Taming Koepke's Zoo

Merlin Carl[1], Sabrina Ouazzani[2], and Philip Welch[3]

[1] Fachbereich Mathematik und Statistik, Universität Konstanz
Universitätsstraße 10, 78464 Konstanz, Germany
merlin.carl@uni-konstanz.de
[2] LACL, Universitè Paris-Est Crèteil
61 avenue du Gènèral de Gaulle, 94010 Crèteil, France
sabrina.ouazzani@lacl.fr
[3] School of Mathematics, University of Bristol
Clifton, Bristol, BS8 1TW, UK
P.Welch@bristol.ac.uk

Abstract. For ordinals α and β, Koepke defined (α, β)-Turing machines as Turing machines with tape length α and working time bounded above by β. So far, their computational strength was determined for $\alpha = \beta$ exponentially closed, $\alpha = \beta = \mathrm{On}$ and $(\alpha, \beta) = (\omega, \mathrm{On})$. In this paper, we determine the set of (α, β)-writable subsets of α when α is multiplicatively closed and $\beta > \alpha$ is admissible. This answers some open questions by Koepke in [5].

1 Introduction

In [2], Hamkins and Lewis introduced Turing machines which run along an ordinal time axis, while 'keeping' the tape length ω. Soon afterwards, the prospects of expanding the tape length as well were considered, along with the idea of restricting the working time to a certain ordinal. In this way, one obtains Koepke's (α, β)-Turing machines, which have a tape of length α while their working time is bounded by β. We are thus facing a family of $\mathrm{On} \times \mathrm{On}$ many models of computation, which we label 'Koepke's Zoo'.

We briefly describe the mechanics of (α, β)-Turing machines; a detailed account can be found in [8].

An (α, β)-Turing machine ((α, β)-TM, (α, β)-machine) has an input tape, an output tape and a scratch tape, each of length $\alpha \in \mathrm{On}$, each cell of which can contain 0 or 1. When α is closed under the Cantor pairing function, which will always be the case in this paper, one may grant the machine finitely many scratch tapes instead of a single one without changing any of the results in this paper; we will make silent use of this where convenient. Programs for (α, β)-TMs are regular Turing programs. We imagine the inner states occurring in a program indexed uniquely with natural numbers. The computation of such a program on an (α, β)-TM works by carrying out the usual Turing operation at successor ordinals. When the read-write-head is commanded to move to the left while currently occupying a cell indexed by a limit ordinal, it is reset to position

0. At a limit time $\delta < \beta$, the inner state is determined as the inferior limit of the sequence of earlier inner states (which, as we recall, are labelled with natural numbers). For $\iota < \alpha$, the content of the ιth tape cell at time δ is determined as the inferior limit of the sequence of earlier contents of that cell. The head position at time δ is also the inferior limit of the sequence of earlier head positions, if this is $<\alpha$. Otherwise, the head is reset to cell 0. A computation is considered to be halting if the halting state is reached in $<\beta$ many steps. Otherwise, it is considered as non-halting. We will consider computations relative to parameters. For a machine with an α-tape, a parameter is a finite sequence $\boldsymbol{p} \subseteq \alpha$ which is given to the machine by marking the cells with indices in $\boldsymbol{p}$ with 1.

The following cases have so far been considered:

(ω, ω)-machines are of course regular Turing machines. (ω, On)-machines were invented by Kidder and introduced by Hamkins and Lewis in [2] as 'Infinite Time Turing Machines' and were extensively studied e.g. in [2], [1], [11].

(α, α)-machines for admissible α were studied by Koepke and Seyfferth in [8] and turned out to compute exactly the $\Delta_1(L_\alpha)$-subsets of α. $(\mathrm{On}, \mathrm{On})$-machines are known as 'Ordinal Turing Machines', and it was shown by Koepke in [4] that, if ordinal parameters are admitted, then they compute the constructible sets of ordinals. Rin [9] considered parameter-free (α, On)-machines for arbitrary α and showed that they exhibit a somewhat extravagant behaviour; for example, there are $\alpha < \alpha'$ such that there are (α', On)-incomputable subsets of α that are nevertheless (α, On)-computable. Also, note that (α, β)-machines are equivalent to (β, β)-machines for $\alpha \geq \beta$, as no cells with index $\geq \beta$ can be reached in fewer than β many steps.

In this paper, we determine the computational strength for many of the remaining machine models; more specifically, we show that, if α is multiplicatively closed (i.e. if $\gamma, \delta < \alpha$, then $\gamma \cdot \delta < \alpha$) and $\beta > \alpha$ is admissible, then the subsets of α that are computable by an (α, β)-machine with parameters (i.e. finitely many ordinals $<\alpha$) are exactly those contained in L_β. This answers some questions left open by Koepke in section 3 of [5]. The analysis can be extended to broader classes of 'reasonably closed' α and β, which, however, leads to some extra technical complications, which we decided to avoid mainly for the sake of readability.

2 The Computational Strength of (α, β)-Turing machines

A crucial concept in the analysis of (α, β)-machines is that of the 'clockability' of an ordinal. We say that γ is α-clockable if and only if there are an α-Turing machine program P and a parameter $\boldsymbol{p}$ such that $P(\boldsymbol{p})$ runs for exactly $\gamma + 1$ many steps.

A subset $A \subseteq \alpha$ is (α, β)-writable if and only if there are an α-Turing machine program P and a parameter $\boldsymbol{p}$ such that $P(\boldsymbol{p})$ halts in $<\beta$ many steps with (the characteristic function of) A on the output tape. If this holds for some $\beta \in \mathrm{On}$, then A is α-writable.

An ordinal δ is α-writable if and only if there is an α-writable $A \subseteq \alpha$ which codes a well-ordering of length δ. Here, coding of well-orderings via subsets of α works as follows: Let p denote Cantor's ordinal pairing function. Then $x \subseteq \alpha$ is a code for the ordinal δ if and only if there is a bijection $f : \alpha \to \delta$ such that $x = \{p(\iota, \iota') : f(\iota) \in f(\iota')\}$. Note that multiplicative closure of α implies closure under p.

In the following, when we say that a program P halts, diverges, has a certain output etc. without specifying an input, we mean that P is run on the empty input, i.e. tapes that initially contain 0s everywhere.

It is not hard to see that, for any α, the α-writable ordinals form a downwards closed set: If $c \subseteq \alpha$ codes an ordinal η and $\beta < \eta$, then there is $\iota < \alpha$ such that restricting c to elements smaller than ι in the sense of c yields a code for β; this restriction is easily computable in the parameter ι. However, the same does not hold for the clockable ordinals: It is shown in [2] that the set of ω-clockable ordinals has 'gaps', i.e. there are ω-clockable ordinals β, η with $\beta + \omega < \eta$ such that no $\xi \in [\beta, \eta)$ is ω-clockable. For machines with tape length α, we call such intervals 'α-clockable gaps'. If in this situation (β', η) contains an α-clockable ordinal for every $\beta' < \beta$, then β is called the 'start of the gap'. The ordinal δ 'belongs to the gap $[\beta, \eta)$' if and only if $\beta < \delta < \eta$.

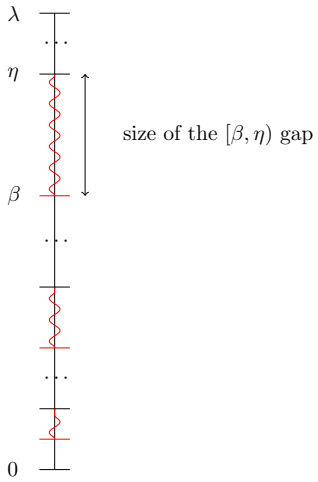

Fig. 1. A $[\beta, \eta)$ gap in the ω-clockable ordinals.

Let us observe at this point that α^ι is α-clockable for any $\iota \in \omega + 1$, which will become relevant later on.

Lemma 1. *If α is additively closed, then α^ι is α-clockable for every $\iota \in \omega + 1$.*

Proof. We show inductively that α^n is α-clockable for any $n \in \omega$. To this end, we show how to obtain, recursively in $n \in \omega$, an α-machine program Q_n that clocks α^n. It then follows that there is a program Q that runs all Q_i one after another and will thus clock α^ω.

The Q_1 case is dealt with by Rin's 'LoopAround' algorith, see [9]: Clearly, α is α-clockable by a program that simply moves the head to the right until it is back at the starting position (which is e.g. the unique cell marked with 1).

Now suppose that Q_n is given; then Q_{n+1} works as follows: Until the end of the tape is reached, place $0\underbrace{11\cdots1}_{(n+1)\times}$ to the right of the supremum ρ of all blocks of $(n+1)$ successive 1s on the working tape. Then run Q_n on the rest of the tape (when the head is reset to position 0 during the run of Q_n, it is moved to position ρ before Q_n is continued).

By multiplicative closure of α and the inductive assumption, this will take $\alpha^{n+1}+1$ many steps and thus clock α^{n+1}.

Note that Q_i can easily be obtained recursively from i for $i \in \omega$. We can thus run all Q_i in succession, which will clock α^ω.

We fix some notation. If α is an ordinal, then α^+ denotes the smallest admissible ordinal $> \alpha$. If $\alpha, \beta \in \mathrm{On}$ and β belongs to an α-clockable gap, then $\mathrm{gap}_\alpha(\beta)$ denotes the ordinal that starts this gap; otherwise, $\mathrm{gap}_\alpha(\beta) = \beta$. When $\alpha = \omega$, the subscript is dropped. We also fix natural enumerations $(P_i : i \in \omega)$ and $(\phi_i : i \in \omega)$ of the Turing programs and the $\in$-formulas, respectively.

2.1 Tape Length ω

We start by determining the computational strength of (α, β)-machines in the case that $\alpha = \omega$ and $\beta > \omega$ is admissible. This already reveals the main structure of the argument.

We will need the following facts about (ω, On)-machines, which are better known as ITTMs:

Theorem 1. *[Welch, see [11]] There are ordinals λ, ζ, Σ such that $x \subseteq \omega$ is ITTM-writable if and only if $x \in L_\lambda$. Moreover, every ITTM will either halt in $<\lambda$ many steps or cycle from time Σ on, repeating its sequence of configurations between ζ and Σ over and again. Finally, (λ, ζ, Σ) is characterized as the lexically minimal triple such that $L_\lambda \prec_{\Sigma_1} L_\zeta \prec_{\Sigma_2} L_\Sigma$.*

Theorem 2. *No admissible ordinal is ω-clockable. [Hamkins and Lewis, see [2]] Conversely, if β starts an ω-clockable gap, then β is admissible. [Welch, see [11]]*

Theorem 3. *[See [11], Lemma 48 and [1], Remark after Lemma 2] Suppose that α is ω-clockable. Then a code for α is ω-writable in exactly α many steps and a code c for L_α is ω-writable in $<\alpha^+$ many steps.*

Using these ingredients, we prove:

Theorem 4. *Let $\beta > \omega$ be admissible. Then $x \subseteq \omega$ is (ω, β)-writable if and only if one of the following holds:*

(i) β does not belong to an ω-clockable gap and $x \in L_\beta$.

(ii) β does belong to an ω-clockable gap started by some β' and $x \in L_{\beta'}$. Thus, $x \subseteq \omega$ is (ω, β)-writable if and only if $x \in L_{gap_\omega(\beta)}$.

Proof. First, note that the definition of the computation of an ω-tape Turing machine is absolute between V and the constructible hierarchy; also, by an easy induction, the computation up to time γ will be contained in $L_{\gamma+\omega}$. Thus, if $x \subseteq \omega$ is (ω, β)-writable, then it is written at some time $\gamma < \beta$; the whole computation, and hence x, will thus be contained in $L_{\gamma+\omega} \subseteq L_\beta$. (This requires only that $\gamma + \omega < \beta$ for all $\gamma < \beta$, i.e. that β is a multiple of ω^2.)

If β belongs to an ω-clockable gap started by β', then no ITTM-program will halt between times β' and β; the interval $[\beta', \beta)$ is 'empty time' as far as ITTM-Turing machine computations are concerned. Thus $x \subseteq \omega$ is (ω, β)-writable if and only if it is (ω, β') writable and (ii) reduces to (i).

Now suppose that β does not belong to an ω-clockable gap. It remains to show that every $x \in L_\beta$ is (ω, β)-writable. As β does not belong to a gap, there must be cofinally many ω-clockable ordinals below β. In particular then, there is an ω-clockable ordinal $\gamma < \beta$ such that $x \in L_\gamma$. By Theorem 3, the ω-clockability of γ implies that a code c for L_γ is writable in $<\gamma^+ \leq \beta$ many steps. In the code for L_γ, x is coded by some natural number i which can be given to the program as a parameter.[4] Now, given i and c, the code x can easily be computed in time $<\beta$, thus x is (ω, β)-writable.

2.2 The General Case

We now turn to the general case. In this section, α is always multiplicatively closed (and thus a fortiori closed under Cantor pairing) and β is always admissible and strictly greater than α. In contrast to the last section, we work in Jensen's J-hierarchy (see [3]) rather than in the Gödel-hierarchy. In addition to allowing more direct generalizations of the relevant results from [1] to the α-tape case, this has a number of technical advantages, such as the existence of uniformly and parameter-freely Σ_1-definable Σ_1-Skolem functions, i.e. maps $f : \omega \times J_\gamma^{<\omega} \to J_\gamma$ that map the pair $(i, \boldsymbol{p})$ to some (in fact, the $<_L$-minimal) witness for the ith Σ_1-formula in the parameter $\boldsymbol{p}$ (see e.g. Theorem 1.15 of [10]). Also recall from [3] that, for $\omega\alpha = \alpha$, we have $L_\alpha = J_\alpha$; thus, for reasonably closed ordinals such as α and β, but also the ordinals $\lambda(\alpha)$, $\zeta(\alpha)$ and $\Sigma(\alpha)$ defined below, this does not make a difference; in these cases, we will prefer to state our results in terms of the better-known L-hierarchy.

We start by showing that an (α, β)-machine can produce a code for L_α. To this end, we introduce two-dimensional ordinal machines as an auxiliary concept.

[4] This, of course, is unnecessary, as every natural number is ITTM-writable. We proceed in this way for the sake of analogy with the general case, in which it is no longer the case that there is a parameter-freely writable code for each ordinal below the tape length.

Definition 1. *An $\alpha\times\alpha$-Turing machine has a two-dimensional 'tape' of dimension $\alpha\times\alpha$. The head is initially located at position $(0,0)$. The head can be moved to the right, to the left, up and down. The programs are like regular Turing programs, with the modification that 'up' and 'down' are added to the legitimate head movements. Here, going 'up' from position (δ,γ) results in position $(\delta,\gamma+1)$, while going 'down' from $(\delta,\gamma+1)$ leads to (δ,γ). If δ is a limit ordinal, moving the head to the left (resp. downwards) from position (δ,γ) (resp. (γ,δ)) results in the head location $(0,\gamma)$ (resp. $(\gamma,0)$). At limit times, the head position is determined by coordinate-wise inferior limits (or resets to 0, if the limit is α). A set $A\subseteq\alpha$ is $(\alpha\times\alpha,\beta)$-writable if and only if there is an $\alpha\times\alpha$-program P that, when run on the empty 'tape', halts in $<\beta$ many steps with the characteristic function of A on the portion $\{(0,\iota):\iota<\alpha\}$ of its 'tape'.*

Lemma 2. *If $A\subseteq\alpha$ is $(\alpha\times\alpha,\beta)$-writable, then it is (α,β)-writable.*

Proof. This is proved by simulating $\alpha\times\alpha$-machines by α-machines. The simulation works by splitting the α-tape into α many disjoint portions via Cantor pairing, each representing one 'row' of the two-dimensional tape. The details are omitted for the sake of brevity.

Lemma 3. *There is an (α,β)-writable code $x\subseteq\alpha$ for $L_\alpha=J_\alpha$.*

Proof. This can be seen by using an $\alpha\times\alpha$-machine to build up the L-hierarchy iteratively, writing a code for L_γ in the $(\gamma+1)$th 'row' of the tape for $\gamma<\alpha$ and finally one for L_α on the 0th row. It can be checked that this works in $<\alpha^+$ many steps, so that the claim follows from the last lemma. The details are again omitted.

Definition 2. *Let $\alpha,\gamma\in On$. Then Σ_2-$Th(J_\gamma,\alpha)$ denotes the set of pairs $([\phi],\boldsymbol{p})$, where $[\phi]$ is (the Gödel number of) a Σ_2-formula and $\boldsymbol{p}\subseteq\alpha$ is a finite sequence such that $J_\gamma\models\phi(\boldsymbol{p})$.*

Lemma 4. *Let $c\subseteq\alpha$ be an (α,β)-writable code for a structure (X,E), where E is a binary relation on X. Then the elementary diagram T of (X,E) is (α,β)-writable. In particular, this holds for $\Sigma_2-Th(J_\gamma,\alpha)$ whenever J_γ has an (α,β)-writable code.*

Proof. This is an easy adaption of Lemma 4 of [8].

We now generalize Theorem 2.

Lemma 5. *For any multiplicatively closed $\alpha\in On$, there are ordinals $\lambda(\alpha)$, $\zeta(\alpha)$, $\Sigma(\alpha)$ such that a subset of α is writable by an (α,On)-Turing machine with finitely many ordinal parameters $<\alpha$ if and only if it is contained in $L_{\lambda(\alpha)}$, it is eventually writable by an (α,On)-Turing machine (i.e. every tape cell eventually stabilizes at the correct value) if and only if it is contained in $L_{\zeta(\alpha)}$ and it is accidentally writable by an (α,On)-tape Turing machine (i.e. it appears at some point on the output tape, whether it stays there or not) if and only if it is contained in $L_{\Sigma(\alpha)}$. Moreover, $\lambda(\alpha)$ is the supremum of the halting times of (α,On)-Turing machines with parameters. Finally, $(\lambda(\alpha),\zeta(\alpha),\Sigma(\alpha))$ is the lexically minimal triple of ordinals such that $\lambda(\alpha)>\alpha$ and $L_{\lambda(\alpha)}\prec_{\Sigma_1}L_{\zeta(\alpha)}\prec_{\Sigma_2}L_{\Sigma(\alpha)}$.*

Proof. The proof is analogous to that of the λ-ζ-Σ-theorem, see [11].

We use the following version of a subclaim from the proof of Lemma 1 from [1] (see also [12]). Here, $h_1^{J_\beta}$ is the canonical Σ_1-Skolem function for J_β (see [3]); the superscript β is dropped when it is clear from the context.

Lemma 6. *Let $\alpha \in On$, and let $\delta > \alpha$ be a limit ordinal such that $\delta < \Sigma(\alpha)$. Furthermore, let $\phi(\boldsymbol{p}) \equiv \exists x \forall y \psi(x, y, \boldsymbol{p})$ be a Σ_2-formula, where ψ is Δ_0 and $\boldsymbol{p} \subseteq \alpha$ is a finite sequence. Then $J_\delta \models \phi(\boldsymbol{p})$ if and only if there is an $n \in \omega$ such that, for all sufficiently large $\beta < \delta$, we have that the following holds in J_β: There is γ with $\gamma = 0$ or $J_\gamma \prec_{\Sigma_1} J_\beta$ such that $J_\gamma \models \phi(\boldsymbol{p})$ or $h_1(n, \gamma)$ exists and $J_\beta \models \forall y \psi(h_1(n, \gamma), \boldsymbol{p})$.*

Proof. The proof is analogous to that given in [1]; see also [12], section 4.1.

Corollary 1. *There is an α-machine program P_{limit} such that the following holds:*

Let $\delta > \alpha$ be a limit ordinal such that $\delta < \Sigma(\alpha)$, and let $\widehat{T}_\delta := \{(\phi, \boldsymbol{p}) : \boldsymbol{p} \subseteq \alpha \wedge \phi \in \Sigma_2 \wedge \exists \beta < \delta \forall \gamma < \delta (\beta < \gamma \to J_\gamma \models \phi(\boldsymbol{p}))\}$.

Then, given a subset of α coding $\widehat{T}_\delta$, P_{limit} writes a code for Σ_2-$Th(J_\delta, \alpha)$ in α many steps.

Proof. For each Σ_2-formula ϕ and each finite sequence $\boldsymbol{p}$ of ordinals in α, the question whether $L_\delta \models \phi(\boldsymbol{p})$ reduces, via Lemma 6, to the question whether there is $n \in \omega$ such that a certain Σ_2-formula ϕ' with parameter $\boldsymbol{p}$ and n holds in J_β for all sufficiently large $\beta < \delta$. The appropriate ϕ' given by Lemma 6 is recursive in ϕ and thus computable from ϕ in finitely many steps. Looking up $(\phi', \boldsymbol{p})$ in $\widehat{T}_\delta$ is done by running to and reading out the appropriate cell of the input tape, which takes $<\alpha$ many steps. This is done α many times. Searching for an appropriate $n \in \omega$ requires iterating this ω many times. By multiplicative closure of α, this takes α many steps in total.

We now work towards an α-tape version of the 'theory machine' in [1]. Intuitively, the theory machine is a program that writes codes of the Σ_2-theories of the levels of the J-hierarchy in a controllable amount of time.

Lemma 7. *There are α-TM programs P_{lft} ('level from theory') and P_{succ} such that the following holds for every $\rho < \Sigma(\alpha)$:*

(i) Given $T := \Sigma_2 - Th(J_\rho, \alpha)$ as the input, P_{lft} halts in $<\alpha^2 \cdot \omega$ many steps and outputs a code for J_ρ.

(ii) Given $T := \Sigma_2$-$Th(J_\rho, \alpha)$ as the input, P_{succ} halts in $<\alpha^\omega \cdot 2$ many steps and outputs Σ_2-$Th(J_{\rho+1}, \alpha)$.

Proof. (i) We use the following Fact generalising Theorem 3 of [13]. For this we let $\beta_0(\alpha)$ be the least $\beta > \alpha$ with $J_\beta \models ZF^-$.

Fact (Uniform Σ_n-Skolem Functions) For every $n < \omega$ there is a single Σ_n-definition of a partial function h_n, which defines a Σ_n-Skolem function over any

$\langle J_\gamma, \in\rangle$ with $\gamma < \beta_0(\alpha)$, in the sense that for any $X \subseteq J_\gamma$ then $h_n^{J_\gamma}$"$\omega \times (X \cup \alpha)$ is the least Σ_n elementary substructure of J_γ.

By our choice of ρ we have that $\rho < \beta_0(\alpha)$. Taking $n = 2$ we see there is a partial map $h = h_2^{J_\gamma}$ map of α onto J_γ. (Since $J_\gamma \models$"$\forall x(|x| \leq \alpha)$" and any $\tau \in h$"$\omega \times \alpha$ is the range of an onto map $f_\tau \in h$"$\omega \times \alpha$ from α, we shall have that h"$\omega \times \alpha$ is transitive and so is some $J_\delta \prec_{\Sigma_2} J_\gamma$ with $\delta \leq \gamma$. By our minimal choice of ρ we must have $\delta = \gamma$.) Now suppose we are given T. We have shown that h is thus a partial function from α onto J_γ. Furthermore statements such as "$h(i, \boldsymbol{\xi}) = \backslash\in h(i', \boldsymbol{\xi}')$" are included in T. Thus given the set of $(i, \boldsymbol{\xi})$ so that $h(i, \boldsymbol{\xi})$ is defined, define the equivalence relation $\sim$ and equality relation E by: $h(i, \boldsymbol{\xi}) = \backslash\in h(i', \boldsymbol{\xi}')$ respectively, then this yields a model isomorphic to $(J_\gamma, \in)$ and we are done.

(ii) Is an easy consequence of (i): Use $\mathrm{P}_{\mathrm{lft}}$ to compute a code $c \subseteq \alpha$ for J_γ from T. Then compute a code $c' \subseteq \alpha$ for $J_{\gamma+1}$ from c by successively computing the intermediate S-levels and using Lemma 4 to evaluate the $\in$-relations in these levels. Finally, read off Σ_2-$\mathrm{Th}(J_{\gamma+1}, \alpha)$ from c'.

The following is an α-TM version of Theorem 4 of [1], there called as the 'theory machine':

Lemma 8. *Let $\beta > \alpha$ be admissible. There are an α-TM program P_{theory} and an α-clockable ordinal $\gamma < \beta$ with the following property: For any $\tau < \Sigma(\alpha)$, P_{theory} contains Σ_2-$Th(J_\tau, \alpha)$ on its output tape at time $\gamma + \alpha^\omega \cdot 2 \cdot \tau$.*

Proof. We reserve the first two tape cells as 'flags'. Initially, they will be set to the contents 0 and 1. By Lemma 3, a code for J_α is (α, β)-writable; let $\gamma < \beta$ be the halting time of the respective program, then γ is clearly α-clockable. Now, when Σ_2-$\mathrm{Th}(J_\delta, \alpha)$ is given on the output tape at time τ, we use P_{succ} to write Σ_2-$\mathrm{Th}(J_{\delta+1}, \alpha)$ on the output tape; in parallel, we clock $\alpha^\omega \cdot 2$ and continue with computing Σ_2-$\mathrm{Th}(J_{\delta+2}, \alpha)$ from Σ_2-$\mathrm{Th}(J_{\delta+1}, \alpha)$ only after the clocking program has halted. Whenever a new Σ_2-$\mathrm{Th}(J_\delta, \alpha)$ has been written, the contents of the flag cells are flipped. The flag cells will thus have equal content if and only if the order type of the sequences of theories so far computed is a limit ordinal δ, in which case we use P_{limit} to compute Σ_2-$\mathrm{Th}(J_\delta, \alpha)$ from the 'limit theory' currently on the output tape. Then, we use the clocking program to wait for $\alpha^\omega 2$ many steps to ensure that the theory is not overwritten 'too early'.

Thus, we have at least the following α-version of 'quick writing':

Lemma 9. *If τ is a halting time of an α-program P with parameters $\boldsymbol{p}$, then a code for some J_γ with $\gamma \geq \tau$ is α-writable in time $<\tau^+$ in the parameter $\boldsymbol{p}$.*

Proof. We run the program P_{theory}. The statement that $P(\boldsymbol{p})$ has halted is Σ_1 in the parameter $\boldsymbol{p}$. Thus, given the parameter $\boldsymbol{p}$, it can be read off from Σ_2-$\mathrm{Th}(J_\gamma, \alpha)$ whether $P(\boldsymbol{p})$ has halted at time γ. As soon as this happens, P_{lft} is applied to the content of the output tape (which will be Σ_2-$\mathrm{Th}(J_\gamma, \alpha)$ for some γ

with $\omega\gamma > \tau$) to obtain a code for some J-level with index $\geq \tau$. By the estimates given on the running time of P_{theory} and P_{lft}, this takes place in time well before τ^+.

Theorem 5. *For any multiplicatively closed $\alpha \geq \omega$ and any admissible $\beta > \alpha$, if β does not belong to an α-clockable gap, then the subsets of α which are (α, β)-writable with finitely many ordinal parameters $<\alpha$ are exactly those contained in $L_\beta = J_\beta$. If β belongs to a gap started by γ, then it is the set of those in $L_\gamma = J_\gamma$.*

Thus, the (α, β)-writable subsets of α are exactly those in $L_{gap_\alpha(\beta)}$.

Proof. The case that β belongs to a gap is reduced to the case where it does not as in the proof of Theorem 4. So let us now assume that β does not belong to a gap, so that the α-clockable ordinals are cofinal in β.

That the (α, β)-writable subsets of α (with parameters) are contained in L_β is shown as in the proof of Theorem 4.

Thus, let $A \subseteq \alpha$ be contained in J_β. Pick $\gamma < \beta$ such that $A \in J_\gamma$. By Lemma 9, a code c for some J_δ with $\delta \geq \gamma$ can be written in time $<\beta$. From c, one can write A, as it is coded in the code for L_γ by some ordinal $\iota < \alpha$, which in turn we can give to the machine as a parameter. Thus A is (α, β)-writable.

3 Conclusion and Further Work

By the results in this paper, the missing fields in the table given on p. 8 of [5] can be filled out (new results are written in boldface and marked with an asterisk):

	Tape length ω	Tape length α multiplicatively closed	Tape length On
Time ω	$\Delta_1(L_\omega) = \Delta_1^0$ (Folklore)	$\Delta_1(L_\omega) = \Delta_1^0$ (Folklore)	$\Delta_1(L_\omega) = \Delta_1^0$ (Folklore)
Time β admissible	${}^*\mathfrak{P}(\omega) \cap L_{\mathbf{gap}_\omega(\beta)}$	$\Delta_1(L_{\min(\alpha,\beta)})$ if $\beta \leq \alpha$ and α is exponentially closed (Koepke, Seyfferth) ${}^*L_{\mathbf{gap}_\alpha(\beta)} \cap \mathfrak{P}(\alpha)$ **if** $\beta > \alpha$	$\Delta_1(L_\beta)$ (Koepke, Seyfferth)
Time On	$\mathfrak{P}(\omega) \cap L_\lambda$ (Hamkins, Lewis, Welch)	${}^*\mathfrak{P}(\alpha) \cap L_{\lambda(\alpha)}$	$\mathfrak{P}(\text{Ord}) \cap L$ (Koepke)

A natural next question concerns the notions of eventual and accidental writability associated with tape models of infinitary computability (see e.g. [2]). What are the sets of eventually and accidentally writable subsets of α for an (α, β)-TM?

Also, the arguments in this paper can probably be applied to work for weaker closure conditions on α than multiplicative closure: With some extra effort, the closure condition can probably be removed altogether. It might also well be that

it is sufficient to require that the time bound β is exponentially closed rather than admissible.

In contrast, a question left wide open by the above is the strength of (α, β)-TMs without ordinal parameters, which are basically a time-bounded version of the machines considered in [9].

Finally, we will in further work consider the (α, β)-variants of the machine model based on register machines, such as Koepke's Infinite Time Register Machines (ITRMs, [6]) and Ordinal Register Machines (ORMs, [7]). We conjecture that, for appropriately closed α and admissible $\beta > \alpha$, the (α, β)-register machine computable subsets of α will be those in $L_{\min\{\alpha^{+\omega}, \beta\}}$, where $\alpha^{+\omega}$ denotes the smallest limit of admissible ordinals $> \alpha$ (the arguments in [6] can easily be adapted to show that this is in fact an upper bound).

References

[1] S. Friedman, P. Welch. Two observations concerning infinite time Turing machines. In: I. Dimitriou (ed.), BIWOC Report, pp. 44–47 (2007) Hausdorff Centre for Mathematics, Bonn.

[2] J. D. Hamkins, A. Lewis. Infinite Time Turing Machines. Journal of Symbolic Logic 65(2), 567–604 (2000)

[3] R. Jensen. The Fine Structure of the Constructible Hierarchy. Annals of Mathematical Logic, vol. 4, (1972), pp. 229–308

[4] P. Koepke. Computing a Model of Set Theory. In New Computational Paradigms. S. Barry Cooper et al, eds., Lecture Notes in Computer Science 3988 (2006), 223–232

[5] P. Koepke. Ordinal computability. In Mathematical Theory and Computational Practice. K. Ambos-Spies et al, eds., Lecture Notes in Computer Science 5635 (2009), 280–289.

[6] P. Koepke, R. Miller. An enhanced theory of infinite time register machines. In Logic and Theory of Algorithms. A. Beckmann et al, eds., Lecture Notes in Computer Science 5028 (2008), 306–315

[7] P. Koepke, R. Syders. Computing the recursive truth predicate on ordinal register machines. In Logical Approaches to Computational Barriers, Arnold Beckmann et al., eds., Computer Science Report Series 7 (2006), Swansea, 160–169

[8] P. Koepke, B. Seyfferth. Ordinal machines and admissible recursion theory. Annals of Pure and Applied Logic, 160 (2009), 310–318.

[9] B. Rin. The computational strengths of α–tape infinite time Turing machines. Annals of Pure and Applied Logic, 165 (9), (pp. 1501–1511)

[10] R. Schindler, M. Zeman. Fine structure. In: Foreman, Matthew, Kanamori, Akihiro (Eds.). Handbook of Set Theory. Springer Netherlands (2010) (pp. 605-656).

[11] P. Welch. Characteristics of discrete transfinite time Turing machine models: halting times, stabilization times, and Normal Form theorems. Theoretical Computer Science, vol. 410, Jan. 2009, 426–442

[12] P. Welch. Some Observations on Truth Hierarchies. Review of Symbolic Logic, vo.l 7, No. 1, (2014), 1–30.

[13] P. Welch. Weak systems of determinacy and arithmetical quasi-inductive definitions. Journal of Symbolic Logic, vol. 76. No 2, (2011), (pp418-436).