WRITABILITY AND REACHABILITY FOR α-TAPE INFINITE TIME TURING MACHINES

MERLIN CARL, BENJAMIN RIN, AND PHILIPP SCHLICHT

Abstract. Infinite time Turing machines with tape length α (denoted T_α) were introduced in [Rin14] to strengthen the ω-tape machines of Hamkins and Kidder from [HL00]. It is known that for some countable ordinals α, these machines' properties are quite different from those of the ω-tape case.

We answer the main question in [Rin14] about the size of the least ordinal δ such that not all cells are halting positions of T_δ by giving various characterizations of δ. For instance, it is the least ordinal with any of the properties

(a) there is a T_α-writable real that is not T_δ-writable for some $\alpha < \delta$,
(b) δ is uncountable in L_{λ_δ}, or
(c) δ is a regular cardinal in L_{λ_δ},

where λ_δ denotes the supremum of ordinals with a T_δ-writable code of length δ.

We further use these characterizations together with an analogue to Welch's submodel characterization of the ordinals λ, ζ and Σ, to show that δ is closed under the function $\alpha \mapsto \Sigma_\alpha$, where Σ_α denotes the supremum of the ordinals with a T_α-accidentally writable code of length α.

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. A motivation for studying these machines is the fact that they model a class of functions that is closely related to the classes of Σ^1_1 and Π^1_1 sets in descriptive set theory. Moreover, several variations were studied soon after these were introduced, 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 bounds 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-ITTMs. They do not include ordinal parameters, which are present in most other models [Koe05, KS09, COW]. It is easy to see that the computability strength[1] can increase with α, though it remains the same with relatively small increases of α. When α itself is not too large, increasing its size necessarily makes the computational strength either the same or greater. However, it turns out that for sufficiently large tapes, the machines' computability strengths are not always commensurable: there exist pairs of countable ordinals such that two machines[2] 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

Date: February 15, 2018.

[1]Computability strength in the present sense refers to the ability to compute functions $f\colon 2^\alpha \to 2^\alpha$, where α is the minimum of the tape lengths to be compared.

[2]In [Rin14], the term *machine* referred to the hardware consisting of a head and a tape with specified length, together with a specified program p. The term *device* (or *machine model*) referred

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 with ordinal parameters included (as in [COW]), since one can then always simulate a shorter tape on a longer one (c.f. [Rin14, Proposition 2.1]). This is because one can easily move the head up to cell χ and halt there whenever one is allowed to mark the χth cell, as is possible when computing with parameter χ. Indeed, it is straightforward to see that an ordinal parameter $\chi < \alpha$ is equivalent to an oracle that allows a machine T_α to emulate the computational behavior of smaller machine T_χ.

In the present article, we are mainly 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[3] all of its cells. In particular, the least ordinal δ such that the computability strength of T_δ is incomparable with that of some machine with a shorter tape is equal to the least ordinal δ such that T_δ does not reach all of its cells [Rin14, Proposition 2.1]. The main question left open was the size of δ.[4] We answer this question by giving various characterizations of δ via constructible set theory. The characterizations resemble fine-structural properties of the constructible universe L, where first-order definability is replaced with infinite time writability. To state the next result, let $\hat{\lambda}_\alpha$ denote the supremum of ordinals with a code of length α that is writable by an α-ITTM with ordinal parameters. We will prove the following in Section 2.3.

Theorem 1.1. [5] *δ is equal to the least ordinal μ that is a regular cardinal in $L_{\hat{\lambda}_\mu}$.*

We will further use the characterization in Theorem 1.1 and its variants to show in Section 2.4 that δ is least such that for some $\alpha < \delta$, the writability strength of T_δ for reals strictly decreases in comparison with the machines T_χ for $\alpha \leq \chi < \delta$.

Theorem 1.2. [6] *δ is least such that for some $\alpha < \delta$, there is some T_α-writable subset of ω that is not T_δ-writable.*

By the elementary submodel characterization of λ, ζ and Σ (c.f. [Wel09, Theorem 30 & Corollary 32]), this implies that δ is larger than these ordinals, since we can obtain triples that satisfy the characterization by forming countable elementary substructures of L_δ in $L_{\hat{\lambda}_\delta}$. We will use a variant of Welch's characterization (see Theorem 2.4) to extend this result in Section 2.5.

to the hardware alone without regard to any particular program (in other words, a class of machines). However, in the present paper we use *machine* for both these notions, trusting context for clarity.

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

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

[5]see Theorem 2.15

[6]see Theorem 2.18

Theorem 1.3. [7] *$\Sigma_\xi < \delta$ for all $\xi < \delta$.* [8]

In the next section, we will give some background on α-ITTMs. The following main section contains several auxiliary result about writable and clockable ordinals as well as the proofs of the main results. We assume that the reader is familiar with infinite time Turing machines and basic facts about Gödel's constructible universe. Moreover, in the proof of Theorem 2.4, we will use Welch's proof of the submodel characterization of λ, ζ and Σ from [Wel09, Theorem 30 & Corollary 32].

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 lengths α are closed under Gödel pairing. An α-ITTM, which we also call T_α, 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. Thus, the machine can process subsets of α by representing them on its tapes via their characteristic functions, and we will freely identify subsets of α with their characteristic functions. The *input tape* carries the subset of α that is given to the machine at the start of the computation, while the results of a computation are written on the *output tape*. The remaining tape is a *work tape*, and it is easy to see that one can equivalently allow any finite number of such tapes, or in fact α many if α is closed under Gödel pairing. Moreover, each tape has a head for reading and writing, all of which move independently of each other. Note that we could equivalently use the model from [HL00] with a single head, since this can simulate our model.

The machine T_α 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. The machine can compute relative to a finite parameter subset p of α 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. These three notions of writability are different (see [HL00, Theorem 3.8]). As for Turing machines, there is a *universal T_α-program U_α* that simulates all computations on

[7] see Theorem 2.22

[8] This strengthens the result from [Rin14] that $\zeta < \delta$ and an unpublished result by Robert Lubarsky that $\Sigma < \delta$, where ζ and Σ respectively denote the suprema of eventually and accidentally writable ordinals for ITTMs.

empty input. This can be obtained by dividing the work and output tapes into infinitely many tapes of the same length. Thus every T_α-accidentally writable real is accidentally written by U_α.

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

We will further work with T_α-writable codes for ordinals. Assuming that α is closed under Gödel pairing, an *α-code* is a subset of α interpreted as a binary relation on α via Gödel pairing. This is isomorphic to a transitive set and the coded set is the image of 0 in the transitive collapse. We thus 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. [9]

2. Connections between writability strength, reachability and the constructible universe

2.1. Writable and 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. The next lemma shows that any T_α-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 T_α-program with ordinal parameters either halts before time $\hat{\lambda}_\alpha$ or runs into a forever repeating loop in which the configuration at time $\hat{\zeta}_\alpha$ is the same as that of time $\hat{\Sigma}_\alpha$.*

Proof. We refer the reader to the proof of this fact for ITTMs [Wel00, Theorem 1.1] 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 $\hat{\Sigma}_\alpha$. This means that if the contents of the χ-th cell converges when the time appoaches $\hat{\zeta}_\alpha$, then it converges to the same value at $\hat{\Sigma}_\alpha$ and otherwise it diverges at $\hat{\Sigma}_\alpha$. 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

[9] 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.

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 lim inf of the head positions is calculated and the remaining contents of the tape 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. □

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.

Theorem 2.2. *$\hat{\lambda}_\alpha$ is equal to the supremum of the ordinals that are T_α-clockable with ordinal parameters below α.*

Proof. We give a quick sketch for the reader and again refer to the proof of [Wel00, Theorem 1.1] for details. To see that each T_α-writable ordinal is majorized by some T_α-clockable ordinal, consider the program that first writes an α-code for the ordinal and then counts through the code by successively deleting the next remaining element. For the reverse implication, Lemma 2.1 implies that any program that does not halt before $\hat{\zeta}_\alpha$ will run forever. Given a halting program P, we consider each eventually writable ordinal by successively writing its versions χ and running P up to time χ. Every time χ changes, we begin a new simulation of P and as soon as P halts, we output an α-code for χ. It follows that χ is T_α-writable and that the halting time of P is below $\hat{\lambda}_\alpha$. □

In the following, we will need the fact that $\hat{\lambda}_\alpha$ is admissible and $\hat{\zeta}_\alpha$ is Σ_2-admissible, i.e., every total Σ_2-definable function on a set in $L_{\hat{\zeta}_\alpha}$ is itself an element of $L_{\hat{\zeta}_\alpha}$.

Lemma 2.3. *$\hat{\lambda}_\alpha$ is admissible and $\hat{\zeta}_\alpha$ is Σ_2-admissible for all α.*

Proof. The proof is essentially the same as that of admissibility of λ [HL00, Corollary 8.2]. We give a sketch for the reader. Otherwise there is an ordinal χ that is T_α-writable with parameters and a cofinal function $f : \chi \to \lambda_\alpha$ that is Σ_1-definable over L_{λ_α}. We now consider a T_α-program that first computes a code for χ and then runs the universal T_α-program U_α. Whenever this produces a code for an L-level, we mark all ordinals $\beta < \chi$ such that $f(\beta)$ is defined over this level. We stop when all ordinals below χ are marked. The computation halts after at least λ_α many steps, but this contradicts Theorem 2.5. The proof of Σ_2-admissibility of $\hat{\zeta}_\alpha$ is similar and we refer the reader to [Wel09, Lemma 31] for the details. □

We will also use a version of the submodel characterisation of λ, ζ and Σ.

Theorem 2.4. *The triple $(\hat{\lambda}_\alpha, \hat{\zeta}_\alpha, \hat{\Sigma}_\alpha)$ is lexicographically least with $\alpha < \hat{\lambda}_\alpha < \hat{\zeta}_\alpha < \hat{\Sigma}_\alpha$ and $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\zeta}_\alpha} \prec_{\Sigma_2} L_{\hat{\Sigma}_\alpha}$ in any order.*

Proof. The proof from [Wel09] for the case $\alpha = \omega$ to show that (λ, ζ, Σ) is lexicographically least with this property adapts to this more general statement. We briefly discuss the crucial role of the parameters. In Welch's proof, the distinction between computations with and without parameters is not visible, 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 χ-th cell if χ is given as a parameter. Second, it is frequently needed that any element x of a set y with a T_α-writable code has itself a T_α-writable code. This need not be true for α-tape machines without parameters, as x might correspond in the code for y by an ordinal that is not T_α-reachable. However, this clearly holds with parameters and an analogous statement holds for eventually writable sets. Finally, for our machines the read-write-heads are no longer reset to 0 at all limit times, which is used in the ω-case 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.

We can further see that the triple remains lexicographically least in any other permutation of its order. We first claim that there is no triple $(\lambda', \zeta', \Sigma')$ with the submodel property and $\Sigma' < \hat{\Sigma}_\alpha$. Since $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\Sigma}_\alpha}$, otherwise there is such a triple in L_λ, but this contradicts the minimality in any order. Moreover, there is no such triple $(\lambda', \zeta', \Sigma')$ with $\zeta' < \hat{\zeta}$, since a proper Σ_2-elementary submodel of L_ζ would contain all eventually writable subsets of α, and similarly $L_{\hat{\lambda}_\alpha}$ has no proper Σ_1-elementary submodels, as these would contain all writable subsets of α. $\square$

We will further need the next version of Theorem 2.2 without parameters.

Theorem 2.5. *The ordinal λ_α is equal to the supremum of the T_α-clockable ordinals.*

Proof. We first claim that $\hat{\lambda}_\alpha < \zeta_\alpha$. To see this, we simulate all programs P simultaneously for all parameters below α. When a copy of P halts, we save its output if it is a code for an ordinal and discard it otherwise. At any time, we output a code for the sum of all such ordinals. This eventually writes an ordinal above all writable ordinals with parameters below α. To see conversely that every T_α-clockable ordinal μ is below λ_α, note that it is below $\hat{\lambda}_\alpha$ by Theorem 2.2 and hence below ζ_α. We fix a program P whose halting time is μ and a program Q that eventually writes an ordinal above μ. For each ordinal ν that is output by Q, we simulate P up to time ν and output ν if P halts. Hence $\mu \leq \nu$ and ν is writable. $\square$

The previous lemma suggests the question whether the versions of λ_α, ζ_α and Σ_α with and without parameters are equal. It is easy to see that $\Sigma_\alpha = \hat{\Sigma}_\alpha$, since any α-code that is accidentally writable with a parameter below α can also be written by simulating the same program with all possible parameters. However, we now show that this is not the case for λ_α and ζ_α. To this end, we consider countable ordinals α with $\omega_1^{L_{\hat{\lambda}_\alpha}} = \alpha$ such as the least ordinal α that is uncountable in $L_{\hat{\lambda}_\alpha}$. An example for an ordinal with the latter property is the image $\pi(\omega_1)$ of ω_1 in the collapsing map $\pi\colon h^L_{\Sigma_1}(\{\omega_1\}) \to M$.

Theorem 2.6. *If $\omega_1^{L_{\hat{\lambda}_\alpha}} = \alpha$, then $\lambda_\alpha < \hat{\lambda}_\alpha$ and $\zeta_\alpha < \hat{\zeta}_\alpha$.*

Proof. For λ_α, we consider the set S of programs that halt with parameter α and output an α-code for an ordinal. The set S is definable over $L_{\hat{\lambda}_\alpha}$ and hence an element of $L_{\hat{\Sigma}_\alpha}$. It is easy to see that $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\Sigma}_\alpha}$, since one can go through all accidentally writable codes for L-levels and output this as soon as witness for the Σ_1-statement is found. Our choice of α implies that α is uncountable in $L_{\hat{\Sigma}_\alpha}$. Thus $x \in L_\alpha$ by condensation and hence there is some $\chi < \alpha$ such that S is the χ-th element in the canonical well-order of L. The Σ_1-Skolem hull $H = h^{L_{\hat{\lambda}_\alpha}}_{\Sigma_1}(\{\alpha\})$ of $\{\alpha\}$ in $L_{\hat{\lambda}_\alpha}$ is equal to the set of all elements of $L_{\hat{\lambda}_\alpha}$ that are Σ_1-definable over $L_{\hat{\lambda}_\alpha}$ in the parameter α. Since S is Σ_1-definable from χ, there is a surjection $f\colon \omega \to H$ that is

Σ_1-definable over $L_{\hat{\lambda}_\alpha}$ from χ. Moreover, it is easy to see that $\lambda_\alpha = \sup(H \cap \mathrm{Ord})$. Since $\hat{\lambda}_\alpha$ is admissible by Lemma 2.3, we have $\lambda_\alpha < \hat{\lambda}_\alpha$.

For ζ_α, let $\varphi(n,x)$ be a Σ_2-formula such that the Σ_2-formulas with the free variable x are exactly the formulas of the form $\varphi(n,x)$ for some $n \in \omega$ up to equivalence in every Σ_2-admissible set. Let further $\psi(n,x)$ be a Σ_2-formula that uniformizes $\varphi(n,x)$ over $L_{\hat{\zeta}_\alpha}$ [Jen72, Theorem 3.1] and $S = \{n \in \omega \mid \exists x\ \varphi(n,x)\}$ its projection. Since $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\zeta}_\alpha}$, by Theorem 2.4, the Σ_2-hull $H = h_{\Sigma_2}^{L_{\hat{\lambda}_\alpha}}(\{\alpha\})$ of $\{\alpha\}$ in $L_{\hat{\zeta}_\alpha}$ is equal to the range of ψ. Using the fact that α is uncountable in $L_{\hat{\zeta}_\alpha}$, we find some $\chi < \alpha$ such that S is the χ-th element in the canonical well-order of L. Since S is Σ_1-definable from χ and ψ is a Σ_2-formula, there is a surjection $f\colon \omega \to H$ that is Σ_2-definable over $L_{\hat{\zeta}_\alpha}$ from χ. Since it is easy to show that $\zeta_\alpha = \sup(H \cap \mathrm{Ord})$ and $\hat{\zeta}_\alpha$ is Σ_2-admissible by Lemma 2.3, we have $\zeta_\alpha < \hat{\zeta}_\alpha$. □

Moreover, we will frequently use that for any α with a T_β-writable χ-code, where χ is closed under Gödel pairing, L_α also has a T_β-writable χ-code. To see this, one partitions χ into χ many pieces of length χ and successively writes χ-codes for L_ξ onto the ξ-th piece for all $\xi < \chi$ to finally construct a single χ-code for L_α. Thus any subset of α in L_{λ_α} is contained in a set with a T_α-writable α-code, assuming that α is closed under Gödel pairing. For the converse, take any set x with a T_α-writable α-code, which we can assume to be transitive. Now consider a program that iteratively identifies elements of the same $\in$-rank. Its output is an α-code for an ordinal β with $x \in L_\beta$ and hence $x \in L_{\lambda_\alpha}$.

2.2. **L-levels.** In this section, we give various characterizations of δ by connecting properties of levels of the constructible universe with writability strength.

Definition 2.7. (a) Let μ_χ be least such that there is no surjection $f\colon \chi \to \mu_\chi$ in $L_{\lambda_{\mu_\chi}}$.
(b) Let $\mu_<$ be least such that there is no surjection $f\colon \chi \to \mu_\chi$ in $L_{\lambda_{\mu_<}}$ for any $\chi < \mu_<$.

We will further omit the subscript χ when $\chi = \omega$. It follows from the next observation that these ordinals are well-defined.

Observation 2.8. *For any uncountable cardinal κ, the set of $\alpha < \kappa$ such that there is no surjection $f\colon \chi \to \alpha$ in L_{λ_α} for any $\chi < \alpha$ is unbounded in κ.*

Proof. It is sufficient to prove this for uncountable regular cardinals κ. To see that there is such an ordinal above any $\chi < \kappa$, let $\pi\colon h^{L_{(\kappa^+)^L}}(\chi + 1) \to L_\beta$ be the transitive collapse and $\alpha = \pi(\kappa)$. Since π is elementary, α is a cardinal in L_β and hence it has the required property. □

On the other hand, the next result shows that many ordinals don't satisfy the condition in the previous observation.

Observation 2.9. *For any uncountable cardinal κ, the set of $\alpha < \kappa$ such that there is a surjection $f\colon \chi \to \alpha$ in L_{λ_α} for some $\chi < \alpha$ contains intervals of arbitrarily large length below κ.*

Proof. We use the fact that there are arbitrarily large $\chi < \kappa$ such that a new subset of some $\xi < \chi$ appears in $L_{\chi+1}$. Since the L-hierarchy is acceptable [BP68, Theorem 1] (see [SZ10, Definition 1.20] for the definition), some surjection $f\colon \xi \to \chi$ is definable over L_χ. We work with the ordinal $\alpha = \omega^\chi$, which is closed under Gödel pairing, and obtain a surjection $g\colon \xi \to \alpha$ that is definable over L_α for some $\xi < \chi$. We can find a T_α-writable ξ-code for α for some $\xi < \alpha$ by first writing an α-code for $L_{\alpha+1}$ and then searching for the required ξ-code. Moreover, T_β can write an α-code for β for any $\beta \in [\alpha, \alpha+\alpha)$, since α is the largest power of ω below β. Hence all ordinals in the interval $[\alpha, \alpha+\alpha)$ have the required property. $\square$

Lemma 2.10. *For each of the following objects, the ordinal μ_χ is the least one for which this doesn't exist.*

(a) *A surjection $f\colon \chi \to \mu_\chi$ in $L_{\lambda_{\mu_\chi}}$.*
(b) *The same as (a), but for $L_{\hat{\lambda}_{\mu_\chi}}$.*
(c) *A μ_χ-code for a surjection $f\colon \chi \to \mu_\chi$ that is T_{μ_χ}-writable from χ.*
(d) *The same as (c), but with finitely many additional ordinal parameters below μ_χ.*

Proof. The first condition simply restates the definition of μ_χ. We first show that μ_χ is equal to the ordinal defined by (c). Note that there is no μ_χ-code as in (c), since any such surjection would be an element of $L_{\lambda_{\mu_\chi}}$ by Theorem 2.5. To see that μ_χ is least with the property in (c), suppose that $\eta < \mu_\chi$ and $f\colon \chi \to \eta$ is a surjection in L_{λ_η}. We now simulate all T_η-programs and whenever one of them writes an η-code for L_β for some $\beta < \lambda_\eta$, we check whether this contains such a surjection f and in this case output an η-code for f. Moreover, the proof of the equivalence of the conditions defined by (b) and (d) is virtually the same.

It remains to show that the conditions given by (c) and (d) are equivalent. To this end, suppose that there is an α-code for a surjection $f\colon \chi \to \alpha$ that is T_α-writable from χ and additional parameters below α. It suffices to write such a code without additional parameters. This can be done by simulating the program for all parameter tuples of the appropriate length simultaneously and halting as soon as one of them writes an α-code for a surjection $f\colon \chi \to \alpha$. $\square$

Since $\lambda_\alpha < \hat{\lambda}_\alpha$ is possible by Theorem 2.6, the equivalence between the conditions given by (a) and (b) in the previous lemma is not immediate. We further obtain the next equivalence by virtually the same proof as that of Lemma 2.10.

Lemma 2.11. *For each of the following objects, the ordinal $\mu_<$ is the least one where this doesn't exist.*

(a) *A surjection $f\colon \chi \to \mu_<$ in $L_{\lambda_{\mu_<}}$ for some $\chi < \mu_<$.*
(b) *The same as (a), but for $L_{\hat{\lambda}_{\mu_<}}$.*
(c) *A μ_χ-code for a surjection $f\colon \chi \to \mu_<$ that is $T_{\mu_<}$-writable from χ for some $\chi < \mu_<$.*
(d) *The same as (c), but with additional ordinal parameters below $\mu_<$.*

Note that $\mu_<$ is least such that $\mu_<$ is a regular cardinal in $L_{\lambda_{\mu_<}}$. The next lemma shows that the ordinals considered in Lemmas 2.10 and 2.11 are in fact equal.

Lemma 2.12. $\mu_\omega = \mu_<$.

Proof. Since $\mu_\omega \leq \mu_<$ is clear, we assume towards a contradiction that $\mu_\omega < \mu_<$. Then there is some $\chi < \mu_\omega$ and a surjection $f\colon \chi \to \mu_\omega$ in $L_{\lambda_{\mu_\omega}}$. We now simulate all T_{μ_ω}-programs and search for such a surjection in every μ_ω-code that is produced in the simulation. As soon as a code for a function $f\colon \chi \to \mu_\omega$ appears, we search for a T_χ-writable code for a surjection $g\colon \omega \to \chi$. It is now easy to obtain a T_{μ_ω}-writable μ_ω-code for the surjection $f \circ g\colon \omega \to \mu_\omega$ and this would contradict the definition of μ_ω. $\square$

We can also obtain different characterizations of μ_ω and $\mu_<$ by replacing the surjections in Definition 2.7 by cofinal functions as follows.

Definition 2.13. (a) Let ν_χ be least such that there is no cofinal $f\colon \chi \to \nu_\chi$ in $L_{\lambda_{\nu_\chi}}$.
(b) Let $\nu_<$ be least such that there is no cofinal $f\colon \chi \to \nu_<$ in $L_{\lambda_{\nu_<}}$ for any $\chi < \nu_<$.

Virtually the same proofs as in Lemmas 2.10, 2.11 and 2.12 yield analogous equivalences as above. Using this, we obtain that all ordinals that we just considered are equal.

Lemma 2.14. $\mu_\omega = \nu_\omega$.

Proof. Since it is clear that $\mu_\omega \leq \nu_\omega$, we assume towards a contradiction that $\mu_\omega < \nu_\omega$. Then there is a T_{μ_ω}-writable ω-code for a cofinal function $f\colon \omega \to \mu_\omega$. Moreover, there are T_α-writable ω-codes for surjections $f_\alpha\colon \omega \to \alpha$ for all $\alpha < \mu_\omega$, since we assume that $\mu_\omega < \nu_\omega$. We can now find a T_{μ_ω}-writable ω-code for μ_ω by first writing an ω-code for f and then producing the required code from ω-codes for $f(n)$ for all $n \in \omega$, but this contradicts the definition of μ_ω. □

Hence $\mu = \nu$ is least with either of the properties (a) μ is uncountable in $L_{\hat{\lambda}_\mu}$ or (b) μ is regular in $L_{\hat{\lambda}_\mu}$.

2.3. **Reachable cells.** We can now give a characterization of δ with the help of the results in the previous section.

Theorem 2.15. $\delta = \mu$.

Proof. To show 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 $\alpha < \mu$ to the least halting time of a program that halts in the α-th cell. Since the halting times are bounded by λ_α by Theorem 2.5 and $\lambda_\alpha \leq \hat{\lambda}_\alpha$, this map is Σ_1-definable over $L_{\hat{\lambda}_\mu}$. Since $\hat{\lambda}_\mu$ is admissible by Lemma 2.3, $\mathrm{ran}(f)$ is bounded by some T_μ-writable ordinal χ. We consider the following T_μ-computable function $g\colon \omega \to \chi$. If the n-th program halts before time χ, let $g(n)$ be its halting position and otherwise $g(n) = 0$. We have $\mathrm{cof}(\mu) = \omega$ in $L_{\hat{\lambda}_\mu}$, since this function is an element of $L_{\hat{\lambda}_\mu}$, but this contradicts the fact that μ is regular in $L_{\hat{\lambda}_\mu}$ as a consequence of Lemma 2.14. Finally, we have $\mu \leq \delta$, since T_α can write an ω-code for α for any $\alpha < \mu$ and hence it can reach all its cells by counting through the code. □

It is natural to ask whether we can define δ via a notion of eventual reachability. We will call a cell *eventually T_α-reachable* if the head on the output tape eventually stabilizes on this cell. The next result shows that this leads to another characterization of δ.

Lemma 2.16. *The ordinal δ is least such that not every cell is eventually reachable.*

Proof. Since every cell of T_α is T_α-reachable for all $\alpha < \delta$, it is also eventually reachable. Now suppose towards a contradiction that every cell of T_δ is eventually reachable. We partition the tapes into δ many portions of length δ. For each cell χ, we work in the χ-th portions and enumerate *χ-candidates* (n, α) that consist of a natural number and an ordinal by accidentally writing them with U_δ. While the current χ-candidate is considered, we pause U_δ and run the n-th program on the χ-th portions of the tapes as long as the position of the head on the output tape is stable at the χ-th 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 χ-candidate. If the n-th program stabilizes at all, then it does so at or before some accidentally writable ordinal, since the sequence of positions of the head on the output tape runs into a loop by time $\hat{\zeta}_\delta$ by Lemma 2.1 and the fact that $\hat{\zeta}_\delta < \hat{\Sigma}_\delta = \Sigma_\delta$. The program eventually writes an output from which we can read off an injective function $f\colon \delta \to \omega$. Since this is an element of $L_{\hat{\zeta}_\delta}$, δ has cofinality ω in $L_{\hat{\zeta}_\alpha}$ and also in $L_{\hat{\lambda}_\alpha}$, since it is easy to see that $L_{\hat{\lambda}_\alpha} \prec_{\Sigma_1} L_{\hat{\zeta}_\alpha}$. But this contradicts Lemma 2.14 and Theorem 2.15. □

It is easy to see that δ is least such that the T_δ-reachable cells are bounded. In fact they form an interval, since T_δ can simulate T_α if $\alpha < \delta$ is T_δ-reachable, but T_α reaches all its cells and hence T_δ reaches all cells below α. However, this is not true in general by the next observation.

Observation 2.17. *There are arbitarily large ordinals α such that T_α can reach unboundedly many cells, but not all of them.*

Proof. We will show that for any limit ordinal χ and any $n < \omega$, the $(\chi + n)$-th cell is $T_{\chi+\omega}$-reachable. The claim follows since there are arbitarily large ordinals χ with $\chi = \omega_1^{L_{\hat{\lambda}_\chi}}$. It is sufficient to show that the χ-th cell is reachable. To this end, we write a string of 1s such that an additional 1 is added at the end of the string in each step and the head then moves ω many cells to the right. Note that it is possible to determine whether the time is a limit with the help of an extra work tape. Reading a 0 afterwards means that we did not reach the χ-th cell in the previous step; we thus move the head to the right until we jump back to the first cell, which is detected by reading 1, and go to the next loop. On the other hand, reading a 1 means that we had previously reached the χ-th cell, which is now marked by the last 1; we can move the head to the last 1 and halt. □

2.4. **Writability strength.**

Theorem 2.18. *For each of the following sets, the ordinal δ is the least such that this exists for some $\alpha < \delta$.*

(a) *A T_α-writable but not T_δ-writable subset of ω.*
(b) *The same as in (a), but for a subset of α.*

Proof. Since T_α reaches all its cells for all $\alpha < \delta$, it can simulate T_χ for all smaller χ. Thus it is sufficient to show that there is a T_α-writable but not T_δ-writable subset of ω for some $\alpha < \delta$. Assuming otherwise, T_α can write an ω-code for α for all $\alpha < \delta$ by Theorem 2.15 and the previous results. Hence T_δ could do the same, but this would imply that it can reach all its cells. □

Thus the writability strength can decrease with an increase of the tape length. The next result shows that the writability strength with respect to subsets of ω is always comparable. However, this does not hold for subsets of arbitrary ordinals for tapes with non-reachable cells.

Theorem 2.19. *For every α, there is an ordinal $\chi_\alpha \leq \lambda_\alpha$ such that the T_α-writable reals are exactly those contained in L_{χ_α}. Hence T_α and T_β are comparable in their writability strength for subsets of ω.*

Proof. Every T_α-writable real number is contained in L_{λ_α} by Lemma 2.3. We claim that every T_α-writable real x is contained in some L_β with a T_α-writable ω-code. 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 and we can run the universal program U_α to search for an ω-code of an L-level that contains x. Eventually, such an ω-code for some L_χ is written on the output tape and the machine stops. It remains to see that every real in some L_χ with a T_α-writable ω-code y for L_χ is itself T_α-writable, but this is clear since each element of L_χ is coded in y by a natural number. □

We now turn to characterizations of δ via eventually and accidentally writable sets. The next result follows from Lemma 2.10 and the fact that every accidentally T_α-writable subset of α is an element of L_{Σ_α} by Lemma 2.1.

Lemma 2.20. *δ is least such that each of the following objects doesn't exist.*
- (a) *An eventually T_δ-writable χ-code of δ for some $\chi < \delta$.*
- (b) *The same as in (a), but with $\chi = \omega$.*
- (c) *An accidentally T_δ-writable χ-code of δ for some $\chi < \delta$.*
- (d) *The same as in (c), but with $\chi = \omega$.*

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 2.21. *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 $[\chi, \delta)$ is the first gap for T_δ, where χ is the least cell that is no T_δ-reachable. To see this, note that it follows from the equality $\delta = \mu$ in Theorem 2.15 that every 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 $\alpha \in [\chi, \delta)$ 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 χ. □

2.5. **Upper and lower bounds.** We obtain an upper bound for δ by considering the least ordinal σ such that the same Σ_1-statements are true in L_σ and L. [10] The function mapping α to λ_α is Σ_1-definable, since λ_α can be calculated in any model of sufficiently large fragments of ZFC that contains α by testing which programs halt and which run into an infinite loop. Thus $\delta < \sigma$ by the definition of δ and Theorem 2.5.

Finally, it is useful to understand the properties of the functions mapping α to λ_α, $\hat{\lambda}_\alpha$, ζ_α, $\hat{\zeta}_\alpha$ and Σ_α. It is clear that $\lambda_\alpha = \hat{\lambda}_\alpha$ and $\lambda_\alpha \leq \lambda_\beta$ for $\alpha \leq \beta < \delta$, since T_β can simulate T_α. Moreover, it turns out that δ is a closure point of the function mapping α to Σ_α. This gives us lower bounds for δ.

Theorem 2.22. *$\Sigma_\xi < \delta$ for all $\xi < \delta$.* [11]

[10] Note that the Σ_1-hull H of the empty set in L is transitive and hence $H = L_\sigma$. Otherwise let $\chi \in H$ be the least ordinal with $\chi \not\subseteq H$ and $\varphi(x)$ a Σ_1-formula such that χ is the unique element of L that satisfies φ. If $\pi\colon H \to \bar{H}$ is the collapsing map, then $\pi(\chi) < \chi$ satisfies φ in $\bar{H}$ and hence in L, contradicting the fact that χ is unique.

[11] This strengthens the result from [Rin14] that $\zeta < \delta$ and an unpublished result by Robert Lubarsky that $\Sigma < \delta$, where ζ and Σ respectively denote the suprema of eventually and accidentally writable ordinals for ITTMs.

Proof. We have $\delta = \nu$ by Lemma 2.14, Theorem 2.15 and the discussion after Definition 2.13. Therefore δ is a regular cardinal in the admissible set $L_{\hat{\lambda}_\delta}$. Hence there is a strictly increasing sequence $\langle \chi_\alpha \mid \alpha < \delta \rangle$ of ordinals χ with $\xi < \chi < \delta$ such that $\langle L_{\chi_\alpha} \mid \alpha < \delta \rangle$ is a chain of elementary substructures of L_δ in $L_{\hat{\lambda}_\delta}$. In particular, $L_{\chi_0} \prec_{\Sigma_1} L_{\chi_1} \prec_{\Sigma_2} L_{\chi_2}$. Since the triple $(\hat{\Sigma}_\delta, \hat{\zeta}_\delta, \hat{\lambda}_\delta)$ is lexicographically least with this property by Theorem 2.4, we have $\hat{\Sigma}_\alpha \leq \chi_2 < \delta$. □

3. Open questions

We conclude with several open questions. Firstly, we ask whether some of the properties above that occur at δ for the first time, are also equivalent above δ.

Question 3.1. *Which of the conditions in Lemmas 2.10 and 2.11 and Theorem 2.18 are equivalent for all ordinals?*

Secondly, the functions mapping α to λ_α, $\hat{\lambda}_\alpha$, ζ_α, $\hat{\zeta}_\alpha$ and Σ_α are not well understood. For instance, they are monotone up to δ by Theorem 2.22, but it is open whether this holds in general.

Question 3.2. *Are the functions mapping α to λ_α, $\hat{\lambda}_\alpha$, ζ_α, $\hat{\zeta}_\alpha$ and Σ_α monotone?*

Moreover, we showed in Theorem 2.6 that $\lambda_\alpha < \hat{\lambda}_\alpha$ and $\zeta_\alpha < \hat{\zeta}_\alpha$ hold for some α, but we don't know precisely for which ordinals.

Question 3.3. *Can the ordinals α for which $\lambda_\alpha < \hat{\lambda}_\alpha$ and $\zeta_\alpha < \hat{\zeta}_\alpha$ hold be characterized by properties of L-levels?*

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.

[COW] Merlin Carl, Sabrina Ouazzini, and Philip Welch. Taming Koepke's Zoo. In preparation.

[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.

[Jen72] R. Björn Jensen. The fine structure of the constructible hierarchy. *Ann. Math. Logic*, 4:229–308; erratum, ibid. 4 (1972), 443, 1972. With a section by Jack Silver.

[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.

[SZ10] Ralf Schindler and Martin Zeman. Fine structure. In *Handbook of set theory. Vols. 1, 2, 3*, pages 605–656. Springer, Dordrecht, 2010.

[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.

Merlin Carl, Fachbereich Mathematik und Statistik, University of Konstanz, 78457 Konstanz, Germany
E-mail address: merlin.carl@uni-konstanz.de

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

Philipp Schlicht, Department of Computer Science, The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand
E-mail address: schlicht@math.uni-bonn.de