat{\lambda}_\delta}$. Thus $\Sigma < \delta$ holds by Welch's submodel characterization of λ, ζ and Σ (cf. [Wel09, Theorem 30 & Corollary 32]).

The next result (cf. Theorem 3.17) is proved via a variant of the submodel characterization for α-ITTMs (cf. Theorem 2.9).

Theorem 1.2. $\Sigma_\xi < \delta$ *for all $\xi < \delta$.*[4]

The structure of the paper is as follows. Section 1.2 contains some background on α-ITTMs. In Section 2, we prove some auxiliary results about writable and clockable ordinals. These are used in Sections 3.1-3.4 to prove the characterizations of δ stated in Theorem 1.1 and the lower bounds for δ in Theorem 1.2.

For reading this paper, we assume that the reader is familiar with infinite time Turing machines, basic facts about Gödel's constructible universe and the proof of Welch's submodel characterization of λ, ζ and Σ from [Wel09, Theorem 30 & Corollary 32]. The latter is used in the proof of Theorem 2.9.

1.2. **The setting.** We briefly introduce the main notions and results related to α-ITTMs and refer the reader to [HL00, KS09, Rin14, Wel09] for details. We always assume that the tape length α is infinite and multiplicatively closed, i.e. $\mu \cdot \nu < \alpha$ for all $\mu, \nu < \alpha$. It is easy to see that this is equivalent to closure under Gödel pairing. An α-ITTM has three tapes of length α for input, working space and output and each cell can contain 0 or 1. Programs for T_α are just regular Turing machine programs. The machine can process a subset of α by representing it on the tapes via its characteristic function. Thus we will freely identify a set with its characteristic function. The *input tape* carries the subset of α that is given to the machine at the start of the computation, and its content is never changed, while the

---

[4]This strengthens the result from [Rin14] that $\zeta < \delta$ and an unpublished result by Robert Lubarsky that $\Sigma < \delta$.

results of a computation are written on the *output tape*. The remaining tape is a *work tape*. Furthermore, each tape has a head for reading and writing, all of which move independently of each other. It is easy to see that one can equivalently allow any finite number or in fact α many work tapes (using that α is multiplicatively closed). Moreover, the model from [HL00] with a single head can simulate our model and is thus equivalent.

The machine $T_\alpha[e]$ runs along an ordinal time axis. At successor times, the configuration of the machine is obtained from the preceding one, as usual for a Turing machine, with the extra convention that a head is reset to position 0 if it is moved to the left from a limit position. At limit times, the content of each cell as well as the head positions are determined as the inferior limits of the sequences of earlier contents of that cell and earlier head positions; if for some head the inferior limit of the sequence of earlier positions is α, then it is reset to 0.

A $\hat{T}_\alpha$-*program* computes relative to a finite parameter subset p of α. This is given to the program by writing the characteristic function of p to one of the work tapes before the computation starts. As we will only be concerned with the case that α is closed under the Gödel pairing function and the function's restriction to α is easily seen to be computable by an α-ITTM, we can assume that parameters are single ordinals below α.

We now turn to various notions of writability from [HL00]. A subset x of α is called T_α-*writable* if there is a T_α-program P that halts with x on the output tape when the initial input is empty, i.e., all cells contain 0. Moreover, x is called *eventually T_α-writable* if there is a T_α-program P such that the output tape will have the contents x and never change again from some point on, if the initial input is empty, although the contents of other tapes might change. Finally, x is called *accidentally T_α-writable* if there is an T_α-program such that x appears as the content of the output tape at some time of the computation with empty input. Analogous to [HL00, Theorem 3.8], these three notions of writability are distinct (see Lemma 2.4).

As for ITTMs, an ordinal is called T_α-*clockable* if it is the halting time of a T_α-program with input $\vec{0}$.

The above notions are defined for $\hat{T}_\alpha$ in an analogous way.

As for Turing machines, there is a *universal T_α-program* U_α that simulates all computations with empty input. This can be obtained by dividing the work and output tapes into infinitely many tapes of the same length. Note that any $\hat{T}_\alpha$-program can be simulated by a T_α-program that considers all possible parameters. Thus U_α accidentally writes every $\hat{T}_\alpha$-accidentally writable subset of α.

To compare the writability strength of these machines for different ordinals, we often consider T_α-writable subsets x of some ordinal $\xi \leq \alpha$. Naively, we could just write x to the initial segment of length ξ of the output tape and leave the rest empty, but then we could no longer distinguish between x as a subset of ξ and as a subset of α. Therefore, we introduce the following notion. A subset x of ξ is called T_α-*writable as a subset of* ξ if there is a T_α-program with empty input that halts with the characteristic function of x on the output tape, and if $\xi < \alpha$, then the head is in position ξ at the end of the computation. Similarly, we call x *eventually writable as a subset of* ξ if the contents of the output tape eventually stabilizes at the characteristic function of x and the head on the output tape eventually stabilizes at ξ. For any $\xi \leq \min\{\alpha, \beta\}$, we say that T_α has strictly greater writability strength than T_β with respect to subsets of ξ if every subset of ξ that is T_β-writable as a subset of ξ is also T_α-writable as a subset of ξ, but not conversely.

Moreover, we frequently use codes for ordinals. An α-*code* is a subset of α interpreted as a binary relation $\in_\alpha$ on α via Gödel pairing such that $(\alpha, \in_\alpha)$ is well-founded and extensional. This structure is isomorphic to a transitive set. The coded set is defined as the image of 0 in the transitive collapse. We work with the image of 0 instead of the whole set, since the former allows us to code arbitrary sets (this is necessary in Lemmas 3.4 and 3.5), while the latter would only yield codes for transitive sets.

We further call an ordinal T_α-*writable*, T_α-*eventually writable* or T_α-*accidentally writable* if it has an α-code with the corresponding property.[5] The analogous notions for $\hat{T}_\alpha$ are defined similarly. Note that one could similarly talk about writability for arbitrary sets, and it is easy to see that for sets of ordinals, this would agree with the original definition of writability. However, for clarity we will only use this terminology for ordinals.

2. Writable and clockable ordinals

In this section, we study variants of writability for T_α and $\hat{T}_\alpha$, the associated ordinals, their characterizations and connections with clockable ordinals.

The ordinals λ, ζ and Σ, which play an important role in the study of infinite time Turing machines, have analogues for α-tape machines. We define $\hat{\lambda}_\alpha$, $\hat{\zeta}_\alpha$, $\hat{\Sigma}_\alpha$ and λ_α, ζ_α, Σ_α as the suprema of the T_α-writable, T_α-eventually writable and T_α-accidentally writable ordinals (with respect to α-codes) with and without ordinal parameters, respectively.

We will prove some basic properties of these ordinals. Similar to the case of ITTMs in [Wel09, Section 2], an α-word (i.e. an α-length bit sequence) is an element of $L_{\hat{\lambda}_\alpha}$, $L_{\hat{\zeta}_\alpha}$ or $L_{\hat{\Sigma}_\alpha}$ if and only if it is $\hat{T}_\alpha$-writable, $\hat{T}_\alpha$-eventually writable or $\hat{T}_\alpha$-accidentally writable, respectively. For $\hat{\lambda}_\alpha$, this follows immediately from Lemma 2.2, for $\hat{\zeta}_\alpha$ from Lemma 2.3 and for $\hat{\Sigma}_\alpha$ from Lemma 2.1 below.

Moreover, an α-word is an element of L_{λ_α}, L_{ζ_α} or L_{Σ_α} if and only if it is contained as an element in some set with a T_α-writable, T_α-eventually writable or T_α-accidentally writable code, respectively. For λ_α this follows from Lemma 2.5 and for ζ_α from Lemma 2.6. The claim for Σ_α follows from the previous one about $\hat{\Sigma}_\alpha$ by Lemma 2.4.

Given the previous characterization, the reader might wonder whether all elements of L_{λ_α} are necessarily T_α-writable. This holds if and only if T_α reaches all its cells: if every ordinal below α is T_α-writable, then one can reach any cell via a program that searches for an isomorphism with an α-code for the given ordinal, and the converse is easy to see.

We will frequently use the fact that for any multiplicatively closed ξ, any γ with a T_α-writable ξ-code, L_γ also has a T_α-writable ξ-code, and the same holds for $\hat{T}_\alpha$-writable codes. To see this, one partitions ξ into γ many pieces with order type ξ and successively writes ξ-codes for L_μ onto the μth piece for all $\mu < \gamma$.

The next lemma is used to prove that $\hat{\lambda}_\alpha$ equals the supremum of $\hat{T}_\alpha$-clockable ordinals. It shows that any $\hat{T}_\alpha$-program that does not halt on input $\vec{0}$ runs into a loop between $\hat{\zeta}_\alpha$ and $\hat{\Sigma}_\alpha$, as for standard ITTMs.

Lemma 2.1. *On input $\vec{0}$, any $\hat{T}_\alpha$-program either halts before time $\hat{\zeta}_\alpha$ or runs into an ever-repeating loop in which the configuration at time $\hat{\zeta}_\alpha$ is the same as that of time $\hat{\Sigma}_\alpha$.*

---

[5] Note that the present terminology differs from that of [Rin14], in which T_α-*writability* and T_α-*eventual writability* referred to ω-length binary output sequences (as in [HL00]), and ω-codes rather than α-codes represented ordinals (and only countable ordinals were considered). Results from there need not hold for the current sense of T_α-writability, T_α-eventual writability, etc.

Proof. We refer the reader to the proof of this fact for ITTMs [Wel00, Section 2] and only sketch the changes that are necessary to adapt it to α-ITTMs. Since ordinal parameters are allowed in the definitions of $\hat{\lambda}_\alpha$, $\hat{\zeta}_\alpha$ and $\hat{\Sigma}_\alpha$, it is sufficient to prove that the limit behaviour in each cell is the same when the time approaches $\hat{\zeta}_\alpha$ and $\hat{\Sigma}_\alpha$. In other words, if the contents of the ξth cell converges when the time appoaches $\hat{\zeta}_\alpha$, then it converges to the same value at $\hat{\Sigma}_\alpha$, otherwise it diverges at $\hat{\Sigma}_\alpha$. We need ordinal parameters, since an ITTM with parameter ξ is used to observe the ξth cell.

The difference to the setting of ITTMs is that here the head doesn't move to the first cell at every limit time. We want to show that for any computation of T_α, the head position at time $\hat{\zeta}_\alpha$ is equal to the head position at time $\hat{\Sigma}_\alpha$. To adapt the proof, we define a program that simulates the given machine, and writes the current head position on an additional tape by writing 1 in every cell that precedes the head position and 0 everywhere else. At every limit time, the inferior limit of the head positions is calculated and the contents of the remaining cells are deleted. Now the proof for ITTMs shows that the tape contents for the simulation are identical at the times $\hat{\zeta}_\alpha$ and $\hat{\Sigma}_\alpha$ and thus the head positions are also equal for the original program. $\square$

Note that the version of the previous lemma for ζ_α and Σ_α fails if $\zeta_\alpha < \hat{\zeta}_\alpha$: a universal T_α-program (see Section 1.2) simulates all $\hat{T}_\alpha$-programs and thus its first ever-repeating loop begins at $\hat{\zeta}_\alpha$. Moreover, $\zeta_\alpha < \hat{\zeta}_\alpha$ is possible by Section 3.4 below.

The fact that the suprema of writable and clockable ordinals are equal [Wel00, Theorem 1.1] easily generalizes as follows to the setting with ordinal parameters.

Lemma 2.2. $\hat{\lambda}_\alpha$ *equals the strict supremum of* $\hat{T}_\alpha$-*clockable ordinals.*

Proof. Let ξ denote the supremum of $\hat{T}_\alpha$-clockable ordinals.

To show $\hat{\lambda}_\alpha \leq \xi$, take any T_α-writable ordinal β. The following program halts after at least β steps. The program writes an α-code for β, counts through the code by successively deleting the next remaining element and halts when all elements are deleted.

To show $\hat{\lambda}_\alpha \geq \xi$, take any $\hat{T}_\alpha$-clockable ordinal β. Let P be a $\hat{T}_\alpha$-program that halts at time β. By Lemma 2.1, $\beta < \hat{\zeta}_\alpha$. Thus there is an eventually $\hat{T}_\alpha$-writable ordinal $\gamma > \beta$.

Consider the following $\hat{T}_\alpha$-program. The program writes each version μ of γ and runs P up to time μ. Whenever μ changes, begin a new simulation. It is clear that this will halt when $\mu \geq \beta$. When this happens, output an α-code for μ. Thus μ is $\hat{T}_\alpha$-writable.

Since $\hat{\lambda}_\alpha$ is itself not $\hat{T}_\alpha$-writable, the previous argument shows that the supremum is strict. $\square$

Lemma 2.3. $\hat{\zeta}_\alpha$ *equals the strict supremum of* $\hat{T}_\alpha$-*stabilization times of the tape contents.*

Proof. Let γ denote the supremum of $\hat{T}_\alpha$-stabilization times of the tape contents.

To show that $\hat{\zeta}_\alpha \leq \gamma$, suppose that P eventually writes ξ. We consider a program Q that simulates P and additionally sets a flag. It is set to 0 when the output of P changes and to 1 once we have counted through μ, if the current output of P codes an ordinal μ. Then Q's stabilization time is at least ξ.

By Lemma 2.1, $\gamma \leq \hat{\zeta}_\alpha$. To show that the supremum is strict, assume that some program P stabilizes exactly at time $\hat{\zeta}_\alpha$. We search via the universal T_α-program for (an accidentally writable code for) some μ such that the output of P eventually stabilizes before μ. Then μ is eventually $\hat{T}_\alpha$-writable, contradicting the fact that $\mu > \hat{\zeta}_\alpha$. $\square$

The next result describes the basic relations between the ordinals associated to T_α and $\hat{T}_\alpha$.

Lemma 2.4.
(1) $\hat{\lambda}_\alpha$ is T_α-eventually writable.
(2) $\hat{\zeta}_\alpha$ is T_α-accidentally writable.
(3) $\hat{\Sigma}_\alpha = \Sigma_\alpha$.
Therefore $\lambda_\alpha \le \hat{\lambda}_\alpha < \zeta_\alpha \le \hat{\zeta}_\alpha < \Sigma_\alpha = \hat{\Sigma}_\alpha$.

Proof. To show 1, we simulate all $\hat{T}_\alpha$-programs, beginning with the first step of each computation and proceeding with one step of each program at a time. This is done by partitioning the tape into α many tapes of length α. For each $i < \alpha$, we define $\gamma_{i,j}$ as follows. If the ith $\hat{T}_\alpha$-program P_i halts in step j of the run of P_i with output a (code for an) ordinal γ, let $\gamma_{i,j} = \gamma$. Otherwise let $\gamma_{i,j} = 0$. Let further $\gamma_j = \sum_{i<\alpha} \gamma_{i,j}$. The output of our algorithm is set to the value $\gamma[j] = \sum_{i \le j} \gamma_i$ once the jth step of each program is completed.

In step $\hat{\lambda}_\alpha$ of the simulation, all steps $j < \hat{\lambda}_\alpha$ of each $\hat{T}_\alpha$-program are completed. Moreover, each $\hat{T}_\alpha$-program has either already halted or diverges by Lemma 2.2. Hence the output of the simulation takes the constant value $\gamma = \sum_{j < \hat{\lambda}_\alpha} \gamma_j = \sup_{j < \hat{\lambda}_\alpha} \gamma[j]$ from step $\hat{\lambda}_\alpha$ onwards.

It remains to show that $\gamma = \hat{\lambda}_\alpha$. To see that $\gamma \le \hat{\lambda}_\alpha$, note that $\gamma[j] < \hat{\lambda}_\alpha$ for all $j < \hat{\lambda}_\alpha$, since $\gamma[j]$ is $\hat{T}_\alpha$-writable. To see that $\gamma \ge \hat{\lambda}_\alpha$, note that every $\hat{T}_\alpha$-writable ordinal is of the form $\gamma_{i,j}$ for some $i < \alpha$ and $j < \hat{\lambda}_\alpha$ and $\gamma_{i,j} \le \gamma_j \le \gamma$.

The proof of 2 is similar. We simulate all $\hat{T}_\alpha$-programs as above. For each $i < \alpha$, let $\gamma_{i,j,k}$ denote the output of the ith program P_i in step k of the run of P_i, if this codes an ordinal, is constant in the interval $[j, k)$ and j is minimal with this property. Let $\gamma_{i,j,k} = 0$ otherwise. Let further $\gamma_{j,k} = \sum_{i<\alpha} \gamma_{i,j,k}$. The algorithm's output is set to $\gamma[k] = \sum_{j<k} \gamma_{j,k}$ after the jth steps of each program are completed for all $j < k$. Now let $k = \hat{\zeta}_\alpha$.

To see that $\gamma[k] \le \hat{\zeta}_\alpha$, note that $\sum_{j<l} \gamma_{j,k}$ is $\hat{T}_\alpha$-eventually writable for all $l < k$ by Lemma 2.1. To see that $\gamma[k] \ge \hat{\zeta}_\alpha$, note that every $\hat{T}_\alpha$-eventually writable ordinal is of the form $\gamma_{i,j,k}$ for some $i < \alpha$ and $j < k$ by Lemma 2.3.[6]

Finally, 3 follows from the fact that any $\hat{T}_\alpha$-accidentally writable subset of α is T_α-accidentally writable. This was already shown in Section 1.2.   □

We will see that $\lambda_\alpha < \hat{\lambda}_\alpha$ and $\zeta_\alpha < \hat{\zeta}_\alpha$ for some α in Section 3.4.

We can now prove a version of Theorem 2.2 without parameters.

Lemma 2.5. λ_α *equals the supremum of* T_α-*clockable ordinals.*

Proof. Let γ denote the supremum of T_α-clockable ordinals.

It is easy to see that $\lambda_\alpha \le \gamma$ (as for $\hat{\lambda}_\alpha$).

To see that $\gamma \le \lambda_\alpha$, take any T_α-clockable ordinal ξ. Then $\xi < \hat{\lambda}_\alpha$ by Lemma 2.2 and thus $\xi < \zeta_\alpha$ by Lemma 2.4. Fix a T_α-program P halting at time ξ and a T_α-program Q that eventually writes some $\mu \ge \xi$. For each ordinal ν output by Q, we simulate P up to time ν and output ν if P halts. The output is a T_α-writable ordinal $\nu \ge \xi$. Thus $\xi < \lambda_\alpha$.   □

---

[6]We did not specify at which time of the simulation the output equals $\gamma[k]$. It can be arranged that this happens at time k for $k = \hat{\zeta}_\alpha$.

Similarly as in Lemma 2.3 for $\hat{\zeta}_\alpha$, we obtain the following version without parameters.

Lemma 2.6. ζ_α *equals the strict supremum of T_α-stabilization times of the tape contents.*

Proof. Let γ denote the supremum of T_α-stabilization times of the tape contents.

It is easy to see that $\zeta_\alpha \leq \gamma$ (as for $\hat{\zeta}_\alpha$).

We have $\gamma \leq \hat{\zeta}_\alpha$ by Lemma 2.1 and hence $\gamma < \Sigma_\alpha$ by Lemma 2.4. The next argument for the inequality $\gamma \leq \zeta_\alpha$ and for the fact that this is a strict supremum is virtually the same as in Lemma 2.3. Suppose that some program P stabilizes at a time $\eta \geq \zeta_\alpha$. Since $\eta \leq \gamma < \Sigma_\alpha$, η is accidentally T_α-writable. We search via the universal T_α-program for (an accidentally writable code for) the least μ such that the output of P eventually stabilizes at time μ. Then $\mu = \eta$ is eventually T_α-writable, contradicting the fact that $\eta \geq \zeta_\alpha$. $\qquad\square$

We next show that $\hat{\lambda}_\alpha$ is admissible and $\hat{\zeta}_\alpha$ is Σ_2-regular. We first fix some notation. Given a class Σ of formulas, an ordinal γ is called Σ-*regular* if for no $\beta < \gamma$, there is a cofinal function $f: \beta \to \gamma$ that is Σ-definable over L_γ from parameters in L_γ. Moreover, Σ_1-regular ordinals are called *admissible*. To show that $\hat{\lambda}_\alpha$ is admissible, we need the following lemma (which must be folklore). To state the lemma, recall that Σ-*collection* states that for any Σ-formula $\varphi(x, y)$ and set A with $\forall x \in A \; \exists y \; \varphi(x, y)$, there is a set B with $\forall x \in A \; \exists y \in B \; \varphi(x, y)$.

Lemma 2.7. *Let $\gamma \in \mathrm{Ord}$ and $n \in \omega$. The following statements are equivalent:*

(a) γ *is Σ_{n+1}-regular.*[7]
(b) $L_\gamma \models \Pi_n$-collection.
(c) $L_\gamma \models \Sigma_{n+1}$-collection.

Proof. Assume that (a) holds. To show (b), take a Π_n-formula $\varphi(x, y, z)$ and $A, B \in L_\gamma$ with $L_\gamma \models \forall x \in A \; \exists y \; \varphi(x, y, B)$. Let further $f_A: \gamma_A \to A$ denote the order-preserving enumeration of A with respect to $\leq_L$ and $f: \gamma_A \to \gamma$ the function with $f(\alpha)$ equal to the least $\beta < \gamma$ with $L_\gamma \models \exists y \in L_\beta \; \varphi(f_A(\alpha), y, B)$. Then f_A is Δ_1-definable over L_γ from A.

We claim that f is Δ_{n+1}-definable over L_γ from A, B. Note that it follows from Σ_k-regularity by induction on $i \leq k$ that Σ_i- and Π_i-formulas are closed under bounded quantification. Thus $\exists y \in L_\beta \; \varphi(f_A(\alpha), y, B)$ is (in L_γ) equivalent to a Π_n-formula and the function sending α to $L_{f(\alpha)}$ is definable by the conjunction of a Σ_n and a Π_n-formula, so it is Δ^1_{n+1}-definable over L_γ from A, B. It follows that f is Δ^1_{n+1}-definable over L_γ from A, B.

By (a), $\mathrm{ran}(f)$ is bounded by some $\beta < \gamma$. Thus L_β witnesses Π_0-collection for $\varphi(x, y, z)$ and A, B. Therefore (b) holds.

It is easy to see that (b) implies (c) and (c) implies (a). $\qquad\square$

The next result is analogous to the fact that λ is admissible [HL00, Corollary 8.2].

Lemma 2.8. [8]

(1) $\hat{\lambda}_\alpha$ *is admissible.*
(2) $\hat{\zeta}_\alpha$ *is Σ_2-regular.*

---

[7]A formula is called Σ_0 if it contains only bounded quantifiers, Σ_{n+1} if it logically equivalent to a formula of the form $\exists x_0, \ldots, x_n \varphi$, where φ is Π_n, and Π_n if it is logically equivalent to a formula of the form $\neg\varphi$, where φ is Σ_n. It follows that these classes of formulas are closed under the connectives $\wedge$ and $\vee$.

[8]An anonymous referee asked whether this also holds for λ_α and ζ_α.

Proof. To show that $\hat{\lambda}_\alpha$ is admissible, it is sufficient to show that $L_{\hat{\lambda}_\alpha}$ is a model of Π_0-collection by Lemma 2.7. To this end, take any Π_0-formula $\varphi(x, y, z)$ and $A, B \in L_{\hat{\lambda}_\alpha}$ with $L_{\hat{\lambda}_\alpha} \models \forall x \in A \, \exists y \, \varphi(x, y, B)$. Thus A has a $\hat{T}_\alpha$-writable code. We generate outputs γ via the universal T_α-program. When $A \in L_\gamma$ and $L_\gamma \models \forall x \in A \, \exists y \, \varphi(x, y, B)$, output γ and halt. This program will halt since $\hat{\lambda}_\alpha$ is T_α-accidentally writable by Lemma 2.4, thus producing some $\gamma < \hat{\lambda}_\alpha$. Hence Π_0-collection holds in $L_{\hat{\lambda}_\alpha}$.

The proof of 2 is similar. By Lemma 2.7, it is sufficient to show Π_1-collection in $L_{\hat{\zeta}_\alpha}$. To see this, take a Π_1-formula $\varphi(x, y, z)$ and $A, B \in L_{\hat{\zeta}_\alpha}$ with $L_{\hat{\zeta}_\alpha} \models \forall x \in A \, \exists y \, \varphi(x, y, B)$. Thus $A, B \in L_{\hat{\zeta}_\alpha}$ have $\hat{T}_\alpha$-eventually writable codes. Consider the following $\hat{T}_\alpha$-program. For the current versions of A and B, we search for (a code for) an ordinal γ via the universal T_α-program U_α and simultaneously for each $x \in A$ for some $y \in L_\gamma$ with $\varphi(x, y, B)$. More precisely, we implement the following (simultaneous) subroutines for all $x \in A$. Take $x \in A$ and $y \in L_\gamma$ as the current candidate for $\varphi(x, y, B)$. We run a search for counterexamples to $\varphi(x, y, B)$ via U_α; if no counterexample is found, then we keep y, but discard it otherwise. If the subroutines eventually stabilize for all x, then γ is the eventual output. On the other hand, there might be some $x \in A$ such that all its candidates are discarded at some time; we then move on to $\gamma + 1$. Clearly there is an eventual output $\gamma < \hat{\zeta}_\alpha$. Hence Π_1-collection holds in $L_{\hat{\zeta}_\alpha}$. $\qquad\square$

We will further use the next variant of the submodel characterisation of λ, ζ and Σ. We say that a tuple $(\alpha_0, \dots, \alpha_n)$ is *least* with a certain property if $\forall i \leq n \, \alpha_i \leq \beta_i$ for any other such tuple $(\beta_0, \dots, \beta_n)$.

Theorem 2.9. $(\hat{\lambda}_\alpha, \hat{\zeta}_\alpha, \hat{\Sigma}_\alpha)$ *is the least triple* (μ, ν, ξ) *with* $\alpha < \mu < \nu < \xi$ *and* $L_\mu \prec_{\Sigma_1} L_\nu \prec_{\Sigma_2} L_\xi$.

Proof. The proof of $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\zeta}_\alpha} \prec_{\Sigma_2} L_{\hat{\Sigma}_\alpha}$ is virtually the same as for (λ, ζ, Σ) in [Wel09, Corollary 32].

The proof of minimality of $\hat{\zeta}_\alpha$ in [Wel09, Theorem 30] for $\alpha = \omega$ adapts to this setting. We briefly discuss the crucial role of parameters in our version. The distinction between computations with and without parameters is not visible in Welch's proof, as finite parameters are always writable. First, to show that the content of a tape cell stabilizes at time $\hat{\zeta}_\alpha$ if and only if it stabilizes at time $\hat{\Sigma}_\alpha$, it is necessary to let the machine check the evolution of the contents of each cell separately for each cell as in Lemma 2.1. This is clearly possible for the ξth cell if ξ is given as a parameter. Second, it is frequently needed that any element x of a set y with a $\hat{T}_\alpha$-writable code has itself a $\hat{T}_\alpha$-writable code. This need not be true for T_α, as x might correspond to an ordinal in the code for y that is not T_α-reachable. However, the statement for $\hat{T}_\alpha$-writable codes and its analogue for $\hat{T}_\alpha$-eventually writable codes clearly hold for $\hat{T}_\alpha$. Finally, for our machines the read-write-head is no longer reset to 0 at all limit times. In the ω-case, this is used to show that the snapshots at times ζ and Σ agree. But this issue has already been dealt with in the proof of Lemma 2.1.

To see that $\hat{\lambda}_\alpha$ is also minimal, take a triple (μ, ν, ξ) as above. Since $\nu \geq \hat{\zeta}_\alpha$, every halting $\hat{T}_\alpha$-program halts before μ and hence $\mu \geq \hat{\lambda}_\alpha$. To finally see that $\hat{\Sigma}_\alpha$ is minimal, suppose that (μ, ν, ξ) is a triple with $\xi < \hat{\Sigma}_\alpha$. Since $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\Sigma}_\alpha}$ there is such a triple below $\hat{\lambda}_\alpha$, but this contradicts the fact that $\nu \geq \hat{\zeta}_\alpha$. $\qquad\square$

Is there a version of the previous result for $(\lambda_\alpha, \zeta_\alpha, \Sigma_\alpha)$? This was asked by one of the referees of this paper. For this triple, it is natural to consider the class $\Sigma_n^{(\alpha)}$ of formulas with parameter α, and in fact $L_{\lambda_\alpha} \prec_{\Sigma_1^{(\alpha)}} L_{\zeta_\alpha} \prec_{\Sigma_2^{(\alpha)}} L_{\Sigma_\alpha}$ remains valid. To see that $(\lambda_\alpha, \zeta_\alpha, \Sigma_\alpha)$ is not necessarily the least such triple, suppose that $\zeta_\alpha < \hat{\zeta}_\alpha$ (this is possible by Lemma 3.15 below). We have $L_{\zeta_\alpha} \prec_{\Sigma_2^{(\alpha)}} L_{\hat{\zeta}_\alpha}$, since $L_{\zeta_\alpha} \prec_{\Sigma_2^{(\alpha)}} L_{\Sigma_\alpha}$ and $L_{\hat{\zeta}_\alpha} \prec_{\Sigma_2^{(\alpha)}} L_{\hat{\Sigma}_\alpha}$. Hence $(\lambda_\alpha, \zeta_\alpha, \hat{\zeta}_\alpha)$ is a triple with the required property, but $\hat{\zeta}_\alpha < \Sigma_\alpha$ by Lemma 2.4.

3. WRITABILITY STRENGTH, REACHABILITY AND L-LEVELS

3.1. **Local cardinals.** We characterize δ by connecting properties of levels of the constructible universe with writability strength. To aid this, we begin with some elementary observations.

Let Card_* denote the set of ordinals $\alpha > \omega$ that are cardinals in L_{λ_α}. The next observation states some properties of this set. We will see in Section 3.2 that $\delta = \min(\mathrm{Card}_*)$.

Observation 3.1. Suppose that κ is an uncountable cardinal.
(1) Card_* is unbounded in κ.
(2) For any $\alpha < \kappa$, there is a sequence of length α of successive multiplicatively closed ordinals below κ that is disjoint from Card_*.[9]

Proof. It is sufficient to prove 1 assuming that κ is regular. Take any $\xi < \kappa$ and let $\pi \colon h^{L_{(\kappa^+)^L}}(\xi + 1) \to L_\beta$ denote the transitive collapse. Then $\alpha = \pi(\kappa)$ is a cardinal in L_β and $\xi < \alpha < \lambda_\alpha < \beta$. Hence $\alpha \in \mathrm{Card}_*$.

For 2, we again take a regular κ. Let $\beta = \omega^{\omega^\alpha}$, $1 \leq \eta < \alpha$ and $\gamma = \omega^{\omega^{\alpha+\eta}}$ the ηth multiplicatively closed ordinal above β. Note that the inductive definition of multiplication can be carried out in L_γ, since γ is multiplicatively closed. It follows that there is a definable (over L_γ) surjection from η onto the set of multiplicatively closed ordinals between β and γ.

Moreover, it is easy to see that there are uniformly in $n \in \omega$ definable (over L_γ) functions sending ordinals θ to surjections $\theta \to \theta^n$. From these, we obtain a function sending θ to a surjection $\theta \to \theta^\omega$, the least multiplicatively closed ordinal above θ.

Using the previous functions, one easily obtains a definable (over L_γ) surjection from β onto γ. In particular, γ is not a cardinal in L_{λ_γ}. $\qquad\square$

By Observation 3.1, the following ordinals are well-defined.

Definition 3.2.
(a) For any ordinal ξ, let μ_ξ be the least α with $L_{\lambda_\alpha} \models |\alpha| > \xi$.
(b) Let $\mu_* = \min(\mathrm{Card}_*)$.

The next lemma shows that the ordinals in the previous definition are equal for $\xi = \omega$.

Lemma 3.3. $\mu_* = \mu_\omega$.

---

[9]We defined λ_α only for multiplicatively closed ordinals, so the remaining ordinals are by definition not elements of Card_*.

Proof. $\mu_\omega \leq \mu_*$ is clear. Assume towards a contradiction that $\mu_\omega < \mu_*$. By the definition of μ_*, there is a surjection $f \colon \xi \to \mu_\omega$ in $L_{\lambda_{\mu_\omega}}$ for some $\xi < \mu_\omega$. Consider a T_{μ_ω}-program that searches for such a $\xi < \mu_\omega$ and a surjection $f \colon \xi \to \mu_\omega$. We fix ξ and f that are found by the program.

Note that ξ is T_{μ_ω}-writable, so T_{μ_ω} can simulate T_ξ. Once our search suceeds, we search for a surjection $g \colon \omega \to \xi$ via a T_ξ-program. This will also suceed, since such a surjection exists in L_{λ_ξ} by the definition of μ_ω. We have produced a surjection $f \circ g \colon \omega \to \mu_\omega$ in L_{μ_ω}. But this contradicts Definition 3.2. $\square$

Lemma 3.4. *The following statements are equivalent:*
(a) $L_{\lambda_\alpha} \models |\alpha| > \xi$.
(b) *There is no α-code for a surjection $f \colon \xi \to \alpha$ that is T_α-writable with ξ as a parameter.*
(c) $L_{\hat\lambda_\alpha} \models |\alpha| > \xi$.
(d) *There is no $\hat T_\alpha$-writable α-code for a surjection $f \colon \xi \to \alpha$.*

Proof. It is easy to see that (a) is equivalent to (b), (c) to (d) and (d) implies (b). To see that (b) implies (d), it suffices to write such a code only from ξ. This can be done by simulating the program for (d) simultaneously for all ordinal parameters and halting when the required code appears. $\square$

Using the previous lemma, one can observe that μ_ξ equals the least $\alpha > \xi$ in Card_* and therefore, the function $\xi \mapsto \mu_\xi$ enumerates the successor elements of Card_* (i.e. those which are not limits of Card_*).[10] To see this, it suffices to show that for $\alpha = \mu_\xi$, we have $L_{\hat\lambda_\alpha} \models \xi^+ = \alpha$. Note that $\nu := (\xi^+)^{L_{\hat\lambda_\alpha}} \leq \alpha$ by the definition of μ_ξ and Lemma 3.4. Since one can simulate shorter tapes by using ordinal parameters, the function $\gamma \mapsto \hat\lambda_\gamma$ is monotone. We must then have $\nu = \alpha$, since $\nu < \alpha$ would contradict the minimality of α.

We further obtain the next equivalences by virtually the same proof as for Lemma 3.4.

Lemma 3.5. *The following statements are equivalent:*
(a) $L_{\lambda_\alpha} \models \alpha$ *is a cardinal.*
(b) *There is no T_α-writable α-code for a surjection $f \colon \xi \to \alpha$ for some $\xi < \alpha$.*
(c) $L_{\hat\lambda_\alpha} \models \alpha$ *is a cardinal.*
(d) *There is no $\hat T_\alpha$-writable α-code for a surjection $f \colon \xi \to \alpha$ for some $\xi < \alpha$.*

We can replace the surjections in Definition 3.2 by cofinal functions. This yields results analogous to Lemmas 3.3, 3.4 and 3.5 with virtually the same proofs, which we do not state explicitly.

Definition 3.6.
(1) For any ordinal ξ, let ν_ξ be the least α with $L_{\lambda_\alpha} \models \mathrm{cof}(\alpha) > \xi$.
(2) Let ν_* be the least $\alpha > \omega$ that is regular in L_{λ_α}.

The previous results yield the next equality.

Lemma 3.7. $\mu_\omega = \nu_\omega$.

Proof. It is clear that $\mu_\omega \leq \nu_\omega$. Assume towards a contradiction that $\mu_\omega < \nu_\omega$. We first search for a T_{μ_ω}-writable cofinal function $f \colon \omega \to \mu_\omega$ using the analogue to Lemma 3.4 for ν_ω. We then search for a sequence of surjections $f_n \colon \omega \to f(n)$. This will succeed since

---

[10]We would like to thank an anonymous referee for this observation.

$f(n) < \mu_\omega$ for all $n \in \omega$ and by the definition of μ_ω. The algorithm yields a T_{μ_ω}-writable surjection from ω onto μ_ω, contradicting the definition of μ_ω. $\square$

It follows from the combined results in this section that $\mu_* = \nu_*$ is the least α with either of the properties (a) α is uncountable in L_{λ_α} (b) α is regular in L_{λ_α} or (c) α is a cardinal in L_{λ_α}. This proves that the least ordinals satisfying (f)-(h) of Theorem 1.1 are equal.

To see that these equal the least ordinal with (i) of Theorem 1.1, first note that for $\hat{\lambda}_\alpha$, this follows from the previous results. Moreover, the claim for $\Sigma_\alpha = \hat{\Sigma}_\alpha$ and $\hat{\zeta}_\alpha$ holds, since we have $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\Sigma_\alpha}$ and $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\zeta}_\alpha}$ by Theorem 2.9. The argument for ζ_α is analogous to the proofs of Lemmas 3.4 and 3.5.

3.2. **Reachable cells.** We now give characterizations of δ via some results in the previous section.

Proposition 3.8. $\delta = \mu_*$.

Proof. To see that $\delta \leq \mu_*$, it suffices to show that T_{μ_*} doesn't reach all its cells. We thus assume otherwise. Then there is a well-defined map $f\colon \mu_* \to \mathrm{Ord}$ that sends each $\alpha < \mu_*$ to the least halting time of a program that halts with its head in the αth cell. Since the values are bounded by $\lambda_{\mu_*} \leq \hat{\lambda}_{\mu_*}$ by Lemma 2.5, f is Σ_1-definable over $L_{\hat{\lambda}_{\mu_*}}$. Since $\hat{\lambda}_{\mu_*}$ is admissible by Lemma 2.8, $\mathrm{ran}(f)$ is bounded by some $\hat{T}_{\mu_*}$-writable ordinal ξ. Now consider the following $\hat{T}_{\mu_*}$-computable function $g\colon \omega \to \mu_*$. Let $g(n)$ denote the halting position of the nth program, if this halts before time ξ, and $g(n) = 0$ otherwise. Thus $L_{\hat{\lambda}_{\mu_*}} \models \mathrm{cof}(\mu_*) = \omega$. But Lemmas 3.3 and 3.7 imply that $L_{\hat{\lambda}_{\mu_*}} \models \mathrm{cof}(\mu_*) > \omega$.

To see that $\mu_* \leq \delta$, take any $\alpha < \mu_*$. Since $\mu_* = \mu_\omega$ by Lemma 3.3, T_α can write an ω-code for α. Therefore, T_α can reach all cells by counting through this code. $\square$

We call a cell *eventually T_α-reachable* if the head on the output tape eventually stabilizes on this cell. It is natural to ask whether a similar result holds for this notion of reachability. Let η denote the least ordinal such that T_η does not eventually reach all its cells.

Proposition 3.9. $\delta = \eta$.

Proof. It is clear that $\delta \leq \eta$. Assume towards a contradiction that $\delta < \eta$. Then every cell of T_δ is eventually reachable. We partition the tapes into δ many portions of length δ. For each cell ξ, we work in the ξth portion and enumerate ξ-*candidates* (n, α) that consist of a natural number and an ordinal by accidentally writing them via U_δ. While the current ξ-candidate is considered, we pause U_δ and run the nth program on the ξth portions of the tapes as long as the head position on the output tape is stable at the ξth cell from time α onwards, with a code for n on the output tape. Once the head moves, we run U_δ for the next step and switch to the next ξ-candidate. Note that if the nth program stabilizes at all, then it does so at or before time $\hat{\zeta}_\delta$ by Lemma 2.1. This is accidentally writable by Lemma 2.4. Thus the program eventually writes an output from which we can read off a function $f\colon \delta \to \omega$ mapping ξ to n as above. It is injective, since the nth program has a unique eventual head position, if its head stabilizes at all. Since $f \in L_{\hat{\zeta}_\delta}$, we have $L_{\hat{\zeta}_\alpha} \models |\delta| = \omega$. Since $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\zeta}_\alpha}$, $L_{\hat{\lambda}_\alpha} \models |\delta| = \omega$. But this contradicts Lemma 3.3 and Proposition 3.8. $\square$

It is easy to see that the T_δ-reachable cells form an interval, since T_δ can simulate T_α for all T_δ-reachable $\alpha < \delta$ and T_α reaches all its cells. Hence δ equals the least α such that the T_α-reachable cells are bounded.

Observation 3.10. There are arbitarily large countable ordinals α such that T_α can reach unboundedly many cells, but not all of them.

Proof. Recall that that the tape length is always assumed to be multiplicatively closed (see Section 1.2).

We first claim that for any limit ordinal ξ and any $i \in \omega$, the ξ^ith cell is T_{ξ^ω}-reachable. (Note that ξ^ω is multiplicatively closed.) To see this, note that it is easy to implement a T_α-program for ordinal multiplication (uniformly in α) that sets the head to position $\beta\gamma$ when the cells with indices β and γ are marked. in this way, for any ordinal $\beta < \alpha$ and $i \in \omega$, we can move the head to position β^i and write 1s to the first β^i many cells if the βth tape cell is marked with 1 at the beginning of the computation. By carrying out these procedures one after the other, we can also write 1s to the first β^ω many cells and move the head β^ω many positions to the right. Now, to identify ξ when $\alpha = \xi^\omega$, carry out this procedure successively for all $\nu < \alpha$, starting with $\nu = 0$. As long as $\nu < \xi$, this will still leave 0s on the tape, but when considering $\nu = \xi$, the whole tape will be filled with 1s, which can be detected. Thus, we can identify the ξth tape cell. Using the multiplication algorithm, it is now easy to see that ξ^i is also reachable for all $i \in \omega$.

We further claim that that $\hat\lambda_\xi = \hat\lambda_{\xi^\omega}$ for all multiplicatively closed ordinals ξ. To see this, it suffices to note that tapes of length ξ^ω can be simulated on tapes of length ξ by splitting the tape into ω many portions and simulating a tape of length ξ^i on the ith portion.

Now take any countable ordinal ξ such that $L_{\hat\lambda_\xi + 1} \models \xi$ is uncountable. There are unboundedly many such countable ordinals, since any image of ω_1 in the transitive collapse of a countable elementary substructure of L_{ω_2} is of this form. Since we have seen that there are unboundedly many T_{ξ^ω}-reachable cells, it remains to show that not all cells are T_{ξ^ω}-reachable. Assuming otherwise, $L_{\hat\lambda_{\xi^\omega}+1} = L_{\hat\lambda_\xi+1}$ contains a surjection $f\colon \omega \to \xi^\omega$, contradicting the fact that ξ is uncountable in $L_{\hat\lambda_\xi+1}$. $\square$

3.3. **Writability strength.** The next result shows that writability strength can decrease when the tape length increases.

Proposition 3.11. *The following property of an ordinal α occurs for the first time at δ: for some μ, ν with $\omega \le \mu \le \nu < \alpha$, there is a T_ν-writable but not T_α-writable subset of μ.*

Proof. To see that δ has the required property, it suffices to find a T_ν-writable but not T_δ-writable subset of ω for some $\nu < \delta$. Note that T_ν can write an ω-code of ν for all $\nu < \delta$ by Lemma 3.3 and Proposition 3.8. Assuming the claim fails, T_δ could thus write an ω-code for any $\nu < \delta$ and would therefore reach its νth cell.

That δ is least follows from the fact that smaller devices can reach all their cells and therefore simulate all devices smaller than they are. $\square$

This suggests the question whether the writability strength for subsets of ω is comparable for different machines. The next result shows that this is the case.

Proposition 3.12. *For every α, there is an ordinal $\tau_\alpha \le \lambda_\alpha$ such that the T_α-writable subsets of ω are exactly those contained in L_{τ_α}. Hence T_α and T_β are comparable in their writability strength for subsets of ω for all α, β.*

Proof. We first claim that every T_α-writable real x is contained in some L_β with a T_α-writable ω-code. Note that $x \in L_{\lambda_\alpha}$ by Lemma 2.5. If β is least with $x \in L_\beta$, then L_β has a real code in $L_{\beta+1}$ by acceptability of the L-hierarchy. Hence such a code is T_α-accidentally writable without parameters. We run the universal T_α-program U_α to search for an ω-code of an L-level that contains x. Eventually, such an ω-code for some L_τ is written on the output tape and the machine stops.

It remains to see that every real in some L_τ with a T_α-writable ω-code y for L_τ is itself T_α-writable, but this is clear since each element of L_τ is coded in y by a natural number. $\square$

It is easy to see that the previous result fails for subsets of other ordinals if the machine has non-reachable cells.

We now turn to characterizations of δ via eventually and accidentally writable sets. The next result follows from Proposition 3.8, the fact that every T_α-accidentally writable subset of α is an element of L_{Σ_α} by Lemma 2.1 and the discussion at the end of Section 3.1.

Proposition 3.13. *The following properties of α occur first at δ:*

(a) *There is no T_α-eventually writable ξ-code (ω-code) of α for some $\xi < \alpha$.*
(b) *There is no T_α-accidentally writable ξ-code (ω-code) of α for some $\xi < \alpha$.*

We say that a set of ordinals *has a gap* if it is not an interval. For standard ITTMs there are no gaps in the writable ordinals, since from a code for an ordinal one can write a code for any smaller ordinal by simply truncating the code [HL00, Theorem 3.7]. However, for δ-codes truncating would require addressing every tape cell, which is not possible when there are non-reachable cells.

Lemma 3.14. *δ is least such that the T_δ-writable ordinals have a gap.*

Proof. There are no gaps in the T_α-writable ordinals for $\alpha < \delta$, since every cell is reachable and hence codes can be truncated at any length. We now show that $[\theta, \delta)$ is the first gap for T_δ, where θ is the least cell that is not T_δ-reachable. To see this, note that it follows from the equality $\delta = \mu_* = \mu_\omega$ in Lemma 3.3 and Proposition 3.8 that every T_δ-reachable α has a T_δ-writable ω-code and it is also clear that δ has a T_δ-writable δ-code. If some $\alpha \in [\theta, \delta)$ had a T_δ-writable δ-code, then one would be able to reach α by counting through the code, but this contradicts the choice of θ. $\square$

3.4. **The role of parameters.** While $\Sigma_\alpha = \hat{\Sigma}_\alpha$ by Lemma 2.4, the next result shows that the analogous statement for $\hat{\lambda}_\alpha$ and $\hat{\zeta}_\alpha$ fails. We would like to thank Philip Welch for providing a proof of the implication from (a) to (c). This answered an open question in a preliminary version of this paper.

Theorem 3.15. *The following statements are equivalent:*

(a) *α is countable in L_{λ_α}.*
(b) *As in (a), but with λ_α replaced by $\hat{\lambda}_\alpha$, ζ_α, $\hat{\zeta}_\alpha$ or $\Sigma_\alpha = \hat{\Sigma}_\alpha$.*
(c) *$\hat{\lambda}_\alpha = \lambda_\alpha$.*
(d) *$\hat{\zeta}_\alpha = \zeta_\alpha$.*

Proof. To see the equivalence of (a) and (b), recall that $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\zeta}_\alpha} \prec_{\Sigma_1} L_{\hat{\Sigma}_\alpha} = L_{\Sigma_\alpha}$ by Theorem 2.9 and $L_{\lambda_\alpha} \prec_{\Sigma_1^{(\alpha)}} L_{\zeta_\alpha} \prec_{\Sigma_1^{(\alpha)}} L_{\Sigma_\alpha}$, where $\Sigma_1^{(\alpha)}$ denotes Σ_1-formulas only in the

parameter α, as discussed at the end of Section 2. Thus the claim follows from the fact that countability of α is expressible by a $\Sigma_1^{(\alpha)}$-formula.

Now assume (a) and (b). Since the set of ordinals with T_α-writable ω-codes is downwards closed, α has a T_α-writable ω-code. Then any cell is T_α-reachable by counting through the code. Hence (c) and (d) hold.

Conversely, assume that (a) and (b) fail. Thus α is uncountable in L_{Σ_α}.

To show that (c) fails, let H denote the set of $e \in \mathbb{N}$ such that $T_\alpha[e]$ halts and outputs an α-code for an ordinal γ_e. Since all T_α-clockable ordinals are below λ_α by Lemma 2.5, $H \in L_{\lambda_\alpha+1}$. Since α is uncountable in L_{Σ_α} by our assumption and the L-hierarchy is acceptable [BP68, Theorem 1], $H \in L_\alpha \subseteq L_{\lambda_\alpha}$. Note that the function $f\colon H \to \lambda_\alpha$, $f(e) = \gamma_e$ is Σ_1-definable from H over $L_{\hat{\lambda}_\alpha}$. Since $\hat{\lambda}_\alpha$ is admissible by Lemma 2.8, $\lambda_\alpha = \sup_{e \in H} \gamma_e < \hat{\lambda}_\alpha$.

To show that (d) fails, let H^* denote the set of $e \in \mathbb{N}$ such that $T_\alpha[e]$ eventually outputs an α-code for an ordinal γ_e^*. Since all stabilization times are below ζ_α by Lemma 2.3, $H^* \in L_{\zeta_\alpha+1}$ and therefore $H^* \in L_\alpha \subseteq L_{\zeta_\alpha}$, as above. Moreover, the function $g\colon H^* \to \zeta_\alpha$, $g(e) = \gamma_e^*$ is Σ_2-definable from H^* over $L_{\hat{\zeta}_\alpha}$. Since $\hat{\zeta}_\alpha$ is Σ_2-regular by Lemma 2.8, $\zeta_\alpha = \sup_{e \in H^*} \gamma_e^* < \hat{\zeta}_\alpha$. $\square$

We obtain the next Corollary via Lemma 3.4 and Proposition 3.8.

Corollary 3.16. *δ equals the least ordinal α with each of the following properties:*

(a) $\lambda_\alpha < \hat{\lambda}_\alpha$.
(b) $\zeta_\alpha < \hat{\zeta}_\alpha$.

3.5. **Upper and lower bounds.** We have the following upper bound for δ. Let σ be the least ordinal α such that every Σ_1-statement true in L already holds in L_α (L_σ equals the Σ_1-hull of $\emptyset$ in L, since it contains every uniquely Σ_1-definable set). Since both the statement that a program halts and the existence of an ever-repeating loop are Σ_1-statements, the existence of δ is a Σ_1-statement and hence $\delta < \sigma$.

For a lower bound, we see that δ is a closure point of the function mapping α to Σ_α.

Theorem 3.17. *$\Sigma_\alpha < \delta$ for all $\alpha < \delta$.*[11]

Proof. By Proposition 3.8, Lemma 3.3, Lemma 3.7 and the discussion after it, δ is a regular cardinal in the admissible set $L_{\hat{\lambda}_\delta}$. Hence there is a strictly increasing sequence $\langle \xi_\beta \mid \beta < \delta \rangle \in L_{\hat{\lambda}_\delta}$ of ordinals with $\alpha < \xi_\beta < \delta$ such that $\langle L_{\xi_\beta} \mid \beta < \delta \rangle \in L_{\hat{\lambda}_\delta}$ is a chain of elementary substructures of L_δ. In particular, $L_{\xi_0} \prec_{\Sigma_1} L_{\xi_1} \prec_{\Sigma_2} L_{\xi_2}$. Since the triple $(\hat{\lambda}_\alpha, \hat{\zeta}_\alpha, \hat{\Sigma}_\alpha)$ is least with this property by Theorem 2.9, we have $\Sigma_\alpha = \hat{\Sigma}_\alpha \leq \xi_2 < \delta$. $\square$

4. OPEN QUESTIONS

Since we considered various conditions that occur at δ for the first time, it is natural to ask which of them are equivalent everywhere.

Question 4.1. Which of the conditions in Theorem 1.1 are equivalent for all ordinals?

---

[11]This strengthens the result from [Rin14] that $\zeta < \delta$ and an unpublished result by Robert Lubarsky that $\Sigma < \delta$.

Throughout the paper, we worked with the functions mapping a multiplicatively closed ordinal α to the values λ_α, $\hat{\lambda}_\alpha$, ζ_α, $\hat{\zeta}_\alpha$ and $\Sigma_\alpha = \hat{\Sigma}_\alpha$. The versions of these functions with parameters are monotone, since $\hat{T}_\beta$ can simulate $\hat{T}_\alpha$ for $\alpha \leq \beta$. Moreover, the versions without parameters are monotone below δ for a similar reason, and at δ by Theorem 3.17.

Question 4.2. Are the functions $\alpha \mapsto \lambda_\alpha, \zeta_\alpha$ monotone above δ?

We are further interested in the supremum θ_α of T_α-reachable cells. For instance, one can ask the next question.

Question 4.3. Is the function $\alpha \mapsto \theta_\alpha$ monotone?

Finally, we ask whether similar results to those in this paper hold for machines with Σ_n-limit rules [FW11].

References

[BP68]   George Boolos and Hilary Putnam. Degrees of unsolvability of constructible sets of integers. *J. Symbolic Logic*, 33:497–513, 1968.

[COW18] Merlin Carl, Sabrina Ouazzani, and Philip D. Welch. Taming Koepke's Zoo. In *CiE*, 2018.

[FW11]   Sy-David Friedman and Philip Welch. Hypermachines. *J. Symbolic Logic*, 76(2):620–636, 2011.

[HL00]   Joel David Hamkins and Andy Lewis. Infinite time Turing machines. *J. Symbolic Logic*, 65(2):567–604, 2000.

[Koe05] Peter Koepke. Turing computations on ordinals. *Bull. Symbolic Logic*, 11(3):377–397, 2005.

[Koe09] Peter Koepke. Ordinal computability. In *Mathematical theory and computational practice*, volume 5635 of *Lecture Notes in Comput. Sci.*, pages 280–289. Springer, Berlin, 2009.

[KS09]   Peter Koepke and Benjamin Seyfferth. Ordinal machines and admissible recursion theory. *Ann. Pure Appl. Logic*, 160:310–318, 2009.

[Rin14] Benjamin Rin. The computational strengths of α-tape infinite time Turing machines. *Ann. Pure Appl. Logic*, 165(9):1501–1511, 2014.

[Wel00] P. D. Welch. The length of infinite time Turing machine computations. *Bull. London Math. Soc.*, 32(2):129–136, 2000.

[Wel09] P. D. Welch. Characteristics of discrete transfinite time Turing machine models: halting times, stabilization times, and normal form theorems. *Theoret. Comput. Sci.*, 410(4-5):426–442, 2009.