THE VEBLEN FUNCTIONS FOR COMPUTABILITY THEORISTS

ALBERTO MARCONE AND ANTONIO MONTALBÁN

ABSTRACT. We study the computability-theoretic complexity and proof-theoretic strength of the following statements: (1) "If $\mathcal{X}$ is a well-ordering, then so is $\boldsymbol{\varepsilon}_{\mathcal{X}}$", and (2) "If $\mathcal{X}$ is a well-ordering, then so is $\boldsymbol{\varphi}(\alpha, \mathcal{X})$", where α is a fixed computable ordinal and $\boldsymbol{\varphi}$ represents the two-placed Veblen function. For the former statement, we show that ω iterations of the Turing jump are necessary in the proof and that the statement is equivalent to ACA_0^+ over RCA_0. To prove the latter statement we need to use ω^α iterations of the Turing jump, and we show that the statement is equivalent to $\Pi^0_{\omega^\alpha}$-CA_0. Our proofs are purely computability-theoretic. We also give a new proof of a result of Friedman: the statement "if $\mathcal{X}$ is a well-ordering, then so is $\boldsymbol{\varphi}(\mathcal{X}, 0)$" is equivalent to ATR_0 over RCA_0.

1. INTRODUCTION

The Veblen functions on ordinals are well-known and commonly used in proof theory. Proof theorists know that these functions have an interesting and complex behavior that allows them to build ordinals that are large enough to calibrate the consistency strength of different logical systems beyond Peano Arithmetic. The goal of this paper is to investigate this behavior from a computability viewpoint.

The well-known ordinal ε_0 is defined to be the first fixed point of the function $\alpha \mapsto \omega^\alpha$, or equivalently $\varepsilon_0 = \sup\{\omega, \omega^\omega, \omega^{\omega^\omega}, \dots\}$. In 1936 Gentzen [Gen36], used transfinite induction on primitive recursive predicates along ε_0, together with finitary methods, to give a proof of the consistency of Peano Arithmetic. This, combined with Gödel's Second Incompleteness Theorem, implies that Peano Arithmetic does not prove that ε_0 is a well-ordering. On the other hand, transfinite induction up to any smaller ordinal can be proved within Peano Arithmetic. This makes ε_0 the *proof-theoretic ordinal* of Peano Arithmetic.

This result kicked off a whole area of proof theory, called ordinal analysis, where the complexity of logical systems is measured in terms of (among other things) how much transfinite induction is needed to prove their consistency. (We refer the reader to [Rat06] for an exposition of the general ideas behind ordinal analysis.) The proof-theoretic ordinal of many logical systems have been calculated. An example that is relevant to this paper is the system ACA_0^+ (see Section 2.4 below), whose proof-theoretic ordinal is $\varphi_2(0) = \sup\{\varepsilon_0, \varepsilon_{\varepsilon_0}, \varepsilon_{\varepsilon_{\varepsilon_0}}, \dots\}$; the first fixed point of the *epsilon function* [Rat91, Thm. 3.5]. The *epsilon function* is the one that given γ, returns ε_γ, the γth fixed point of the function $\alpha \mapsto \omega^\alpha$ starting with $\gamma = 0$.

The Veblen functions, introduced in 1908 [Veb08], are functions on ordinals that are commonly used in proof theory to obtain the proof-theoretic ordinals of predicative theories beyond Peano Arithmetic.

- $\varphi_0(\alpha) = \omega^\alpha$.

Date: last saved: July 5, 2010
Compiled: July 5, 2010.

Marcone's research was partially supported by PRIN of Italy. Montalbán's research was partially supported by NSF grant DMS-0901169.

We thank the referees for their careful reading of the first draft of the paper and their many suggestions for improving the exposition.

- $\varphi_{\beta+1}(\alpha)$ is the αth fixed point of φ_β starting with $\alpha = 0$.
- when λ is a limit ordinal, $\varphi_\lambda(\alpha)$ is the αth simultaneous fixed point of all the φ_β for $\beta < \lambda$, also starting with $\alpha = 0$.

Note that φ_1 is the epsilon function.

The Feferman-Schütte ordinal Γ_0 is defined to be the least ordinal closed under the binary Veblen function $\varphi(\beta, \alpha) = \varphi_\beta(\alpha)$, or equivalently

$$\Gamma_0 = \sup\{\varphi_0(0), \varphi_{\varphi_0(0)}(0), \varphi_{\varphi_{\varphi_0(0)}(0)}(0), \dots\}.$$

Γ_0 is the proof-theoretic ordinal of Feferman's Predicative Analysis [Fef64, Sch77], and of ATR_0 [FMS82][1]. Again, this means that the consistency of ATR_0 can be proved by finitary methods together with transfinite induction up to Γ_0, and that ATR_0 proves the well-foundedness of any ordinal below Γ_0.

Sentences stating that a certain linear ordering is well-ordered are $\mathbf{\Pi}^1_1$. So, even if they are strong enough to prove the consistency of some theory, they have no set-existence implications. However, a sentence stating that an operator on linear orderings preserves well-orderedness is $\mathbf{\Pi}^1_2$, and hence gives rise to a natural reverse mathematics question. The following theorems answer two questions of this kind.

Theorem 1.1 (Girard, [Gir87, p. 299])**.** *Over RCA_0, the statement "if $\mathcal{X}$ is a well-ordering then $\boldsymbol{\omega}^{\mathcal{X}}$ is also a well-ordering" is equivalent to ACA_0.*

Theorem 1.2 (H. Friedman, unpublished)**.** *Over RCA_0, the statement "if $\mathcal{X}$ is a well-ordering then $\boldsymbol{\varphi}(\mathcal{X}, 0)$ is a well-ordering" is equivalent to ATR_0.*

Let $\mathbf{F}$ be an operator on linear orderings. We consider the statement

$$\mathsf{WOP}(\mathbf{F}): \quad \forall \mathcal{X}\ (\mathcal{X} \text{ is a well-ordering} \implies \mathbf{F}(\mathcal{X}) \text{ is a well-ordering}).$$

We study the behavior of $\mathbf{F}$ by analyzing the computational complexity of the proof of $\mathsf{WOP}(\mathbf{F})$ as follows. The statement $\mathsf{WOP}(\mathbf{F})$ can be restated as "if $\mathbf{F}(\mathcal{X})$ has a descending sequence, then $\mathcal{X}$ has a descending sequence to begin with". Given $\mathbf{F}$, the question we ask is:

> Given a linear ordering $\mathcal{X}$ and a descending sequence in $\mathbf{F}(\mathcal{X})$, how difficult is to build a descending sequence in $\mathcal{X}$?

From Hirst's proof of Girard's result [Hir94], we can extract the following answer for $\mathbf{F}(\mathcal{X}) = \boldsymbol{\omega}^{\mathcal{X}}$.

Theorem 1.3. *If $\mathcal{X}$ is a computable linear ordering, and $\boldsymbol{\omega}^{\mathcal{X}}$ has a computable descending sequence, then $0'$ computes a descending sequence in $\mathcal{X}$. Furthermore, there exists a computable linear ordering $\mathcal{X}$ with a computable descending sequence in $\boldsymbol{\omega}^{\mathcal{X}}$ such that every descending sequence in $\mathcal{X}$ computes $0'$.*

The first statement of the theorem follows from the results of Section 3, which includes the upper bounds of the computability-theoretic results and the "forward directions" of the reverse mathematics results. We include a proof of the second statement in Section 4, where we modify Hirst's idea to be able to apply it on our other results later. In doing so, we give a new definition of the Turing jump which, although computationally equivalent to the usual jump, is combinatorially easier to manage. This allows us to define computable approximations to the Turing jump, and we can also define a computable operation on trees that produces trees whose paths are the Turing jumps of the input tree. Furthermore, our definition of the Turing jump behaves nicely when we take iterations.

In Section 5 we use these features of our proof of Theorem 1.3. First, in Section 5.1 we consider finite iterations of the Turing jump and of ordinal exponentiation.

[1] for the definition of ATR_0 and of other subsystems of second order arithmetic mentioned in this introduction see Section 2.4 below.

(We write $\boldsymbol{\omega}^{\langle n,\mathcal{X}\rangle}$ for the nth iterate of the operation $\boldsymbol{\omega}^{\mathcal{X}}$; see Definition 2.2.) In Theorem 5.3, we prove:

Theorem 1.4. *Fix $n \in \mathbb{N}$. If $\mathcal{X}$ is a computable linear ordering, and $\boldsymbol{\omega}^{\langle n,\mathcal{X}\rangle}$ has a computable descending sequence, then $0^{(n)}$ computes a descending sequence in $\mathcal{X}$. Conversely, there exists a computable linear ordering $\mathcal{X}$ with a computable descending sequence in $\boldsymbol{\omega}^{\langle n,\mathcal{X}\rangle}$ such that the jump of every descending sequence in $\mathcal{X}$ computes $0^{(n)}$.*

From this, in Section 5.4, we obtain the following reverse mathematics result.

Theorem 1.5. *Over RCA_0, $\forall n$ $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\omega}^{\langle n,\mathcal{X}\rangle})$ is equivalent to ACA_0'.*

The first main new result of this paper is obtained in Section 5.2 and analyzes the complexity behind the epsilon function.

Theorem 1.6. *If $\mathcal{X}$ is a computable linear ordering, and $\boldsymbol{\varepsilon}_{\mathcal{X}}$ has a computable descending sequence, then $0^{(\omega)}$ can compute a descending sequence in $\mathcal{X}$. Conversely, there is a computable linear ordering $\mathcal{X}$ with a computable descending sequence in $\boldsymbol{\varepsilon}_{\mathcal{X}}$ such that the jump of every descending sequence in $\mathcal{X}$ computes $0^{(\omega)}$.*

We prove this result in Theorems 3.4 and 5.21. Then, as a corollary of the proof, we obtain the following result in Section 5.4.

Theorem 1.7. *Over RCA_0, $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varepsilon}_{\mathcal{X}})$ is equivalent to ACA_0^+.*

Our proof is purely computability-theoretic and plays with the combinatorics of the ω-jump and the epsilon function. By generalizing the previous ideas, we obtain a new definition of the ω-Turing jump, which we can also approximate by a computable function on finite strings and by a computable operator on trees. An important property of our ω-Turing jump operator is that it is essentially a fixed point of the jump operator: for every real Z, the ω-Turing jump of Z is equal to the ω-Turing jump of the jump of Z, except for the first bit (we mean equal as sequences of numbers, not only Turing equivalent). Notice the analogy with the $\boldsymbol{\varepsilon}$ and $\boldsymbol{\omega}$ operators.

After a draft of the proof of Theorem 1.7 was circulated, Afshari and Rathjen [AR09] gave a completely different proof using only proof-theoretic methods like cut-elimination, coded ω-models and Schütte deduction chains. They prove that $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varepsilon}_{\mathcal{X}})$ implies the existence of countable coded ω-models of ACA_0 containing any given set, and that this in turn is equivalent to ACA_0^+. To this end they prove a completeness-type result: given a set Z, they can either build an ω-model of ACA_0 containing Z as wanted, or obtain a proof tree of '0=1' in a suitable logical system with formulas of rank at most ω. The latter case leads to a contradiction as follows. The logical system where we get the proof tree has cut elimination, increasing the rank of the proof tree by an application of the $\boldsymbol{\varepsilon}$ operator. Using $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varepsilon}_{\mathcal{X}})$, $\mathcal{X}$ being the Kleene-Brouwer ordering on the proof tree of '0=1', they obtain a well-founded cut-free proof tree of '0=1'.

In Section 6, we move towards studying the computable complexity of the Veblen functions. Given a computable ordinal α, we calibrate the complexity of $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varphi}(\alpha,\mathcal{X}))$ with the following result, obtained by extending our definitions to ω^α-Turing jumps.

Theorem 1.8. *Let α be a computable ordinal. If $\mathcal{X}$ is a computable linear ordering, and $\boldsymbol{\varphi}(\alpha,\mathcal{X})$ has a computable descending sequence, then $0^{(\omega^\alpha)}$ computes a descending sequence in $\mathcal{X}$. Conversely, there is a computable linear ordering $\mathcal{X}$ such that $\boldsymbol{\varphi}(\alpha,\mathcal{X})$ has a computable descending sequence but every descending sequence in $\mathcal{X}$ computes $0^{(\omega^\alpha)}$.*

This result will follow from Theorem 3.6 and Theorem 6.15. In Section 6.3, as a corollary, we get the following result.

Theorem 1.9. *Let α be a computable ordinal. Over* RCA_0*, WOP*$(\mathcal{X} \mapsto \boldsymbol{\varphi}(\alpha, \mathcal{X}))$ *is equivalent to* $\Pi^0_{\omega^\alpha}$-CA_0.

Exploiting the uniformity in the proof of Theorem 1.8, we also obtain a new purely computability-theoretic proof of Friedman's result (Theorem 1.2). Before our proof, Rathjen and Weiermann [RW] found a new, fully proof-theoretic proof of Friedman's result. They use a technique similar to the proof of Afshari and Rathjen mentioned above. Friedman's original proof has two parts, one computability-theoretic and one proof-theoretic.

The table below shows the systems studied in this paper (with the exception of ACA_0'). The second column gives the proof-theoretic ordinal of the system, which were calculated by Gentzen, Rathjen, Feferman, and Schütte. The third column gives the operator $\mathbf{F}$ on linear orderings such that $\mathsf{WOP}(\mathbf{F})$ is equivalent to the given system. The last column gives references for the different proofs of these equivalences in historical order ([MM] refers to this paper).

System	p.t.o.	$\mathbf{F}(\mathcal{X})$	references
ACA_0	ε_0	$\boldsymbol{\omega}^{\mathcal{X}}$	Girard [Gir87]; Hirst [Hir94]
ACA_0^+	$\varphi_2(0)$	$\boldsymbol{\varepsilon}_{\mathcal{X}}$	[MM]; Afshari-Rathjen [AR09]
$\Pi^0_{\omega^\alpha}$-CA_0	$\varphi_{\alpha+1}(0)$	$\boldsymbol{\varphi}(\alpha, \mathcal{X})$	[MM]
ATR_0	Γ_0	$\boldsymbol{\varphi}(\mathcal{X}, 0)$	Friedman [FMW]; Rathjen-Weiermann [RW]; [MM]

Notice that in every case, the proof-theoretic ordinal equals

$$\sup\{\mathbf{F}(0), \mathbf{F}(\mathbf{F}(0)), \mathbf{F}(\mathbf{F}(\mathbf{F}(0))), \dots\}.$$

2. Background and definitions

2.1. **Veblen operators and ordinal notation.** We already know what the ω, ε and φ functions do on ordinals. In this section we define operators $\boldsymbol{\omega}$, $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varphi}$, that work on all linear orderings. These operators are computable, and when they are applied to a well-ordering, they coincide with the ω, ε and φ functions on ordinals.

To motivate the definition of $\boldsymbol{\omega}^{\mathcal{X}}$ we use the following observation due to Cantor [Can97]. Every ordinal below ω^α can be written in a unique way as a sum

$$\omega^{\beta_0} + \omega^{\beta_1} + \dots + \omega^{\beta_{k-1}},$$

where $\alpha > \beta_0 \geq \beta_1 \geq \dots \geq \beta_{k-1}$.

Definition 2.1. Given a linear ordering $\mathcal{X}$, $\boldsymbol{\omega}^{\mathcal{X}}$ is defined as the set of finite strings $\langle x_0, x_1, \dots, x_{k-1}\rangle \in \mathcal{X}^{<\omega}$ (including the empty string) where $x_0 \geq_{\mathcal{X}} x_1 \geq_{\mathcal{X}} \dots \geq_{\mathcal{X}} x_{k-1}$. We think of $\langle x_0, x_1, \dots, x_{k-1}\rangle \in \boldsymbol{\omega}^{\mathcal{X}}$ as $\omega^{x_0} + \omega^{x_1} + \dots + \omega^{x_{k-1}}$. The ordering on $\boldsymbol{\omega}^{\mathcal{X}}$ is the lexicographic one: $\langle x_0, x_1, \dots, x_{k-1}\rangle \leq_{\boldsymbol{\omega}^{\mathcal{X}}} \langle y_0, y_1, \dots, y_{l-1}\rangle$ if either $k \leq l$ and $x_i = y_i$ for every $i < k$, or for the least i such that $x_i \neq y_i$ we have $x_i <_{\mathcal{X}} y_i$.

We use the following notation for the iteration of the $\boldsymbol{\omega}$ operator.

Definition 2.2. Given a linear ordering $\mathcal{X}$, let $\boldsymbol{\omega}^{\langle 0, \mathcal{X}\rangle} = \mathcal{X}$ and $\boldsymbol{\omega}^{\langle n+1, \mathcal{X}\rangle} = \boldsymbol{\omega}^{\boldsymbol{\omega}^{\langle n, \mathcal{X}\rangle}}$.

To motivate the definition of the $\boldsymbol{\varepsilon}$ operator we start with the following observations. On the ordinals, the closure of the set $\{0\}$ under the operations $+$ and $t \mapsto \omega^t$, is the set of the ordinals strictly below ε_0. The closure of $\{0, \varepsilon_0\}$ under the same operations, is the set of the ordinals strictly below ε_1. In general, if we take the closure of $\{0\} \cup \{\, \varepsilon_\beta : \beta < \alpha \,\}$ we obtain all ordinals strictly below ε_α.

Definition 2.3. Let $\mathcal{X}$ be a linear ordering. We define $\boldsymbol{\varepsilon}_{\mathcal{X}}$ to be the set of formal terms defined as follows:

- 0 and ε_x, for $x \in \mathcal{X}$, belong to $\boldsymbol{\varepsilon}_{\mathcal{X}}$, and are called "constants",
- if $t_1, t_2 \in \boldsymbol{\varepsilon}_{\mathcal{X}}$, then $t_1 + t_2 \in \boldsymbol{\varepsilon}_{\mathcal{X}}$,
- if $t \in \boldsymbol{\varepsilon}_{\mathcal{X}}$, then $\omega^t \in \boldsymbol{\varepsilon}_{\mathcal{X}}$.

Many of the terms we defined represent the same element, so we need to find normal forms for the elements of $\boldsymbol{\varepsilon}_{\mathcal{X}}$. The definition of the ordering on $\boldsymbol{\varepsilon}_{\mathcal{X}}$ is what one should expect when $\mathcal{X}$ is an ordinal. We define the normal form of a term and the relation $\leq_{\boldsymbol{\varepsilon}_{\mathcal{X}}}$ simultaneously by induction on terms.

We say that a term $t = t_0 + \cdots + t_k$ is in *normal form* if either $t = 0$ (i.e. $k = 0$ and $t_0 = 0$), or the following holds: (a) $t_0 \geq_{\boldsymbol{\varepsilon}_{\mathcal{X}}} t_1 \geq_{\boldsymbol{\varepsilon}_{\mathcal{X}}} \cdots \geq_{\boldsymbol{\varepsilon}_{\mathcal{X}}} t_k > 0$, and (b) each t_i is either a constant or of the form ω^{s_i}, where s_i is in normal form and $s_i \neq \varepsilon_x$ for any x.

Every $t \in \boldsymbol{\varepsilon}_{\mathcal{X}}$ can be written in normal form by applying the following rules:

- $+$ is associative,
- $s + 0 = 0 + s = s$,
- if $s <_{\boldsymbol{\varepsilon}_{\mathcal{X}}} r$, then $\omega^s + \omega^r = \omega^r$,
- $\omega^{\varepsilon_x} = \varepsilon_x$.

Given $t = t_0 + \cdots + t_k$ and $s = s_0 + \cdots + s_l$ in normal form, we let $t \leq_{\boldsymbol{\varepsilon}_{\mathcal{X}}} s$ if one of the following conditions apply

- $t = 0$,
- $t = \varepsilon_x$ and, for some $y \geq_{\mathcal{X}} x$, ε_y occurs in s,
- $t = \omega^{t'}$, $s_0 = \varepsilon_y$ and $t' \leq_{\boldsymbol{\varepsilon}_{\mathcal{X}}} \varepsilon_y$,
- $t = \omega^{t'}$, $s_0 = \omega^{s'}$ and $t' \leq_{\boldsymbol{\varepsilon}_{\mathcal{X}}} s'$,
- $k > 0$ and $t_0 <_{\boldsymbol{\varepsilon}_{\mathcal{X}}} s_0$,
- $k > 0$, $t_0 = s_0$, $l > 0$ and $t_1 + \cdots + t_k \leq_{\boldsymbol{\varepsilon}_{\mathcal{X}}} s_1 + \cdots + s_l$.

The observation we made before the definition shows how the $\boldsymbol{\varepsilon}$ operator coincides with the ε-function when $\mathcal{X}$ is an ordinal (this includes the case $\mathcal{X} = \emptyset$, when 0 is the only constant and we obtain ε_0 as expected).

Definition 2.4. In analogy with Definition 2.2, for $t \in \boldsymbol{\varepsilon}_{\mathcal{X}}$ we use $\omega^{\langle n,t\rangle}$ to denote the term in $\boldsymbol{\varepsilon}_{\mathcal{X}}$ obtained by applying the ω function symbol n times to t.

Definition 2.5. If $\mathcal{X}$ is a linear ordering and $x \in \mathcal{X}$, let $\mathcal{X} \restriction x$ be the linear ordering with domain $\{\, y \in \mathcal{X} : y <_{\mathcal{X}} x \,\}$.

The following lemma expresses the compatibility of the $\boldsymbol{\omega}$ and $\boldsymbol{\varepsilon}$ operators.

Lemma 2.6. *If $\mathcal{X}$ is a linear ordering, then for every $t \in \boldsymbol{\varepsilon}_{\mathcal{X}}$ and $n \in \mathbb{N}$*

$$\boldsymbol{\omega}^{\langle n, \boldsymbol{\varepsilon}_{\mathcal{X}} \restriction t\rangle} \cong \boldsymbol{\varepsilon}_{\mathcal{X}} \restriction \omega^{\langle n,t\rangle}$$

via a computable isomorphism. In particular, $\boldsymbol{\omega}^{\boldsymbol{\varepsilon}_{\mathcal{X}} \restriction t} \cong \boldsymbol{\varepsilon}_{\mathcal{X}} \restriction \omega^t$.

Proof. The proof is by induction on n. When $n = 0$ the identity is the required isomorphism. If $\psi : \boldsymbol{\omega}^{\langle n, \boldsymbol{\varepsilon}_{\mathcal{X}} \restriction t\rangle} \to \boldsymbol{\varepsilon}_{\mathcal{X}} \restriction \omega^{\langle n,t\rangle}$ is an isomorphism, then the function mapping the empty string to 0 and $\langle t_0, \ldots, t_k\rangle$ to $\omega^{\psi(t_0)} + \cdots + \omega^{\psi(t_k)}$ witnesses $\boldsymbol{\omega}^{\langle n+1, \boldsymbol{\varepsilon}_{\mathcal{X}} \restriction t\rangle} \cong \boldsymbol{\varepsilon}_{\mathcal{X}} \restriction \omega^{\langle n+1,t\rangle}$. □

To define the $\boldsymbol{\varphi}$ operator we start with the following observations. If we take the closure of the set $\{0\}$ under the operations $+$, $t \mapsto \omega^t$ and $t \mapsto \varepsilon_t$, we get all the ordinals up to $\varphi_2(0)$. If we take the closure of $\{0\} \cup \{\, \varphi_2(\beta) : \beta < \alpha \,\}$ we get all the ordinals below $\varphi_2(\alpha)$. In general, we obtain $\varphi_\gamma(\alpha)$ as the closure of $\{0\} \cup \{\, \varphi_\gamma(\beta) : \beta < \alpha \,\}$ under the operations $+$, and $t \mapsto \varphi_\delta(t)$, for all $\delta < \gamma$.

Definition 2.7. Let $\mathcal{X}$ and $\mathcal{Y}$ be linear orderings. We define $\boldsymbol{\varphi}(\mathcal{Y}, \mathcal{X})$ to be the set of formal terms defined as follows:

- 0 and $\varphi_{\mathcal{Y},x}$, for $x \in \mathcal{X}$, belong to $\boldsymbol{\varphi}(\mathcal{Y}, \mathcal{X})$, and are called "constants",

- if $t_1, t_2 \in \boldsymbol{\varphi}(\mathcal{Y}, \mathcal{X})$, then $t_1 + t_2 \in \boldsymbol{\varphi}(\mathcal{Y}, \mathcal{X})$,
- if $t \in \boldsymbol{\varphi}(\mathcal{Y}, \mathcal{X})$ and $\delta \in \mathcal{Y}$, then $\varphi_\delta(t) \in \boldsymbol{\varphi}(\mathcal{Y}, \mathcal{X})$.

We define the normal form of a term and the relation $\leq_{\boldsymbol{\varphi}(\mathcal{Y},\mathcal{X})}$ simultaneously by induction on terms. We write $\leq_\varphi$ instead of $\leq_{\boldsymbol{\varphi}(\mathcal{Y},\mathcal{X})}$ to simplify the notation.

We say that a term $t = t_0 + \cdots + t_k$ is in *normal form* if either $t = 0$, or the following holds: (a) $t_0 \geq_\varphi t_1 \geq_\varphi \cdots \geq_\varphi t_k > 0$, and (b) each t_i is either a constant or of the form $\varphi_\delta(s_i)$, where s_i is in normal form and $s_i \neq \varphi_{\delta'}(s'_i)$ for $\delta' > \delta$.

Every $t \in \boldsymbol{\varphi}(\mathcal{Y}, \mathcal{X})$ can be written in normal form by applying the following rules:

- $+$ is associative,
- $s + 0 = 0 + s = s$,
- if $\varphi_{\delta'}(s) <_\varphi \varphi_\delta(r)$, then $\varphi_{\delta'}(s) + \varphi_\delta(r) = \varphi_\delta(r)$.
- if $\delta' > \delta$, then $\varphi_\delta(\varphi_{\delta'}(r)) = \varphi_{\delta'}(r)$.
- if $\delta \in \mathcal{Y}$ then $\varphi_\delta(\varphi_{\mathcal{Y},r}) = \varphi_{\mathcal{Y},r}$.

The motivation for the last two items is that if $\delta' > \delta$, anything in the image of $\varphi_{\delta'}$ is a fixed point of φ_δ.

Given $t = t_0 + \cdots + t_k$ and $s = s_0 + \cdots + s_l$ in normal form, we let $t \leq_\varphi s$ if one of the following conditions apply

- $t = 0$,
- $t = \boldsymbol{\varphi}_{\mathcal{Y},x}$ and, for some $y \geq_{\mathcal{X}} x$, $\varphi_{\mathcal{Y},y}$ occurs in s,
- $t = \varphi_\delta(t')$, $s_0 = \varphi_{\delta'}(s')$ and $\begin{cases} \delta < \delta' \text{ and } t' \leq_\varphi \varphi_{\delta'}(s'), \text{ or} \\ \delta = \delta' \text{ and } t' \leq_\varphi s', \text{ or} \\ \delta > \delta' \text{ and } \varphi_\delta(t') \leq_\varphi s', \end{cases}$
- $k > 0$ and $t_0 <_\varphi s_0$,
- $k > 0$, $t_0 = s_0$, $l > 0$ and $t_1 + \cdots + t_k \leq_\varphi s_1 + \cdots + s_l$.

2.2. **Notation for strings and trees.** Here we fix our notation for sequences (or strings) of natural numbers. The *Baire space* $\mathbb{N}^\mathbb{N}$ is the set of all infinite sequences of natural numbers. As usual, an element of $\mathbb{N}^\mathbb{N}$ is also called a *real*. If $X \in \mathbb{N}^\mathbb{N}$ and $n \in \mathbb{N}$, $X(n)$ is the $(n+1)$-st element of X. $\mathbb{N}^{<\mathbb{N}}$ is the set of all finite strings of natural numbers. When $\sigma \in \mathbb{N}^{<\mathbb{N}}$ we use $|\sigma|$ to denote its *length* and, for $i < |\sigma|$, $\sigma(i)$ to denote its $(i+1)$-st element. We write $\emptyset$ for the *empty string* (i.e. the only string of length 0), and $\langle n \rangle$ for the string of length 1 whose only element is n. When $\sigma, \tau \in \mathbb{N}^{<\mathbb{N}}$, $\sigma \subseteq \tau$ means that σ is an *initial segment* of τ, i.e. $|\sigma| \leq |\tau|$ and $\sigma(i) = \tau(i)$ for each $i < |\sigma|$. We use $\sigma \subset \tau$ to mean $\sigma \subseteq \tau$ and $\sigma \neq \tau$. If $X \in \mathbb{N}^\mathbb{N}$ we write $\sigma \subset X$ if $\sigma(i) = X(i)$ for each $i < |\sigma|$. We use $\sigma^\frown\tau$ to denote the *concatenation* of σ and τ, that is the string ρ such that $|\rho| = |\sigma| + |\tau|$, $\rho(i) = \sigma(i)$ when $i < |\sigma|$, and $\rho(|\sigma| + i) = \tau(i)$ when $i < |\tau|$. If $X \in \mathbb{N}^\mathbb{N}$, $\sigma \in \mathbb{N}^{<\mathbb{N}}$ and $t \in \mathbb{N}$, $X{\restriction}t$ is the initial segment of X of length t, while $\sigma{\restriction}t$ is the initial segment of σ of length t if $t \leq |\sigma|$, and σ otherwise.

We fix an enumeration of $\mathbb{N}^{<\mathbb{N}}$, so that each finite string is also a natural number, and hence can be an element of another string. This enumeration is such that all the operations and relations discussed in the previous paragraph are computable. Moreover we can assume that $\sigma \subset \tau$ (as strings) implies $\sigma < \tau$ (as natural numbers). For an enumeration with these properties see e.g. [Sim99, §II.2].

The following operation on strings will be useful.

Definition 2.8. If $\sigma \in \mathbb{N}^{<\mathbb{N}}$ is nonempty let $\ell(\sigma) = \langle \sigma(|\sigma| - 1) \rangle$, the string of length one whose only entry is the last entry of σ.

Definition 2.9. A *tree* is a set $T \subseteq \mathbb{N}^{<\mathbb{N}}$ such that $\sigma{\restriction}t \in T$ whenever $\sigma \in T$ and $t < |\sigma|$. If T is a tree, $X \in \mathbb{N}^\mathbb{N}$ is *a path through* T if $X{\restriction}t \in T$ for all t. We let $[T]$ be the set of all paths through T.

Definition 2.10. If T is a tree and $\sigma \in \mathbb{N}^{<\mathbb{N}}$ we let $T_\sigma = \{\, \rho \in T : \rho \subseteq \sigma \vee \sigma \subseteq \rho \,\}$.

Definition 2.11. $\leq_{\mathrm{KB}}$ is the usual *Kleene-Brouwer ordering* of $\mathbb{N}^{<\mathbb{N}}$: if $\sigma, \tau \in \mathbb{N}^{<\mathbb{N}}$, we let $\sigma \leq_{\mathrm{KB}} \tau$ if either $\sigma \supseteq \tau$ or there is some i such that $\sigma \restriction i = \tau \restriction i$ and $\sigma(i) < \tau(i)$.

The following is well-known (see e.g. [Sim99, Lemma V.1.3]).

Lemma 2.12. *Let $T \subseteq \mathbb{N}^{<\mathbb{N}}$ be a tree: T is well-founded (i.e. $[T] = \emptyset$) if and only if the linear ordering $(T, \leq_{\mathrm{KB}})$ is well-ordered. Moreover, if $f \colon \mathbb{N} \to T$ is a descending sequence with respect to $\leq_{\mathrm{KB}}$, there exists $Y \in [T]$ such that $Y \leq_T f'$.*

We will need some terminology to describe functions between partial orderings.

Definition 2.13. Let $f \colon P \to Q$ be a function, $\leq_P$ and $\leq_Q$ be partial orderings of P and Q respectively, with $<_P$ and $<_Q$ the corresponding strict orderings. We say that f is *$(<_P, <_Q)$-monotone* if for every $x, y \in P$ such that $x <_P y$ we have $f(x) <_Q f(y)$.

2.3. **Computability theory notation.** We use standard notation from computability theory. In particular, for a string $\sigma \in \mathbb{N}^{\leq\mathbb{N}}$, $\{e\}^\sigma(n)$ denotes the output of the eth Turing machine on input n, run with oracle σ, for at most $|\sigma|$ steps (where $|\sigma| = \infty$ when $\sigma \in \mathbb{N}^{\mathbb{N}}$). If this computation does not halt in less than $|\sigma|$ steps we write $\{e\}^\sigma(n)\uparrow$, otherwise we write $\{e\}^\sigma(n)\downarrow$. We write $\{e\}^\sigma_t(n)\downarrow$ if the computation halts in less than $\min(|\sigma|, t)$ steps.

Given $X, Y \subseteq \mathbb{N}$, the predicate $X = Y'$ is defined as usual:

$$X = Y' \iff \forall e (e \in X \leftrightarrow \{e\}^Y(e)\downarrow).$$

Definition 2.14. Given an ordinal β (or actually any presentation of a linear ordering with first element 0), we say that $X = Y^{(\beta)}$ if

$$X^{[0]} = Y \quad , \quad \forall \gamma < \beta \; (X^{[\gamma]} = X^{[<\gamma]\prime}) \; \text{ and } X = X^{[<\beta]}.$$

where $X^{[\gamma]} = \{\, y : \langle \gamma, y \rangle \in X \,\}$ and $X^{[<\gamma]} = \{\, \langle \delta, y \rangle : \delta < \gamma \;\&\; \langle \delta, y \rangle \in X \,\}$.

2.4. **Subsystems of second order arithmetic.** We refer the reader to [Sim99] for background information on subsystems of second order arithmetic. All subsystems we consider extend RCA_0 which consists of the axioms of ordered semi-ring, plus Δ^0_1-comprehension and Σ^0_1-induction. Adding set-existence axioms to RCA_0 we obtain WKL_0, ACA_0, ATR_0, and Π^1_1-CA_0, completing the so-called “big five” of reverse mathematics.

In this paper we are interested in ACA_0, ATR_0, and some theories which lie between these two. All these theories can be presented in terms of “jump-existence axioms”, as follows:

ACA_0: $\mathsf{RCA}_0 +\ \forall Y \exists X \; (X = Y')$
ACA_0': $\mathsf{RCA}_0 +\ \forall Y \forall n \exists X \; (X = Y^{(n)})$
ACA_0^+: $\mathsf{RCA}_0 +\ \forall Y \exists X \; (X = Y^{(\omega)})$
Π^0_β-CA_0: $\mathsf{RCA}_0 +\ \beta$ well-ordered $\wedge\, \forall Y \exists X \; (X = Y^{(\beta)})$,
where β is a presentation of a computable ordinal[2]
ATR_0: $\mathsf{RCA}_0 +\ \forall \alpha (\alpha$ well-ordered $\implies \forall Y \exists X \; (X = Y^{(\alpha)}))$

Notice that Π^0_1-CA_0 is ACA_0 and Π^0_ω-CA_0 is ACA_0^+. Π^0_β-CA_0 is strictly stronger than Π^0_γ-CA_0 if and only if $\beta \geq \gamma \cdot \omega$. In fact the ω-model $\bigcup_{\alpha < \gamma \cdot \omega} \{\, X : X \leq_T 0^{(\alpha)} \,\}$ satisfies Π^0_α-CA_0 for all $\alpha < \gamma \cdot \omega$, but not $\Pi^0_{\gamma \cdot \omega}$-$\mathsf{CA}_0$. Each theory in the above list is strictly stronger than the preceding ones if we assume $\beta \geq \omega^2$.

[2] The system Π^0_β-CA_0 is sometimes denoted by $(\Pi^0_1$-$\mathsf{CA}_0)_\beta$ in the literature.

ACA_0 and ATR_0 are well-known and widely studied: [Sim99] includes a chapter devoted to each of them and their equivalents. (The axiomatization of ATR_0 given above is equivalent to the usual one by [Sim99, Theorem VIII.3.15].) ACA_0^+ was introduced in [BHS87], where it was shown that it proves Hindman's Theorem in combinatorics (to this day it is unknown whether ACA_0^+ and Hindman's Theorem are equivalent). ACA_0^+ has also been used in [Sho06] (where it is proved that ACA_0^+ is equivalent to statements asserting the existence of invariants for Boolean algebras) and in [MM09] (where ACA_0^+ is used to prove a restricted version of Fraïssé's conjecture on linear orders). ACA_0' is also featured in [MM09]. The computation of its proof-theoretic ordinal, which turns out to be ε_ω, is due to Jäger (unpublished notes, a proof appears in [McA85], and a different proof is included in [Afs08]). The theories Π^0_β-CA_0 are natural generalizations of ACA_0^+.

3. Forward direction

In this section we prove the "forward direction"of Theorems 1.1, 1.5, 1.7, 1.9, and 1.2. The results in this section are already known (though often written in different settings) but we include them as our proofs illustrate how the iterates of the Turing jump relate with the epsilon and Veblen functions.

The following theorem is essentially contained in Hirst's proof [Hir94] of the closure of well-orderings under exponentiation in ACA_0.

Theorem 3.1. *If $\mathcal{X}$ is a Z-computable linear ordering, and $\boldsymbol{\omega}^{\mathcal{X}}$ has a Z-computable descending sequence, then Z' can compute a descending sequence in $\mathcal{X}$.*

Proof. Let $(a_k : k \in \mathbb{N})$ be a Z-computable descending sequence in $\boldsymbol{\omega}^{\mathcal{X}}$. We can write a_k in the form $\omega^{x_{k,0}} \cdot m_{k,0} + \omega^{x_{k,1}} \cdot m_{k,1} + \cdots + \omega^{x_{k,l_k}} \cdot m_{k,l_k}$ where each $m_{k,0} \in \mathbb{N}$ is positive and $x_{k,i} >_{\mathcal{X}} x_{k,i+1}$ for all $i < l_k$.

Using Z', we recursively define a function $f : \mathbb{N} \to \mathcal{X} \times \omega$ which is decreasing with respect to the lexicographic ordering $<_{\mathcal{X}\times\omega}$. (We use $x{\cdot}m$ to denote $\langle x, m\rangle \in \mathcal{X}\times\omega$.) Each $f(n)$ is of the form $x_{k,i} \cdot m_{k,i}$ for some k and $i \le l_k$. At the following step, when we define $f(n+1)$, either we increase k and leave i unchanged, or, if this is not possible, we keep k unchanged and increase i by one. We will have that if $f(n)$ is of the form $x_{k,i} \cdot m_{k,i}$, then $x_{h,j} \cdot m_{h,j} = x_{k,j} \cdot m_{k,j}$ for all $h > k$ and $j < i$.

Let $f(0) = x_{0,0} \cdot m_{0,0}$. Assuming we already defined $f(n) = x_{k,i} \cdot m_{k,i}$, we need to define $f(n+1)$. If there exist $h > k$ such that $x_{h,i} \cdot m_{h,i} <_{\mathcal{X}\times\omega} x_{k,i} \cdot m_{k,i}$, then let $f(n+1) = x_{h,i} \cdot m_{h,i}$ for the least such h. If $x_{h,i} \cdot m_{h,i} \geq_{\mathcal{X}\times\omega} x_{k,i} \cdot m_{k,i}$ for all $h > k$ then we must have $i < l_k$ (otherwise $a_k >_{\boldsymbol{\omega}^{\mathcal{X}}} a_{k+1}$ cannot hold) and we can let $f(n+1) = x_{k,i+1} \cdot m_{k,i+1}$.

It is then straightforward to obtain a f-computable, and hence Z'-computable, descending sequence in $\mathcal{X}$. □

The proof above produces an index for a Z'-computable descending subsequence in $\mathcal{X}$, uniformly in $\mathcal{X}$ and the Z-computable descending sequence in $\boldsymbol{\omega}^{\mathcal{X}}$.

Corollary 3.2. $\mathsf{ACA}_0 \vdash$ *WOP*$(\mathcal{X} \mapsto \boldsymbol{\omega}^{\mathcal{X}})$.

Proof. The previous proof can be formalized within ACA_0. □

Corollary 3.3. $\mathsf{ACA}_0' \vdash \forall n$ *WOP*$(\mathcal{X} \mapsto \boldsymbol{\omega}^{\langle n, \mathcal{X}\rangle})$.

Proof. Theorem 3.1 implies that, given n, if $\mathcal{X}$ is a Z-computable linear ordering, and $\boldsymbol{\omega}^{\langle n,\mathcal{X}\rangle}$ has a Z-computable descending sequence, then $Z^{(n)}$ can compute a descending sequence in $\mathcal{X}$. This can be formalized within ACA_0'. □

The following two theorems are new in the form they are stated. However, they can easily be obtained from the standard proof that ACA_0 proves that every ordinal below $\varphi_2(0)$ can be proved well-founded in ACA_0^+, and that every ordinal below Γ_0 can be proved well-ordered in Predicative Analysis [Fef64, Sch77].

Theorem 3.4. *If $\mathcal{X}$ is a Z-computable linear ordering, and $\boldsymbol{\varepsilon}_{\mathcal{X}}$ has a Z-computable descending sequence, then $Z^{(\omega)}$ can compute a descending sequence in $\mathcal{X}$.*

Proof. Let $(a_k : k \in \mathbb{N})$ be a Z-computable descending sequence in $\boldsymbol{\varepsilon}_{\mathcal{X}}$. If no constant term ε_x appears in a_0, then $a_0 < \omega^{\langle n_0, 0\rangle}$ for some n_0 so that we essentially have a descending sequence in $\boldsymbol{\omega}^{\langle n_0, 0\rangle}$. Then, applying n_0 times Theorem 3.1, we have that $Z^{(n_0)}$ computes a descending sequence in 0, a contradiction.

Thus we can let x_0 be the largest $x \in \mathcal{X}$ such that ε_x appears in a_0. It is not hard to prove by induction on terms that $\varepsilon_{x_0} \leq a_0 < \omega^{\langle n_0, \varepsilon_{x_0}+1\rangle}$ for some $n_0 \in \mathbb{N}$. By Lemma 2.6, $\boldsymbol{\varepsilon}_{\mathcal{X}} \restriction \omega^{\langle n_0, \varepsilon_{x_0}+1\rangle}$ is computably isomorphic to $\boldsymbol{\omega}^{\langle n_0, \boldsymbol{\varepsilon}_{\mathcal{X}} \restriction (\varepsilon_{x_0}+1)\rangle}$ and we can view the a_k's as elements of the latter. Using Theorem 3.1 n_0 times, we obtain a $Z^{(n_0)}$-computable descending sequence in $\boldsymbol{\varepsilon}_{\mathcal{X}} \restriction (\varepsilon_{x_0}+1)$. Noticing that the proof of Theorem 3.1 is uniform, we can apply this process again to the sequence we have obtained, and get an $x_1 <_{\mathcal{X}} x_0$ and a descending sequence in $\boldsymbol{\varepsilon}_{\mathcal{X}} \restriction (\varepsilon_{x_1}+1)$ computable in $Z^{(n_0+n_1)}$ for some $n_1 \in \mathbb{N}$. Iterating this procedure we obtain a $Z^{(\omega)}$-computable descending sequence $x_0 >_{\mathcal{X}} x_1 >_{\mathcal{X}} \dots$ in $\mathcal{X}$. □

Corollary 3.5. $\mathsf{ACA}_0^+ \vdash$ *WOP*$(\mathcal{X} \mapsto \boldsymbol{\varepsilon}_{\mathcal{X}})$.

Proof. The previous proof can be formalized within ACA_0^+. □

Theorem 3.6. *Let α be a Z-computable well-ordering. If $\mathcal{X}$ is a Z-computable linear ordering, and $\boldsymbol{\varphi}(\alpha, \mathcal{X})$ has a Z-computable descending sequence, then $Z^{(\omega^\alpha)}$ can compute a descending sequence in $\mathcal{X}$.*

Proof. By Z-computable transfinite recursion on α, we define a computable procedure that given a Z-computable index for a linear ordering $\mathcal{X}$ and for a descending sequence in $\boldsymbol{\varphi}(\alpha, \mathcal{X})$, it returns a $Z^{(\omega^\alpha)}$-computable index for a descending sequence in $\mathcal{X}$. Let $(a_k : k \in \mathbb{N})$ be a computable descending sequence in $\boldsymbol{\varphi}(\alpha, \mathcal{X})$. Let x_0 be the largest $x \in \mathcal{X}$ such that the constant term $\varphi_{\alpha,x}$ appears in a_0 (if no $\varphi_{\alpha,x}$ appears in a_0, just use 0 in place of φ_{α,x_0} in the argument below). It is not hard to prove by induction on terms that $\varphi_{\alpha,x_0} \leq a_0 < \varphi_{\beta_0}^{n_0}(\varphi_{\alpha,x_0}+1)$ for some $\beta_0 < \alpha$ and $n_0 \in \mathbb{N}$, (where $\varphi_\beta^{n_0}(z)$ is obtained by applying the φ_β function symbol n_0 times to z). It also not hard to show that $\boldsymbol{\varphi}(\alpha, \mathcal{X}) \restriction \varphi_{\beta_0}^{n_0}(\varphi_{\alpha,x_0}+1)$ is computably isomorphic to $\varphi^{n_0}(\beta_0, \boldsymbol{\varphi}(\alpha, \mathcal{X} \restriction x_0)+1)$ (where $\boldsymbol{\varphi}^{n_0}(\beta, \mathcal{Z})$ is obtained by applying the $\boldsymbol{\varphi}(\beta, \cdot)$-operator on linear orderings n_0 times to $\mathcal{Z}$). Using the induction hypothesis n_0 times, we obtain a $Z^{(\omega^{\beta_0} \cdot n_0)}$-computable descending sequence in $\boldsymbol{\varphi}(\alpha, \mathcal{X} \restriction x_0) + 1$. Then, we apply this process again to the sequence we have obtained, and get $x_1 <_{\mathcal{X}} x_0$ and a descending sequence in $\boldsymbol{\varphi}(\alpha, \mathcal{X} \restriction x_1)+1$ computable in $Z^{(\omega^{\beta_0} \cdot n_0 + \omega^{\beta_1} \cdot n_1)}$ for some $\beta_1 < \alpha$ and $n_1 \in \mathbb{N}$. Iterating this procedure we obtain a $Z^{(\omega^\alpha)}$ descending sequence $x_0 >_{\mathcal{X}} x_1 >_{\mathcal{X}} \dots$ in $\mathcal{X}$. □

Corollary 3.7. *Let α be a computable ordinal. Then* $\Pi^0_{\omega^\alpha}$-$\mathsf{CA}_0 \vdash$ *WOP*$(\mathcal{X} \mapsto \varphi(\alpha, \mathcal{X}))$.

Proof. The previous proof can be formalized within $\Pi^0_{\omega^\alpha}$-CA_0 for a fixed computable α. □

Corollary 3.8. $\mathsf{ATR}_0 \vdash$ *WOP*$(\mathcal{X} \mapsto \boldsymbol{\varphi}(\mathcal{X}, 0))$.

Proof. Let α be a well-ordering and assume, towards a contradiction, that there exists a descending sequence in $\boldsymbol{\varphi}(\alpha, 0)$. Let Z be a real such that both α and the descending sequence are Z-computable. By Theorem 3.6 $Z^{(\omega^\alpha)}$ (which exists in ATR_0) computes a descending sequence in 0, which is absurd. □

4. Ordinal exponentiation and the Turing Jump

In this section we give a proof of the second part of Theorem 1.3. Our proof is a slight modification of Hirst's proof, and prepares the ground for the generalizations in the following sections.

We start by defining a modification of the Turing jump operator with nicer combinatorial properties. We will then define two computable approximations to this jump operator, one from strings to strings, and the other one from trees to trees.

Definition 4.1. Given $Z \in \mathbb{N}^{\mathbb{N}}$, we define the sequence of *Z-true stages* as follows:

$$t_n = \max\{t_{n-1}+1, \mu t(\{n\}^Z_t(n)\downarrow)\},$$

starting with $t_{-1} = 1$ (so that $t_n \geq n+2$). If there is no t such that $\{n\}^Z_t(n)\downarrow$, then the above definition gives $t_n = t_{n-1}+1$. So, t_n is a stage where Z can correctly guess $Z' \restriction n+1$ because $\forall m \leq n(m \in Z' \iff \{m\}^{Z \restriction t_n}(m)\downarrow)$. With this in mind, we define the *Jump operator* to be the function $\mathcal{J}\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that for every $Z \in \mathbb{N}^{\mathbb{N}}$ and $n \in \mathbb{N}$,

$$\mathcal{J}(Z)(n) = Z \restriction t_n,$$

or equivalently

$$\mathcal{J}(Z) = \langle Z\restriction t_0, Z\restriction t_1, Z\restriction t_2, Z\restriction t_3, \dots\rangle$$

Here is a sample of this definition:

$$Z = \langle \underbrace{\underbrace{\underbrace{\underbrace{Z(0), Z(1)}_{\mathcal{J}(Z)(0)}, \overset{t_0}{Z(2)}, Z(3), Z(4), Z(5)}_{\mathcal{J}(Z)(1)}, \overset{t_1}{Z(6)}}_{\mathcal{J}(Z)(2)}, \overset{t_2}{Z(7)}, Z(8), Z(9), Z(10), Z(11)}_{\mathcal{J}(Z)(3)}, \overset{t_3}{Z(12)}, \cdots\rangle$$

Of course, $\mathcal{J}(Z) \equiv_T Z'$ for every Z as $n \in Z' \iff \{n\}^{\mathcal{J}(Z)(n)}(n)\downarrow$. So, from a computability viewpoint, there is no essential difference between $\mathcal{J}(Z)$ and the usual Z'.

Definition 4.2. The *Jump function* is the mapping $J\colon \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$ defined as follows. For $\sigma \in \mathbb{N}^{<\mathbb{N}}$, define $t_n = \max\{t_{n-1}+1, \mu t(\{n\}^{\sigma\restriction t}(n)\downarrow)\}$, starting with $t_{-1} = 1$ (so that $t_n \geq n+2$). Again, if there is no t such that $\{n\}^{\sigma \restriction t}(n)\downarrow$, then the above definition gives $t_n = t_{n-1}+1$. Let $J(\sigma) = \langle \sigma\restriction t_0, \sigma\restriction t_1, \dots, \sigma\restriction t_{k-1}\rangle$ where k is least such that $t_k > |\sigma|$.

Given $\tau \in J(\mathbb{N}^{<\mathbb{N}})$, we let $K(\tau)$ be the last entry of τ when $\tau \neq \emptyset$, and $K(\emptyset) = \emptyset$.

Remark 4.3. Since we can computably decide whether $\{n\}^{\sigma\restriction t}(n)\downarrow$, the Jump function is computable. The computability of K is obvious.

The following Lemma lists the key properties of J and K. We will refer to these properties as (P1), ..., (P6).

Lemma 4.4. *For every $\sigma, \tau' \in \mathbb{N}^{<\mathbb{N}}$ and $\tau \in J(\mathbb{N}^{<\mathbb{N}})$,*

(P1) *$J(\sigma) = \emptyset$ if and only if $|\sigma| \leq 1$.*
(P2) *$K(J(\sigma)) = \sigma$ when $|\sigma| \geq 2$.*
(P3) *$J(K(\tau)) = \tau$.*
(P4) *If $\sigma \neq \sigma'$ and at least one has length ≥ 2, then $J(\sigma) \neq J(\sigma')$.*
(P5) *$|J(\sigma)| < |\sigma|$ and $|K(\tau)| > |\tau|$ except when $\tau = \emptyset$.*
(P6) *If $\tau' \subset \tau$ then $\tau' \in J(\mathbb{N}^{<\mathbb{N}})$ and $K(\tau') \subset K(\tau)$.*

Proof. (P1) is obvious from the definition.

(P2) follows from the fact that, when $|\sigma| \geq 2$, $t_{k-1} = |\sigma|$ (using the notation of Definition 4.2). In fact $t_{k-1} \leq |\sigma|$ by definition of k, and if $t_{k-1} < |\sigma|$ then we have either $\{k\}^{\sigma\restriction t_k}(k)\downarrow$ (and hence $t_k \leq |\sigma|$) or $t_k = t_{k-1}+1 \leq |\sigma|$, against the definition of k.

(P3) follows from (P2) and $K(\emptyset) = \emptyset$.

(P4) follows immediately from (P1) and (P2).

The first part of (P5) follows from $t_n \geq n+2$. The second part is a consequence of the first, (P1) and (P2).

(P6) is obvious when $\tau' = \emptyset$, using the second part of (P5). Otherwise we have $\tau' = \langle \sigma\restriction t_0, \sigma\restriction t_1, \dots, \sigma\restriction t_j \rangle$ for some $j < k-1$, so that $K(\tau') = \sigma\restriction t_j \subset \sigma\restriction t_{k-1} = K(\tau)$. It is easy to check that $\tau' = J(\sigma\restriction t_j)$. □

The following Lemma explains how the Jump function approximates the Jump operator.

Lemma 4.5. *Given $Y, Z \in \mathbb{N}^{\mathbb{N}}$, the following are equivalent:*

1. $Y = \mathcal{J}(Z)$;
2. *for every n there exists $\sigma_n \subset Z$ with $|\sigma_n| > n$ such that $Y\restriction n = J(\sigma_n)$.*

Proof. Suppose first that $Y = \mathcal{J}(Z)$. When $n = 0$ let $\sigma_n = Z\restriction 1$, which works by (P1). When $n > 0$ let $\sigma_n = K(Y\restriction n) = K(Y\restriction n) = \mathcal{J}(Z)(n-1) \subset Z$. If $\{0\}^Z(0)\downarrow$ then $Y(0) \subset Z$ is such that $\{0\}^{Y(0)}(0)\downarrow$ and $Y(0) \subseteq \sigma_n$ so that also $\{0\}^{\sigma_n}(0)\downarrow$ and $J(\sigma_n)(0) = Y(0)$. If $\{0\}^Z(0)\uparrow$ then $Y(0) = Z\restriction 2 = \sigma_n\restriction 2 = J(\sigma_n)(0)$. This is the base step of an induction that, using the same argument, shows that $Y(i) = J(\sigma_n)(i)$ for every $i < n$. Thus $Y\restriction n \subseteq J(\sigma_n)$. By (P6), we have $Y\restriction n \in J(\mathbb{N}^{<\mathbb{N}})$ and we can apply (P3) and (P5) to obtain $Y\restriction n = J(\sigma_n)$ and $|\sigma_n| > n$.

Now assume that (2) holds, and suppose towards a contradiction that $Y \neq \mathcal{J}(Z)$. Let n be least such that $Y(n-1) \neq \mathcal{J}(Z)(n-1)$. If $\sigma_n \subset Z$ is such that $Y\restriction n = J(\sigma_n)$ we have $J(\sigma_n)(n-1) \neq \mathcal{J}(Z)(n-1)$. This can occur only if $\{n-1\}^{\sigma_n}(n-1)\uparrow$ and $\{n-1\}^Z(n-1)\downarrow$, which implies $n' > |\sigma_n|$, where $n' = |\mathcal{J}(Z)(n-1)|$. Notice that for any $m > n'$ we have $J(Z\restriction m)(n-1) = \mathcal{J}(Z)(n-1)$ and hence $J(Z\restriction m)(n-1) \neq Y(n-1)$. This contradicts the existence of $\sigma_{n'} \subset Z$ with $|\sigma_{n'}| > n'$ such that $Y\restriction n' = J(\sigma_{n'})$. □

The following corollary is obtained by iterating the Lemma.

Corollary 4.6. *For every $m > 0$, given $Y, Z \in \mathbb{N}^{\mathbb{N}}$, the following are equivalent:*

1. $Y = \mathcal{J}^m(Z)$;
2. *for every n there exists $\sigma_n \subset Z$ with $|\sigma_n| \geq n+m$ such that $Y\restriction n = J^m(\sigma_n)$.*

The Jump function leads to the definition of the Jump Tree.

Definition 4.7. Given a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$ we define the *Jump Tree of T* to be

$$\mathcal{JT}(T) = \{\, J(\sigma) : \sigma \in T \,\}.$$

The following lemmas summarize the main properties of the Jump Tree.

Lemma 4.8. *For every tree T, $\mathcal{JT}(T)$ is a tree computable in T.*

Proof. $\mathcal{JT}(T)$ is a tree because if $\tau \subset J(\sigma)$ for $\sigma \in T$, then $\tau = J(K(\tau))$ (by (P6) and (P3)) and $K(\tau) \in T$ (since by (P6), (P2) and (P1), $K(\tau) \subset K(J(\sigma)) \subseteq \sigma$).

$\mathcal{JT}(T)$ is computable in T because $\tau \in \mathcal{JT}(T)$ if and only if $\tau = J(K(\tau))$ (which is equivalent to $\tau \in J(\mathbb{N}^{<\mathbb{N}})$ by (P3)) and $K(\tau) \in T$. □

Lemma 4.9. *For every tree T, $[\mathcal{JT}(T)] = \{\, \mathcal{J}(Z) : Z \in [T] \,\}$.*

Proof. First let $Z \in [T]$. Since by Lemma 4.5 for every $n \in \mathbb{N}$, $\mathcal{J}(Z)\!\upharpoonright\! n = J(\sigma)$ for some $\sigma \subset Z$, so $\mathcal{J}(Z)\!\upharpoonright\! n \in \mathcal{JT}(T)$. This implies $\{\, \mathcal{J}(Z) : Z \in [T] \,\} \subseteq [\mathcal{JT}(T)]$.

To prove the other inclusion, fix $Y \in [\mathcal{JT}(T)]$, notice that $Y(n) \subset Y(n+1) \in \mathbb{N}^{<\mathbb{N}}$ for every n, and let $Z = \bigcup_{n\in\mathbb{N}} Y(n) \in \mathbb{N}^{\mathbb{N}}$. Observe that, again by Lemma 4.5, $Y = \mathcal{J}(Z)$ and $Z \in [T]$. □

We can now define the Z-computable linear ordering of theorem 1.3: let $\mathcal{X}_Z = \langle \mathcal{JT}(T_Z), \leq_{\mathrm{KB}} \rangle$ where $T_Z = \{\, Z\!\upharpoonright\! n : n \in \mathbb{N} \,\}$. Note that $\mathcal{X}_Z$ is indeed a linear ordering and, by Lemma 4.8, it is Z-computable. Since Z is the unique path in T_Z, by Lemma 4.9 $\mathcal{J}(Z)$ is the unique path in $\mathcal{JT}(T_Z)$. Moreover, for every $\tau = J(\sigma) \in \mathcal{JT}(T_Z)$ we have that either $\tau \subset \mathcal{J}(Z)$ or there is some i such that $\tau\!\upharpoonright\! i = \mathcal{J}(Z)\!\upharpoonright\! i$ and $\tau(i) \neq \mathcal{J}(Z)(i)$. This can only happen if $\{i\}^{\sigma}(i)\!\uparrow$ and $\{i\}^{Z}(i)\!\downarrow$, so that $\tau(i) \subset \mathcal{J}(Z)(i)$. By our assumption on the coding of strings, we have $\tau(i) < \mathcal{J}(Z)(i)$ and hence $\tau <_{\mathrm{KB}} \mathcal{J}(Z)\!\upharpoonright\!|\tau|$.

Let $\langle \tau_n \rangle_{n\in\mathbb{N}}$ be an infinite $<_{\mathrm{KB}}$-descending sequence in $\mathcal{JT}(T_Z)$. If $\tau_n \not\subset \mathcal{J}(Z)$ for some n then $\tau_m <_{\mathrm{KB}} \tau_n <_{\mathrm{KB}} \mathcal{J}(Z)\!\upharpoonright\!|\tau_m|$ for all $m > n$, which by Lemma 2.12 implies the existence of a path in $\mathcal{JT}(T_Z)$ different from $\mathcal{J}(Z)$, a contradiction. Therefore any infinite descending sequence in $\mathcal{X}_Z$ consists only of initial segments of $\mathcal{J}(Z)$ and hence computes $\mathcal{J}(Z) \equiv_T Z'$.

We still need to prove the existence of a Z-computable descending sequence in $\boldsymbol{\omega}^{\mathcal{X}_Z}$. To this end we use of the following function.

Definition 4.10. Let T be a tree and order $\mathcal{JT}(T)$ by $\leq_{\mathrm{KB}}$. Define $h\colon T \to \boldsymbol{\omega}^{\mathcal{JT}(T)}$ by

$$h(\sigma) = \left(\sum_{\substack{i<|J(\sigma)| \\ \{i\}^{\sigma}(i)\uparrow}} \omega^{J(\sigma)\upharpoonright i} \right) + \omega^{J(\sigma)} \cdot 2$$

for $\sigma \neq \emptyset$ and $h(\emptyset) = \omega^{\emptyset} \cdot 3$.

The sum above is written in $\leq_{\mathrm{KB}}$-decreasing order, so that indeed $h(\sigma) \in \boldsymbol{\omega}^{\mathcal{JT}(T)}$.

Since J is computable, h is computable as well.

The proof below should help the reader understand the motivation for the definition above.

Lemma 4.11. *h is $(\supset, <_{\boldsymbol{\omega}^{\mathcal{JT}(T)}})$-monotone.*

Proof. Suppose $\rho, \sigma \in T$ are such that $\rho \supset \sigma$; we want to show that $h(\rho) <_{\boldsymbol{\omega}^{\mathcal{JT}(T)}} h(\sigma)$.

If $\sigma = \emptyset$ then $\omega^{\emptyset}$ occurs with multiplicity 3 in $J(\sigma)$ and with multiplicity at most 2 in $J(\rho)$. Since $\emptyset$ is the $\leq_{\mathrm{KB}}$-maximum element in $\mathbb{N}^{<\mathbb{N}}$(and hence also in $\mathcal{JT}(T)$), this implies $h(\rho) <_{\boldsymbol{\omega}^{\mathcal{JT}(T)}} h(\sigma)$.

If $\sigma \neq \emptyset$ then $J(\sigma) \neq J(\rho)$ by (P4). Since $\sigma \subset \rho$, if $\{i\}^{\rho}(i)\!\uparrow$ then $\{i\}^{\sigma}(i)\!\uparrow$ as well. Thus there are two possibilities. If for all $i < |J(\sigma)|$, $\{i\}^{\sigma}(i)\!\uparrow$ whenever $\{i\}^{\rho}(i)\!\uparrow$ then $J(\sigma) \subset J(\rho)$ and the first difference between $h(\sigma)$ and $h(\rho)$ is the coefficient of $\omega^{J(\sigma)}$, which in $h(\sigma)$ is 2 and in $h(\rho)$ is either 1 or 0 (depending on whether $\{|J(\sigma)|\}^{\rho}(|J(\sigma)|)\!\uparrow$ or not). In any case, $h(\rho) <_{\boldsymbol{\omega}^{\mathcal{JT}(T)}} h(\sigma)$. If instead for some $i < |J(\sigma)|$, $\{i\}^{\sigma}(i)\!\uparrow$ and $\{i\}^{\rho}(i)\!\downarrow$ let i_0 be the least such i. Then the first difference between $h(\sigma)$ and $h(\rho)$ occurs at $\omega^{J(\sigma)\upharpoonright i_0)}$, which appears in $h(\sigma)$ but not in $h(\rho)$. Again, we have $h(\rho) <_{\boldsymbol{\omega}^{\mathcal{JT}(T)}} h(\sigma)$. □

We can now finish off the proof of the second part of Theorem 1.3. The sequence $\langle h(Z\!\upharpoonright\! n) \rangle_{n\in\mathbb{N}}$ is Z-computable and strictly decreasing in $\boldsymbol{\omega}^{\mathcal{X}_Z}$ by Lemma 4.11.

Obviously our proof yields the following generalization of the second part of Theorem 1.3.

Theorem 4.12. *For every real Z there exists a Z-computable linear ordering $\mathcal{X}$ with a Z-computable descending sequence in $\boldsymbol{\omega}^{\mathcal{X}}$ such that every descending sequence in $\mathcal{X}$ computes Z'.*

5. The ε function and the ω-Jump

In this section we extend the construction of Section 4. To iterate the construction, even only a finite number of times, requires generalizing the definition of h. Then we tackle the issue of extending the definition at limit ordinals by considering the ω-Jump.

5.1. Finite iterations of exponentiation and Turing Jump.

We start by defining a version of the function h used in the previous section that we can iterate.

Definition 5.1. Let $\mathcal{X}$ be a linear ordering, T a tree and

$$g\colon \mathcal{JT}(T) \to \mathcal{X}$$

a function. Define

$$h_g\colon T \to \boldsymbol{\omega}^{\mathcal{X}}$$

by

$$h_g(\sigma) = \left(\sum_{\substack{i<|J(\sigma)| \\ \{i\}^{\sigma}(i)\uparrow}} \omega^{g(J(\sigma)\restriction i)} \right) + \omega^{g(J(\sigma))} \cdot 2$$

for $\sigma \neq \emptyset$ and $h_g(\emptyset) = \omega^{g(\emptyset)} \cdot 3$.

Note that when g is the identity, then $h_g = h$ of the previous section. Also, h_g is g-computable.

Lemma 5.2. *If g is $(\supset, <_{\mathcal{X}})$-monotone, then h_g is $(\supset, <_{\boldsymbol{\omega}^{\mathcal{X}}})$-monotone.*

Proof. Notice that g $(\supset, <_{\mathcal{X}})$-monotone implies that the sum in the definition of $h_g(\sigma)$ is written in decreasing order. The proof is the same as the one for Lemma 4.11. □

We can now prove the analogue of Theorem 1.3 for iterations of the exponential (recall the notation $\boldsymbol{\omega}^{\langle n, \mathcal{X}\rangle}$ introduced in Definition 2.2).

Theorem 5.3. *For every $n \in \mathbb{N}$ and $Z \in \mathbb{N}^{\mathbb{N}}$, there is a Z-computable linear ordering $\mathcal{X}^n_Z$ such that the jump of every descending sequence in $\mathcal{X}^n_Z$ computes $Z^{(n)}$, but there is a Z-computable descending sequence in $\boldsymbol{\omega}^{\langle n, \mathcal{X}^n_Z\rangle}$.*

Proof. Letting again $T_Z = \{\, Z \restriction n : n \in \mathbb{N} \,\}$, we define a sequence $\langle T_i \rangle_{i \leq n}$ of trees as follows: let $T_0 = T_Z$ and $T_{i+1} = \mathcal{JT}(T_i)$ for every $i < n$. By induction on i, using Lemmas 4.8 and 4.9, we have that each T_i is a Z-computable tree and that the only path through T_i is $\mathcal{J}^i(Z)$ (i.e. the result of applying i times $\mathcal{J}$ starting with Z). We let $\mathcal{X}^n_Z = \langle T_n, \leq_{\mathrm{KB}} \rangle$, which is a Z-computable linear ordering. By Lemma 2.12 if f is a descending sequence in $\mathcal{X}^n_Z$ then $\mathcal{J}^n(Z) \leq_T f'$. Since $Z^{(n)} \equiv_T \mathcal{J}^n(Z)$, the first property of $\mathcal{X}^n_Z$ is proved.

To show that there is a Z-computable descending sequence in $\boldsymbol{\omega}^{\langle n, \mathcal{X}^n_Z\rangle}$ we define by recursion on $m \leq n$ functions $g_m\colon T_{n-m} \to \boldsymbol{\omega}^{\langle m, \mathcal{X}^n_Z\rangle}$. Let $g_0\colon T_n \to \mathcal{X}^n_Z$ be the identity function (T_n is indeed the domain of $\mathcal{X}^n_Z$). We define $g_{m+1}\colon T_{n-m-1} \to \boldsymbol{\omega}^{\langle m+1, \mathcal{X}^n_Z\rangle}$ by $g_{m+1} = h_{g_m}$ as in Definition 5.1. By induction on $m \leq n$, using

$$\begin{tikzcd}[column sep=tiny]
\mathcal{J}^\omega(Z) \arrow[d, equal] & = & Y & & & & & & & & & \\
\frown\,\vdots & & \vdots & & & & & & & & & \\
\mathcal{J}^4(Z)(0) & = & Y(3) & \subset & \cdots & & & & & & & \vdots \\
\mathcal{J}^3(Z)(0) & = & Y(2) & \subset & K(Y(3)) \arrow[ull, "J"'] & \subset & \cdots & & \cdots & & \cdots\subset & \mathcal{J}^2(Z) \\
\mathcal{J}^2(Z)(0) & = & Y(1) & \subset & K(Y(2)) \arrow[ull, "J"'] & \subset & K^2(Y(3)) \arrow[ull, "J"'] & \subset & \cdots & & \cdots\subset & \mathcal{J}(Z) \arrow[u, "\mathcal{J}"'] \\
\mathcal{J}(Z)(0) & = & Y(0) \arrow[d, equal] & \subset & K(Y(1)) \arrow[ull, "J"'] \arrow[d, equal] & \subset & K^2(Y(2)) \arrow[ull, "J"'] \arrow[d, equal] & \subset & K^3(Y(3)) \arrow[ull, "J"'] \arrow[d, equal] & \subset & \cdots\subset & Z \arrow[u, "\mathcal{J}"'] \arrow[d, equal] \\
\smile & & K^\omega(Y{\restriction}1) & \subset & K^\omega(Y{\restriction}2) & \subset & K^\omega(Y{\restriction}3) & \subset & K^\omega(Y{\restriction}4) & \subset & \cdots\subset & \mathcal{K}^\omega(Y)
\end{tikzcd}$$

FIGURE 1. Assuming $Y = \mathcal{J}^\omega(Z)$.

Lemma 5.2, it is immediate that each g_m is $(\supset, <_{\boldsymbol{\omega}^{\langle m,T_n\rangle}})$-monotone and computable. Hence the sequence $\langle g_n(Z \restriction j)\rangle_{j\in\mathbb{N}}$ in $\boldsymbol{\omega}^{\langle n,\mathcal{X}_Z^n\rangle}$ is Z-computable and descending. $\square$

5.2. **The ω-Jump.** Now we define the iteration of the Jump operator at the first limit ordinal ω. Again, our definition is slightly different than the usual one so that it has nicer combinatorial properties. The difference being that instead of pasting all the $\mathcal{J}^i(Z)$ together as columns, we will take only the first value of each. Later we will show that this is enough. We will also define two computable approximations to this ω-jump operator, one from strings to strings, and the other one from trees to trees, and a computable inverse function.

Definition 5.4. We define the *ω-Jump operator* to be the function $\mathcal{J}^\omega\colon \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that for every $Z \in \mathbb{N}^\mathbb{N}$

$$\mathcal{J}^\omega(Z) = \langle \mathcal{J}(Z)(0),\ \mathcal{J}^2(Z)(0),\ \mathcal{J}^3(Z)(0), \dots\rangle,$$

or, in other words, $\mathcal{J}^\omega(Z)(n) = \mathcal{J}^{n+1}(Z)(0)$.

Notice that $\mathcal{J}^\omega(\mathcal{J}(Z))$ equals $\mathcal{J}^\omega(Z)$ with the first element removed. Before showing that $\mathcal{J}^\omega(Z) \equiv_T Z^{(\omega)}$ it is convenient to define the inverse of $\mathcal{J}^\omega$.

Definition 5.5. Given $Y \in \mathcal{J}^\omega(\mathbb{N}^\mathbb{N})$ we define

$$\mathcal{K}^\omega(Y) = \bigcup_n K^n(Y(n)).$$

We need to show that the union above makes sense. Assume $Y = \mathcal{J}^\omega(Z)$. It might help to look at Figure 1. Notice that for each n, $Y(n) \subset \mathcal{J}^n(Z)$ because for every X, $\mathcal{J}(X)(0) \subset X$ and $Y(n) = \mathcal{J}(\mathcal{J}^n(Z))(0)$. We also know that if $\sigma \subset \mathcal{J}(X)$, then $K(\sigma) \subset X$, so that $K^n(Y(n)) \subset Z$. It follows that $\bigcup_n K^n(Y(n)) \subseteq Z$. Applying (P5) n times we get that $|K^n(Y(n))| > n$, and therefore the union above does actually produce Z. We have just proved the following lemma.

Lemma 5.6. *For every $Z \in \mathbb{N}^\mathbb{N}$, $\mathcal{K}^\omega(\mathcal{J}^\omega(Z)) = Z$.*

Lemma 5.7. *For every $Z \in \mathbb{N}^\mathbb{N}$, $\mathcal{J}^\omega(Z) \equiv_T Z^{(\omega)}$.*

Proof. We already know that $Z^{(n)} \equiv_T \mathcal{J}^n(Z)$ uniformly in n and hence that $Z^{(\omega)} = \bigoplus_{n\in\mathbb{N}} Z^{(n)} \equiv_T \bigoplus_{n\in\mathbb{N}} \mathcal{J}^n(Z)$. It immediately follows that $\mathcal{J}^\omega(Z) \leq_T Z^{(\omega)}$.

For the other direction we need to uniformly compute all the reals $\mathcal{J}^n(Z)$ from $\mathcal{J}^\omega(Z)$. We do this as follows. Given $Y \in \mathbb{N}^\mathbb{N}$, let Y^{-n} be Y with its first n elements removed. Then, $\mathcal{J}^\omega(\mathcal{J}^n(Z)) = \mathcal{J}^\omega(Z)^{-n}$. By the lemma above we get that $\mathcal{J}^n(Z) = \mathcal{K}^\omega(\mathcal{J}^\omega(Z)^{-n})$, which we can compute uniformly from $\mathcal{J}^\omega(Z)$. □

As in section 4, where we computably approximated the jump operator, we will now approximate the ω-Jump operator with a computable operation on finite strings.

Definition 5.8. The *ω-Jump function* is the map $J^\omega\colon \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$ defined as follows. Given $\sigma \in \mathbb{N}^{<\mathbb{N}}$, let

$$J^\omega(\sigma) = \langle J(\sigma)(0),\ J^2(\sigma)(0), \dots, J^{n-1}(\sigma)(0)\rangle,$$

where n is the least such that $J^n(\sigma) = \emptyset$ (there is always such an n, because, by (P5), $|J^i(\sigma)| \leq |\sigma| - i$ for $i \leq |\sigma|$). Note that then, by (P1), $|J^{n-1}(\sigma)| = 1$.

J^ω is computable (because J is computable) and we will now define its computable partial inverse K^ω.

Definition 5.9. Given $\tau \in J^\omega(\mathbb{N}^{<\mathbb{N}})$, let $K^\omega(\tau) = K^{|\tau|}(\ell(\tau))$ (recall that $\ell(\tau) = \langle \tau(|\tau|-1|)\rangle$) when $\tau \neq \emptyset$, and $K^\omega(\emptyset) = \emptyset$.

The following properties are the analogues of those of Lemma 4.4 for the ω-Jump function and its inverse. We will refer to them as (P$^\omega$1), …, (P$^\omega$7).

Lemma 5.10. *For $\sigma, \sigma', \tau' \in \mathbb{N}^{<\mathbb{N}}$, $\tau \in J^\omega(\mathbb{N}^{<\mathbb{N}})$,*

(P$^\omega$1) *$J^\omega(\sigma) = \emptyset$ if and only if $|\sigma| \leq 1$.*
(P$^\omega$2) *$K^\omega(J^\omega(\sigma)) = \sigma$ for $|\sigma| \geq 2$.*
(P$^\omega$3) *$J^\omega(K^\omega(\tau)) = \tau$.*
(P$^\omega$4) *If $\sigma \neq \sigma'$ and at least one has length ≥ 2, then $J^\omega(\sigma) \neq J^\omega(\sigma')$.*
(P$^\omega$5) *$|J^\omega(\sigma)| < |\sigma|$ and $|K^\omega(\tau)| > |\tau|$ except when $\tau = \emptyset$.*
(P$^\omega$6) *If $\tau' \subset \tau$ then $\tau' \in J^\omega(\mathbb{N}^{<\mathbb{N}})$ and $K^\omega(\tau') \subset K^\omega(\tau)$.*
(P$^\omega$7) *If $J^\omega(\sigma') \subseteq J^\omega(\sigma)$ then, for every m, $J^m(\sigma') \subseteq J^m(\sigma)$.*

Proof. (P$^\omega$1) follows from (P1) and the fact that $J^\omega(\sigma) = \emptyset$ is equivalent to $J(\sigma) = \emptyset$.

To prove (P$^\omega$2) let $|J^\omega(\sigma)| = n > 0$. Then $\ell(J^\omega(\sigma)) = \langle J^n(\sigma)(0)\rangle = J^n(\sigma)$ because $|J^n(\sigma)| = 1$ as noticed in the definition of J^ω. Thus $K^\omega(J^\omega(\sigma)) = K^n(J^n(\sigma)) = \sigma$ by (P2).

(P$^\omega$3) follows from (P$^\omega$2) and $K(\emptyset) = \emptyset$. (P$^\omega$4) follows immediately from (P$^\omega$1) and (P$^\omega$2). The first part of (P$^\omega$5) is immediate because by (P5) we have $J^{|\sigma|}(\sigma) = \emptyset$. The second part of (P$^\omega$5) is a consequence of the first, (P$^\omega$1) and (P$^\omega$2).

For (P$^\omega$6) look at Figure 2. (P$^\omega$6) is obvious when $\tau' = \emptyset$, using the second part of (P$^\omega$5). Otherwise, let σ be such that $\tau = J^\omega(\sigma)$. The idea is to define $\sigma' \subset \sigma$ as in the picture and then show that $\tau' = J^\omega(\sigma')$. Notice that $|\sigma| > |\tau| \geq 2$ by (P$^\omega$5), and that $\sigma = K^\omega(\tau)$ by (P$^\omega$2). Notice also that

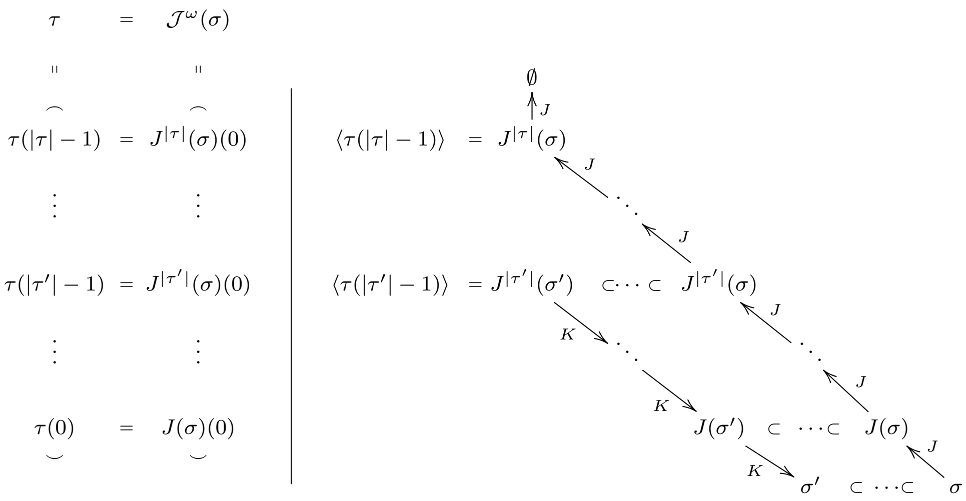
$$\ell(\tau') = \langle \tau(|\tau'|-1)\rangle = \langle J^{|\tau'|}(\sigma)(0)\rangle \subset J^{|\tau'|}(\sigma)$$

because $|\tau'| < |\tau|$ and hence $|J^{|\tau'|}(\sigma)| > 1$. Let $\sigma' = K^{|\tau'|}(\ell(\tau'))$. Using (P6) $|\tau'|$ times we know that $\sigma' \subset \sigma$ and $J^{|\tau'|}(\sigma') = \ell(\tau')$. Now, we need to show that $J^\omega(\sigma') = \tau'$. First notice that $|J^\omega(\sigma')| = |\tau'|$ because $|J^{|\tau'|}(\sigma')| = |\ell(\tau')| = 1$. By induction on $i \leq |\tau'|$ we can show, using (P6) and (P2), that

$$J^{|\tau'|-i}(\sigma') = K^i(\ell(\tau')) \subset K^i(J^{|\tau'|}(\sigma)) = J^{|\tau'|-i}(\sigma). \tag{5.1}$$

Now for $j < |\tau'|$, $J^\omega(\sigma')(j) = J^{j+1}(\sigma')(0) = J^{j+1}(\sigma)(0) = \tau(j) = \tau'(j)$.

For (P$^\omega$7) let $\tau' = J^\omega(\sigma')$. Then, if $i = |\tau'| - m$, equation (5.1) shows that $J^m(\sigma') \subseteq J^m(\sigma)$. □



$\tau = \mathcal{J}^\omega(\sigma)$
$\tau(|\tau|-1) = J^{|\tau|}(\sigma)(0)$
$\tau(|\tau'|-1) = J^{|\tau'|}(\sigma)(0)$
$\tau(0) = J(\sigma)(0)$
$\emptyset$
J
$\langle\tau(|\tau|-1)\rangle = J^{|\tau|}(\sigma)$
$\langle\tau(|\tau'|-1)\rangle = J^{|\tau'|}(\sigma') \subset\cdots\subset J^{|\tau'|}(\sigma)$
K
$J(\sigma') \subset \cdots\subset J(\sigma)$
$\sigma' \subset \cdots\subset \sigma$

FIGURE 2. Assuming $\tau = J^\omega(\sigma)$ and $\tau' \subset \tau$.

As we did in Section 4, we now explain how the ω-Jump function approximates the ω-Jump operator.

Lemma 5.11. *Given $Y, Z \in \mathbb{N}^\mathbb{N}$, the following are equivalent:*

(1) *$Y = \mathcal{J}^\omega(Z)$;*

(2) *for every n there exists $\sigma_n \subset Z$ with $|\sigma_n| > n$ such that $Y \restriction n = J^\omega(\sigma_n)$.*

Proof. First assume $Y = \mathcal{J}^\omega(Z)$. When $n = 0$ let $\sigma_0 = Z \restriction 1$, which works by (P$^\omega$1). When $n > 0$ let $\sigma_n = K^\omega(Y \restriction n)$. We recommend the reader to look at Figure 1 again. By (P$^\omega$3) we have $Y \restriction n = J^\omega(\sigma_n)$. Since $\sigma_n = K^n(\langle Y(n-1)\rangle) = K^{n-1}(Y(n-1))$, σ_n is one of the strings occurring in the definition of $\mathcal{K}^\omega(Y)$ and hence $\sigma_n \subset \mathcal{K}^\omega(Y) = Z$ by Lemma 5.6. We get that $|\sigma_n| > n$ by (P5) applied n times to $\sigma_n = K^n(\langle Y(n-1)\rangle)$.

Suppose now that (2) holds. By (P$^\omega$7), we get that for all m and n, $J^m(\sigma_n) \subseteq J^m(\sigma_{n+1})$. Using Corollary 4.6, it is straightforward to show that for all $m < n$, $\bigcup_n J^m(\sigma_n) = \mathcal{J}^m(Z)$ and hence $J^m(\sigma_n)(0) = \mathcal{J}^m(Z)(0)$. It follows that for every m and $n > m$

$$\mathcal{J}^\omega(Z)(m) = \mathcal{J}^{m+1}(Z)(0) = J^{m+1}(\sigma_n)(0) = J^\omega(\sigma_n)(m) = Y(m). \qquad \square$$

Again as in Section 4, the ω-Jump function leads to the definition of the ω-Jump Tree.

Definition 5.12. Given a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$ the *ω-Jump Tree of T* is

$$\mathcal{JT}^\omega(T) = \{\, J^\omega(\sigma) : \sigma \in T \,\}.$$

Lemma 5.13. *For every tree T, $\mathcal{JT}^\omega(T)$ is a tree computable in T.*

Proof. The proof is identical to the proof of Lemma 4.8, using Lemma 5.10 in place of Lemma 4.4. □

Lemma 5.14. *For every tree T, $[\mathcal{JT}^\omega(T)] = \{\, \mathcal{J}^\omega(Z) : Z \in [T] \,\}$.*

Proof. To prove $\{\, \mathcal{J}^\omega(Z) : Z \in [T] \,\} \subseteq [\mathcal{JT}^\omega(T)]$ we can argue as in the proof of Lemma 4.9, using Lemma 5.11 in place of Lemma 4.5.

To prove the other inclusion, fix $Y \in [\mathcal{JT}^\omega(T)]$. For each n, let $\sigma_n = K^\omega(Y \restriction n) \in T$. Since $J^\omega(\sigma_n) = Y \restriction n$ by (P$^\omega$3), we have $\sigma_n \subset \sigma_{n+1}$ for each n by (P$^\omega$6). Let $Z = \bigcup_{n\in\mathbb{N}} \sigma_n \in [T]$. Then, by Lemma 5.11 we get $Y = \mathcal{J}^\omega(Z)$. □

5.3. **ω-Jumps versus the epsilon function.** Our goal now is to generalize Definition 5.1 with an operator that uses $\boldsymbol{\varepsilon}$ rather than $\boldsymbol{\omega}$. We thus wish to define an operator h^ω that, given an order preserving function $g\colon \mathcal{JT}^\omega(T) \to \mathcal{X}$ (where $\mathcal{X}$ is a linear order), returns an order preserving function $h^\omega_g\colon T \to \boldsymbol{\varepsilon}_{\mathcal{X}}$. To do so we will iterate the h operator of Definition 5.1 along the elements of $\mathcal{JT}^\omega(T)$.

Let us give the rough motivation behind the definition of the operator h^ω below. Suppose we are given an order preserving function $g\colon \mathcal{JT}^\omega(T) \to \mathcal{X}$. For each i, we would like to define a monotone function $f_i\colon \mathcal{JT}^i(T) \to \boldsymbol{\varepsilon}_{\mathcal{X}}$ such that $f_i = h_{f_{i+1}}$, where $h_{f_{i+1}}$ is as in Definition 5.1. Notice that the range of this function is correct, using the fact that $\boldsymbol{\omega}^{\boldsymbol{\varepsilon}_{\mathcal{X}}}$ is computably isomorphic to $\boldsymbol{\varepsilon}_{\mathcal{X}}$. However, we do not have a place to start as to define such f_i we would need f_{i+1}, and this recursion goes the wrong way. Note that if $\tau = J^\omega(\sigma) \in \mathcal{JT}^\omega(T)$, then $\langle \tau(i)\rangle \in \mathcal{JT}^i(T)$, and we could use g to define f_i at least on the strings of length 1, of the form $\langle \tau(i)\rangle$. (This is not exactly what we are going to do, but it should help picture the construction.) The good news is that to calculate $f_{i-1} = h_{f_i}$ on strings of length at most 2, we only need to know the values of f_i on strings of length at most 1. Inductively, this would allow us to calculate $f_0\colon T \to \mathcal{X}$ on strings of length at most i. Since this would work for all i, we get f_0 defined on all T. We now give the precise definition.

First, we need to iterate the Jump Tree operator along any finite string.

Definition 5.15. If T is a tree we define

$$\mathcal{JT}^\omega_\tau(T) = \{\, J^{|\tau|+1}(\sigma) : \sigma \in T \wedge \tau \subseteq J^\omega(\sigma) \,\}.$$

Notice that $\mathcal{JT}^\omega_\tau(T) \subseteq \mathcal{JT}^{|\tau|+1}(T)$ and that $\mathcal{JT}^\omega_\tau(T)$ is empty when $\tau \notin \mathcal{JT}^\omega(T)$. The following Lemma provides an alternative way of defining $\mathcal{JT}^\omega_\tau(T)$ by an inductive definition.

Lemma 5.16. *Given a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$,*

$$\mathcal{JT}^\omega_\emptyset(T) = \mathcal{JT}(T)$$
$$\mathcal{JT}^\omega_{\tau^\frown\langle c\rangle}(T) = \mathcal{JT}(\mathcal{JT}^\omega_\tau(T)_{\langle c\rangle}).$$

($T_{\langle c\rangle}$ was defined in 2.10 as $\{\, \rho \in T : \langle c\rangle \subseteq \rho \vee \rho = \emptyset \,\}$.)

Proof. Straightforward induction on $|\tau|$. □

The next Lemma links $\mathcal{JT}^\omega_\tau(T)$ to $\mathcal{JT}^\omega(T)$.

Lemma 5.17. *Given a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$, $\tau \in \mathbb{N}^{<\mathbb{N}}$, and $c \in \mathbb{N}$,*

$$\tau^\frown\langle c\rangle \in \mathcal{JT}^\omega(T) \iff \langle c\rangle \in \mathcal{JT}^\omega_\tau(T).$$

Proof. This follows immediately from the definitions of $\mathcal{JT}^\omega(T)$ and $\mathcal{JT}^\omega_\tau(T)$. □

Definition 5.18. Let $\mathcal{X}$ be a linear ordering and $g\colon \mathcal{JT}^\omega(T) \to \mathcal{X}$ be a function. We define simultaneously for each $\tau \in \mathcal{JT}^\omega(T)$ a function

$$f_\tau\colon \mathcal{JT}^\omega_\tau(T) \to \boldsymbol{\varepsilon}_{\mathcal{X}}$$

by recursion on $|\sigma|$:

$$f_\tau(\sigma) = \begin{cases} \varepsilon_{g(\tau)} & \text{if } \sigma = \emptyset; \\ h_{f_{\tau^\frown\langle\sigma(0)\rangle}}(\sigma) & \text{if } \sigma \neq \emptyset. \end{cases}$$

Here $h_{f_{\tau^\frown\langle\sigma(0)\rangle}}$ is defined according to Definition 5.1. We then define

$$h^\omega_g = h_{f_\emptyset}\colon T \to \boldsymbol{\varepsilon}_{\mathcal{X}}.$$

Remark 5.19. First of all notice that we are really doing a recursion on $|\sigma|$. In fact, to compute $h_{f_{\tau^\frown\langle\sigma(0)\rangle}}(\sigma)$ when $\sigma \neq \emptyset$ we use $f_{\tau^\frown\langle\sigma(0)\rangle}$ on strings of the form $J(\sigma)\restriction i$, which have length $\leq |J(\sigma)| < |\sigma|$ by (P5).

Let us notice the functions have the right domains and ranges. The proof is done simultaneously for all $\tau \in \mathcal{JT}^\omega(T)$ by induction on $|\sigma|$. Take $\sigma \in \mathcal{JT}^\omega_\tau(T)$ with $|\sigma| = n$. Suppose that for all $\tau' \in \mathcal{JT}^\omega(T)$ and all $\sigma' \in \mathcal{JT}^\omega_{\tau'}(T)$ with $|\sigma'| < n$ we have that $f_{\tau'}(\sigma')$ is defined and $f_{\tau'}(\sigma') \in \boldsymbol{\varepsilon}_{\mathcal{X}}$.

If $\sigma = \emptyset$, then $f_\tau(\sigma) = \varepsilon_{g(\tau)} \in \boldsymbol{\varepsilon}_{\mathcal{X}}$. Suppose $\sigma \neq \emptyset$ and let $\tau' = \tau^\frown\langle\sigma(0)\rangle$. Then $f_\tau(\sigma) = h_{f_{\tau'}}(\sigma)$. When computing $h_{f_{\tau'}}(\sigma)$, we only apply $f_{\tau'}$ to strings of the form $J(\sigma)\restriction i$. These strings have length less than n and, by Lemma 4.8, belong to $\mathcal{JT}(\mathcal{JT}^\omega_\tau(T)_{\langle\sigma(0)\rangle}) = \mathcal{JT}^\omega_{\tau'}(T)$ (by Lemma 5.16). By the induction hypothesis we have that $f_{\tau'}$ is defined on these strings and takes values in $\boldsymbol{\varepsilon}_{\mathcal{X}}$. Therefore, $h_{f_{\tau'}}(\sigma)$ is defined and $h_{f_{\tau'}}(\sigma) \in \boldsymbol{\omega}^{\boldsymbol{\varepsilon}_{\mathcal{X}}}$. Using that $\boldsymbol{\omega}^{\boldsymbol{\varepsilon}_{\mathcal{X}}} = \boldsymbol{\varepsilon}_{\mathcal{X}}$, we get that $f_\tau \colon \mathcal{JT}^\omega_\tau(T) \to \boldsymbol{\varepsilon}_{\mathcal{X}}$.

Finally, since $f_\emptyset \colon \mathcal{JT}(T) \to \boldsymbol{\varepsilon}_{\mathcal{X}}$, we get that $h^\omega_g \colon T \to \boldsymbol{\varepsilon}_{\mathcal{X}}$.

Lemma 5.20. *If $g\colon \mathcal{JT}^\omega(T) \to \mathcal{X}$ is $(\supset, <_{\mathcal{X}})$-monotone, then $h^\omega_g \colon T \to \boldsymbol{\varepsilon}_{\mathcal{X}}$ is $(\supset, <_{\boldsymbol{\varepsilon}_{\mathcal{X}}})$-monotone.*

Proof. First, we note that by Lemma 5.2, it suffices to show that $f_\emptyset$ is $(\supset, <_{\boldsymbol{\varepsilon}_{\mathcal{X}}})$-monotone. We will actually show that for every $\tau \in \mathcal{JT}^\omega(T)$, f_τ is $(\supset, <_{\boldsymbol{\varepsilon}_{\mathcal{X}}})$-monotone.

The proof is again done simultaneously for all $\tau \in \mathcal{JT}^\omega(T)$ by induction on the length of the strings. Suppose that on strings of length less than n, for every τ', $f_{\tau'}$ is $(\supset, <_{\boldsymbol{\varepsilon}_{\mathcal{X}}})$-monotone. Let $\sigma' \subset \sigma \in \mathcal{JT}^\omega_\tau(T)$ with $|\sigma| = n$. Let $\tau' = \tau^\frown\langle\sigma(0)\rangle$. Consider first the case when $\sigma' = \emptyset$. Then $f_\tau(\sigma') = \varepsilon_{g(\tau)}$ while $f_\tau(\sigma)$ is a finite sum of terms of the form $\omega^{f_{\tau'}(J(\sigma)\restriction i)}$. By the induction hypothesis, the exponent of each such term is less than or equal to $f_{\tau'}(\emptyset) = \varepsilon_{g(\tau')} <_{\boldsymbol{\varepsilon}_{\mathcal{X}}} \varepsilon_{g(\tau)}$. So, the whole sum is less than $\varepsilon_{g(\tau)} = f_\tau(\sigma')$. Suppose now that $\sigma' \neq \emptyset$. Since the proof of Lemma 5.2 (based on the proof of Lemma 4.11) uses the monotonicity of $f_{\tau'}$ only for strings shorter then σ (by (P5)), we get that $h_{f_{\tau'}}(\sigma') >_{\boldsymbol{\varepsilon}_{\mathcal{X}}} h_{f_{\tau'}}(\sigma)$. □

Theorem 5.21. *For every $Z \in \mathbb{N}^\mathbb{N}$, there is a Z-computable linear ordering $\mathcal{X}$ such that the jump of every descending sequence in $\mathcal{X}$ computes $Z^{(\omega)}$, but there is a Z-computable descending sequence in $\boldsymbol{\varepsilon}_{\mathcal{X}}$.*

Proof. Let $\mathcal{X} = \langle \mathcal{JT}^\omega(T_Z), \leq_{\mathrm{KB}}\rangle$ where again $T_Z = \{\, Z\restriction n : n \in \mathbb{N}\,\}$. By Lemma 5.13, $\mathcal{X}$ is Z-computable. By Lemma 5.14, $\mathcal{J}^\omega(Z)$ is the unique path in $\mathcal{JT}^\omega(T_Z)$. Therefore, by Lemma 2.12, the jump of every descending sequence in $\mathcal{X}$ computes $\mathcal{J}^\omega(Z) \equiv_T Z^{(\omega)}$.

Let g be the identity on $\mathcal{X}$, which is obviously $(\supset, <_{\mathcal{X}})$-monotone. By Lemma 5.20, h^ω_g is $(\supset, <_{\boldsymbol{\varepsilon}_{\mathcal{X}}})$-monotone. Since h^ω_g is computable, $\{\, h^\omega_g(Z\restriction n) : n \in \mathbb{N}\,\}$ is a Z-computable descending sequence in $\boldsymbol{\varepsilon}_{\mathcal{X}}$. □

5.4. **Reverse mathematics results.** In this section, we work in the weak system RCA_0. Therefore, we do not have an operation that given $Z \in \mathbb{N}^{<\mathbb{N}}$, returns $\mathcal{J}(Z)$, let alone $\mathcal{J}^\omega(Z)$. However, the predicates with two variables Z and Y that say $Y = \mathcal{J}(Z)$ and $Y = \mathcal{J}^\omega(Z)$ are arithmetic as witnessed by Lemmas 4.5 and 5.11. Notice that if condition (2) of Lemma 4.5 holds, then RCA_0 can recover the sequence of t_i's in the definition of $\mathcal{J}(Z)$ and show that $\mathcal{J}(Z)$ is as defined in 4.1. Furthermore, RCA_0 can show that $\mathcal{J}(Z) \equiv_T Z'$ and hence that ACA_0 is equivalent to $\mathsf{RCA}_0 + \forall Z \exists Y (Y = \mathcal{J}(Z))$, and ACA_0' is equivalent to $\mathsf{RCA}_0 + \forall Z \forall n \exists Y (Y = \mathcal{J}^n(Z))$.

Also, if condition (2) of Lemma 5.11 holds, then as in the proof of that lemma, in RCA_0 we can uniformly build $\mathcal{J}^m(Z)$ as $\bigcup_n J^m(\sigma_n)$, and show that $\mathcal{J}^\omega(Z)$ is as defined in Definition 5.8. Furthermore, we can prove Lemma 5.7 in RCA_0: if $Y = \mathcal{J}^\omega(Z)$, then Y can compute $Z^{(\omega)}$, and if $X = Z^{(\omega)}$, then X can compute a real Y such that $Y = \mathcal{J}^\omega(Z)$. Therefore, we get that ACA_0^+ is equivalent to $\mathsf{RCA}_0+\forall Z \exists Y (Y = \mathcal{J}^\omega(Z))$.

We already know, from Girard's result Theorem 1.1 that over RCA_0, the statement "if $\mathcal{X}$ is a well-ordering then $\boldsymbol{\omega}^{\mathcal{X}}$ is a well-ordering" is equivalent to ACA_0. We now start climbing up the ladder.

Theorem 5.22. *Over* RCA_0*,* $\forall n$ *WOP*$(\mathcal{X} \mapsto \boldsymbol{\omega}^{\langle n, \mathcal{X}\rangle})$ *is equivalent to* ACA_0'*.*

Proof. We showed, in Corollary 3.3, that $\mathsf{ACA}_0' \vdash \forall n \, \mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\omega}^{\langle n, \mathcal{X}\rangle})$.

Suppose now that $\forall n \, \mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\omega}^{\langle n, \mathcal{X}\rangle})$ holds. Consider $Z \in \mathbb{N}^\mathbb{N}$ and $n \in \omega$; we want to show that $\mathcal{J}^n(Z)$ exists. By Girard's theorem we can assume ACA_0. Let $\mathcal{X}_Z^n = \langle T_n, \leq_{\mathrm{KB}}\rangle$, where $T_n = \mathcal{JT}^n(T_Z)$ as in the proof of Theorem 5.3. The proof that there is a Z-computable descending sequence in $\boldsymbol{\omega}^{\langle n, \mathcal{X}_Z^n\rangle}$ is finitary and goes through in RCA_0. So, by $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\omega}^{\langle n, \mathcal{X}\rangle})$ we get a descending sequence in $\mathcal{X}_Z^n$. By Lemma 2.12, using ACA_0, we get $Y_n \in [T_n]$. For each $i \leq n$, let $Y_i = \mathcal{K}^{n-i}(Y_n)$. Lemma 4.9 shows that for each i, $Y_i \in [T_i]$ and $Y_i = \mathcal{J}(Y_{i-1})$. Since Z is the only path through T_Z, we get that $Y_0 = Z$, and so $Y_n = \mathcal{J}^n(Z)$. □

Theorem 5.23. *Over* RCA_0*,* *WOP*$(\mathcal{X} \mapsto \boldsymbol{\varepsilon}_{\mathcal{X}})$ *is equivalent to* ACA_0^+*.*

Proof. We already showed that ACA_0^+ proves $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varepsilon}_{\mathcal{X}})$ in Corollary 3.5.

Assume $\mathsf{RCA}_0+\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varepsilon}_{\mathcal{X}})$. Let $Z \in \mathbb{N}^\mathbb{N}$; we want to show that there exists Y with $Y = \mathcal{J}^\omega(Z)$. Build $\mathcal{X} = \langle \mathcal{JT}^\omega(T_Z), \leq_{\mathrm{KB}}\rangle$ as in Theorem 5.21. The proof that $\boldsymbol{\varepsilon}_{\mathcal{X}}$ has a Z-computable descending sequence is completely finitary and can be carried out in RCA_0. By $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varepsilon}_{\mathcal{X}})$, we get that $\mathcal{X}$ has a descending sequence. Since we have ACA_0 we can use this descending sequence to get a path Y through $\mathcal{JT}^\omega(T_Z)$. Now, the proof of Lemma 5.14 translates into a proof in RCA_0 that Y is $\mathcal{J}^\omega$ of some path through T_Z. Since Z is the only path through T_Z, we get $Y = \mathcal{J}^\omega(Z)$ as wanted. □

6. General Case

In this section we define the ω^α-Jump operator, the ω^α-Jump function, and the ω^α-Jump Tree, for all computable ordinals α. The constructions of Sections 4 and 5, where we considered $\alpha = 0$ and $\alpha = 1$ respectively, are thus the simplest cases of what we will be doing here.

The whole construction is by transfinite recursion, and the base case was covered in Section 4. If $\alpha > 0$ is a computable ordinal, we assume that we have a fixed non-decreasing computable sequence of ordinals $\{\, \alpha_i : i \in \mathbb{N} \,\}$ such that $\alpha = \sup_{i \in \mathbb{N}}(\alpha_i + 1)$. (So, if $\alpha = \gamma + 1$, we can take $\alpha_i = \gamma$ for all i.) Notice that we have $\sum_{i\in\mathbb{N}} \omega^{\alpha_i} = \omega^\alpha$. In defining the ω^α-Jump operator, the ω^α-Jump function, and the ω^α-Jump Tree we make use the ω^{α_i}-Jump operator, the ω^{α_i}-Jump function, and the ω^{α_i}-Jump Tree for each i.

6.1. The iteration of the jump.

Our presentation here is different from the one of previous sections, where we defined the operator first. Here we start from the ω^α-Jump function, prove its basic properties, then use it to define the ω^α-Jump Tree, and eventually introduce the ω^α-Jump operator.

Let $\alpha > 0$ be a computable ordinal and $\{\, \alpha_i : i \in \mathbb{N} \,\}$ be its canonical sequence as described above. To simplify the notation in the definition of the ω^α-Jump function, assume we already defined $J^{\omega^{\alpha_i}}$ and $K^{\omega^{\alpha_i}}$ for all i, and let $J_n^{\omega^\alpha} : \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$ and $K_n^{\omega^\alpha} : \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$ be defined recursively by

$$J_0^{\omega^\alpha} = id; \qquad J_{n+1}^{\omega^\alpha} = J^{\omega^{\alpha_n}} \circ J_n^{\omega^\alpha};$$
$$K_0^{\omega^\alpha} = id; \qquad K_{n+1}^{\omega^\alpha} = K_n^{\omega^\alpha} \circ K^{\omega^{\alpha_n}}.$$

In other words:

$$\begin{aligned} J^{\omega^\alpha}_n &= J^{\omega^{\alpha_{n-1}}} \circ J^{\omega^{\alpha_{n-2}}} \circ \dots \circ J^{\omega^{\alpha_0}}, \\ K^{\omega^\alpha}_n &= K^{\omega^{\alpha_0}} \circ K^{\omega^{\alpha_1}} \circ \dots \circ K^{\omega^{\alpha_{n-1}}}. \end{aligned}$$

Definition 6.1. The ω^α-*Jump function* is the map $J^{\omega^\alpha} : \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$ defined by

$$J^{\omega^\alpha}(\sigma) = \langle J^{\omega^\alpha}_1(\sigma)(0), J^{\omega^\alpha}_2(\sigma)(0), \dots, J^{\omega^\alpha}_{n-1}(\sigma)(0)\rangle,$$

where n is least such that $J^{\omega^\alpha}_n(\sigma) = \emptyset$. In this case, since $J^{\omega^\alpha}_n(\sigma) = J^{\omega^{\alpha_{n-1}}}(J^{\omega^\alpha}_{n-1}(\sigma))$, by ($\mathrm{P}^{\omega^\alpha}1$) below applied to α_{n-1}, we have $|J^{\omega^\alpha}_{n-1}(\sigma)| = 1$.

Given $\tau \in J^{\omega^\alpha}(\mathbb{N}^{<\mathbb{N}})$, let

$$K^{\omega^\alpha}(\tau) = K^{\omega^\alpha}_{|\tau|}(\ell(\tau)).$$

In particular $K^{\omega^\alpha}(\emptyset) = \emptyset$, since $K^{\omega^\alpha}_0$ is the identity function.

Since for $\alpha = 1$ we have $\alpha_i = 0$ for every i, the definitions we just gave match exactly Definitions 5.8 and 5.9, where we introduced J^ω and K^ω. We will not mention again this explicitly, but the reader should keep in mind that the case $\alpha = 1$ of Section 5 is the blueprint for the work of this section.

Notice that, by transfinite induction, J^{ω^α} and K^{ω^α} are computable.

The following properties generalize those of Lemmas 4.4 and 5.10. We will refer to them, as usual, as ($\mathrm{P}^{\omega^\alpha}1$), ..., ($\mathrm{P}^{\omega^\alpha}7$).

Lemma 6.2. *For $\sigma, \tau' \in \mathbb{N}^{<\mathbb{N}}$, $\tau \in J^{\omega^\alpha}(\mathbb{N}^{<\mathbb{N}})$,*

($\mathrm{P}^{\omega^\alpha}1$) $J^{\omega^\alpha}(\sigma) = \emptyset$ *if and only if* $|\sigma| \leq 1$.
($\mathrm{P}^{\omega^\alpha}2$) $K^{\omega^\alpha}(J^{\omega^\alpha}(\sigma)) = \sigma$ *for* $|\sigma| \geq 2$.
($\mathrm{P}^{\omega^\alpha}3$) $J^{\omega^\alpha}(K^{\omega^\alpha}(\tau)) = \tau$.
($\mathrm{P}^{\omega^\alpha}4$) *If* $\sigma \neq \sigma'$ *and at least one has length* ≥ 2, *then* $J^{\omega^\alpha}(\sigma) \neq J^{\omega^\alpha}(\sigma')$.
($\mathrm{P}^{\omega^\alpha}5$) $|J^{\omega^\alpha}(\sigma)| < |\sigma|$ *and* $|K^{\omega^\alpha}(\tau)| > |\tau|$ *except when* $\tau = \emptyset$.
($\mathrm{P}^{\omega^\alpha}6$) *If* $\tau' \subset \tau$ *then* $\tau' \in J^{\omega^\alpha}(\mathbb{N}^{<\mathbb{N}})$ *and* $K^{\omega^\alpha}(\tau') \subset K^{\omega^\alpha}(\tau)$.
($\mathrm{P}^{\omega^\alpha}7$) *If* $J^{\omega^\alpha}(\sigma') \subseteq J^{\omega^\alpha}(\sigma)$ *and* $\alpha > 0$ *then for every* m, $J^{\omega^\alpha}_m(\sigma') \subseteq J^{\omega^\alpha}_m(\sigma)$.

Proof. The proof is by transfinite induction on α. The case $\alpha = 0$ is Lemma 4.4.

Since $J^{\omega^\alpha}(\sigma) = \emptyset$ if and only if $J^{\omega^{\alpha_0}}(\sigma) = \emptyset$, ($\mathrm{P}^{\omega^\alpha}1$) follows from the same property for α_0.

To prove ($\mathrm{P}^{\omega^\alpha}2$) let $|J^{\omega^\alpha}(\sigma)| = n - 1 > 0$. Then $\ell(J^{\omega^\alpha}(\sigma)) = \langle J^{\omega^\alpha}_{n-1}(\sigma)(0)\rangle = J^{\omega^\alpha}_{n-1}(\sigma)$ because $|J^{\omega^\alpha}_{n-1}(\sigma)| = 1$ as noticed above. Since $K^{\omega^\alpha}(J^{\omega^\alpha}(\sigma)) = K^{\omega^\alpha}_{n-1}(J^{\omega^\alpha}_{n-1}(\sigma))$, $K^{\omega^\alpha}(J^{\omega^\alpha}(\sigma)) = \sigma$ follows from ($\mathrm{P}^{\omega^\alpha}2$) for $\alpha_{n-2}, \alpha_{n-3}, \dots, \alpha_0$.

As in the proof of the case $\alpha = 1$ in Lemma 5.10, ($\mathrm{P}^{\omega^\alpha}3$), ($\mathrm{P}^{\omega^\alpha}4$) and ($\mathrm{P}^{\omega^\alpha}5$) follow from the properties we already proved.

The proof of ($\mathrm{P}^{\omega^\alpha}6$) is also basically the same as the proof of ($\mathrm{P}^{\omega}6$). We recommend the reader to have Figure 3 in mind while reading the proof. The nontrivial case is when $\tau' \neq \emptyset$. Let σ be such that $\tau = J^{\omega^\alpha}(\sigma)$. The idea is to define $\sigma' \subset \sigma$ as in the picture and then show that $\tau' = J^{\omega^\alpha}(\sigma')$. Notice that $|\sigma| > |\tau| \geq 2$ by ($\mathrm{P}^{\omega^\alpha}5$), and that $\sigma = K^{\omega^\alpha}(\tau)$ by ($\mathrm{P}^{\omega^\alpha}2$). Notice also that $\ell(\tau') = \langle \tau(|\tau'| - 1)\rangle = \langle J^{\omega^\alpha}_{|\tau'|}(\sigma)(0)\rangle \subset J^{\omega^\alpha}_{|\tau'|}(\sigma)$, where the strict inclusion is because $|\tau'| < |\tau|$ and hence $|J^{\omega^\alpha}_{|\tau'|}(\sigma)| > 1$. By induction on $i \leq |\tau'|$ we can show, using ($\mathrm{P}^{\omega^\alpha}6$) and ($\mathrm{P}^{\omega^\alpha}2$) for $\alpha_{|\tau'|-1}, \dots, \alpha_1, \alpha_0$, that

$$\begin{aligned} (K^{\omega^{\alpha_{|\tau'|-i}}} \circ \dots \circ K^{\omega^{\alpha_{|\tau'|-1}}})(\ell(\tau')) &\subset (K^{\omega^{\alpha_{|\tau'|-i}}} \circ \dots \circ K^{\omega^{\alpha_{|\tau'|-1}}})(J^{\omega^\alpha}_{|\tau'|}(\sigma)) \\ &= J^{\omega^\alpha}_{|\tau'|-i}(\sigma) \end{aligned}$$

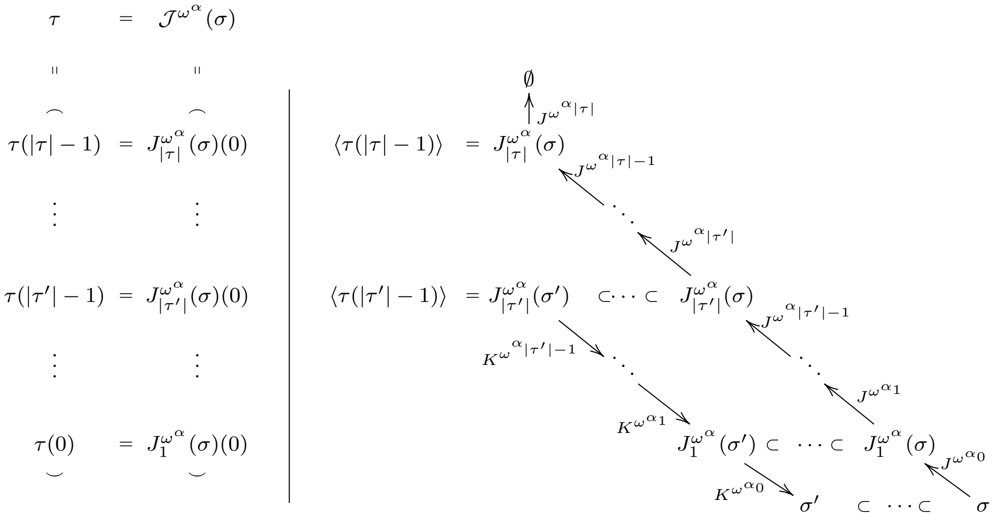

$\tau = \mathcal{J}^{\omega^\alpha}(\sigma)$
$\tau(|\tau|-1) = J^{\omega^\alpha}_{|\tau|}(\sigma)(0)$
$\tau(|\tau'|-1) = J^{\omega^\alpha}_{|\tau'|}(\sigma)(0)$
$\tau(0) = J^{\omega^\alpha}_{1}(\sigma)(0)$
$\emptyset$
$J^{\omega^{\alpha_{|\tau|}}}$
$\langle\tau(|\tau|-1)\rangle = J^{\omega^\alpha}_{|\tau|}(\sigma)$
$J^{\omega^{\alpha_{|\tau|-1}}}$
$J^{\omega^{\alpha_{|\tau'|}}}$
$\langle\tau(|\tau'|-1)\rangle = J^{\omega^\alpha}_{|\tau'|}(\sigma') \subset \cdots \subset J^{\omega^\alpha}_{|\tau'|}(\sigma)$
$J^{\omega^{\alpha_{|\tau'|-1}}}$
$K^{\omega^{\alpha_{|\tau'|-1}}}$
$J^{\omega^{\alpha_1}}$
$K^{\omega^{\alpha_1}}$
$J^{\omega^\alpha}_{1}(\sigma') \subset \cdots \subset J^{\omega^\alpha}_{1}(\sigma)$
$J^{\omega^{\alpha_0}}$
$K^{\omega^{\alpha_0}}$
$\sigma' \subset \cdots \subset \sigma$

FIGURE 3. Assuming $\tau = J^{\omega^\alpha}(\sigma)$ and $\tau' \subset \tau$.

and $(K^{\omega^{\alpha_{|\tau'|-i}}} \circ \cdots \circ K^{\omega^{\alpha_{|\tau'|-1}}})(\ell(\tau')) \in J^{\omega^\alpha}_{|\tau'|-i}(\mathbb{N}^{<\mathbb{N}})$. In particular, when $i = |\tau'|$, if we set $\sigma' = K^{\omega^\alpha}_{|\tau'|}(\ell(\tau'))$, we obtain $\sigma' \subset \sigma$. Furthermore, by ($\mathsf{P}^{\omega^\alpha}2$) applied to $\alpha_0, \ldots, \alpha_{|\tau'|-i-1}$, we also get

$$J^{\omega^\alpha}_{|\tau'|-i}(\sigma') = (K^{\omega^{\alpha_{|\tau'|-i}}} \circ \cdots \circ K^{\omega^{\alpha_{|\tau'|-1}}})(\ell(\tau')) \subset J^{\omega^\alpha}_{|\tau'|-i}(\sigma). \tag{6.1}$$

Therefore, for every $j < |\tau'|$

$$J^{\omega^\alpha}(\sigma')(j) = J^{\omega^\alpha}_{j+1}(\sigma')(0) = J^{\omega^\alpha}_{j+1}(\sigma)(0) = \tau(j) = \tau'(j).$$

Since $J^{\omega^\alpha}_{|\tau'|-1}(\sigma') = \ell(\tau')$ which has length 1, we get that $J^{\omega^\alpha}(\sigma')$ has length $|\tau'|$ as wanted.

For ($\mathsf{P}^{\omega^\alpha}7$) let $\tau' = J^{\omega^\alpha}(\sigma')$. Then, if $i = |\tau'| - m$, equation 6.1 shows that $J^{\omega^\alpha}_m(\sigma') \subseteq J^{\omega^\alpha}_m(\sigma)$. □

We can now introduce the ω^α-Jump Tree and prove its computability.

Definition 6.3. Given a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$ the ω^α-*Jump Tree of* T is

$$\mathcal{JT}^{\omega^\alpha}(T) = \{\, J^{\omega^\alpha}(\sigma) : \sigma \in T \,\}.$$

Lemma 6.4. *For every tree T, $\mathcal{JT}^{\omega^\alpha}(T)$ is a tree computable in T.*

Proof. The proof is again the same as the one of Lemma 4.8, using Lemma 6.2 in place of Lemma 4.4. □

We now define the ω^α-Jump operator $\mathcal{J}^{\omega^\alpha} : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ by transfinite induction: the base case is the Jump operator $\mathcal{J}$ (Definition 4.1). Given α we assume that $\mathcal{J}^{\omega^{\alpha_n}}$ has been defined for all n. To simplify the notation let us define $\mathcal{J}^{\omega^\alpha}_n$ recursively by $\mathcal{J}^{\omega^\alpha}_0 = id$, $\mathcal{J}^{\omega^\alpha}_{n+1} = \mathcal{J}^{\omega^{\alpha_n}} \circ \mathcal{J}^{\omega^\alpha}_n$, so that

$$\mathcal{J}^{\omega^\alpha}_n = \mathcal{J}^{\omega^{\alpha_{n-1}}} \circ \mathcal{J}^{\omega^{\alpha_{n-2}}} \circ \cdots \circ \mathcal{J}^{\omega^{\alpha_0}}.$$

Definition 6.5. Given the computable ordinal α we define the ω^α-*Jump operator* $\mathcal{J}^{\omega^\alpha} : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ and its inverse $\mathcal{K}^{\omega^\alpha}$ by

$$\mathcal{J}^{\omega^\alpha}(Z)(n) = \mathcal{J}^{\omega^\alpha}_{n+1}(Z)(0) \quad \text{and} \quad \mathcal{K}^{\omega^\alpha}(Y) = \bigcup_n K^{\omega^\alpha}(Y \restriction n).$$

We first show that $\mathcal{K}^{\omega^\alpha}$ is indeed the inverse of $\mathcal{J}^{\omega^\alpha}$.

Lemma 6.6. *If $Y = \mathcal{J}^{\omega^\alpha}(Z)$ then $Z = \mathcal{K}^{\omega^\alpha}(Y)$.*

Proof. The proof of the lemma is by transfinite induction. Let $\{\, \alpha_i : i \in \mathbb{N} \,\}$ be the fixed canonical sequence fo α. Recall from the definition of K^{ω^α} that $K^{\omega^\alpha}(Y\restriction n) = K_n^{\omega^\alpha}(\langle Y(n-1)\rangle)$. Since $\langle Y(n-1)\rangle = \langle \mathcal{J}_n^{\omega^\alpha}(Z)(0)\rangle \subseteq \mathcal{J}_n^{\omega^\alpha}(Z)$, by the induction hypothesis applied to $\alpha_{n-1}, \dots, \alpha_0$, we get that $K_n^{\omega^\alpha}(\langle Y(n-1)\rangle) \subseteq Z$. So $Z \supseteq \mathcal{K}^{\omega^\alpha}(Y)$. By (P$^{\omega^\alpha}$5) applied to $\alpha_0, \dots, \alpha_{n-1}$ we get that $|K_n^{\omega^\alpha}(\langle Y(n-1)\rangle)| > n+1$ and hence $Z = \mathcal{K}^{\omega^\alpha}(Y)$. □

Lemma 6.7. *For every $Z \in \mathbb{N}^{\mathbb{N}}$ and computable ordinal α, $\mathcal{J}^{\omega^\alpha}(Z) \equiv_T Z^{(\omega^\alpha)}$.*

Proof. This is again proved by transfinite induction. Assuming that $\mathcal{J}^{\omega^{\alpha_i}}(Z) \equiv_T Z^{(\omega^{\alpha_i})}$ for every i, and uniformly in i, we immediately obtain $\mathcal{J}_n^{\omega^\alpha}(Z) \equiv_T Z^{(\beta_n)}$, where $\beta_n = \sum_{i=0}^{n-1} \omega^{\alpha_i}$, for every n. Since $\beta_n < \omega^\alpha$, $\mathcal{J}^{\omega^\alpha}(Z) \leq_T Z^{(\omega^\alpha)}$ is immediate.

For the other reduction we need to uniformly compute $\mathcal{J}_n^{\omega^\alpha}(Z)$ from $\mathcal{J}^{\omega^\alpha}(Z)$. The same way we compute Z from $\mathcal{J}^{\omega^\alpha}(Z)$ applying $\mathcal{K}^{\omega^\alpha}$, we can compute $\mathcal{J}_n^{\omega^\alpha}(Z)$ by forgetting about $\alpha_0, \dots, \alpha_{n-1}$. In other words, by the same proof as Lemma 6.6 we can show that for every m

$$\mathcal{J}_m^{\omega^\alpha}(Z) = \bigcup_{n>m} K^{\omega^{\alpha_m}}(K^{\omega^{\alpha_{m+1}}}(\dots(K^{\omega^{\alpha_{n-1}}}(\langle Y(n-1)\rangle))\dots))$$

using $K^{\omega^{\alpha_m}} \circ K^{\omega^{\alpha_{m+1}}} \circ \dots \circ K^{\omega^{\alpha_{n-1}}}$ instead of $K_n^{\omega^\alpha}$. □

We can now prove that J^{ω^α} approximates $\mathcal{J}^{\omega^\alpha}$, extending Lemma 5.11.

Lemma 6.8. *Given $Y, Z \in \mathbb{N}^{\mathbb{N}}$, the following are equivalent:*

1. *$Y = \mathcal{J}^{\omega^\alpha}(Z)$;*
2. *for every n there exists $\sigma_n \subset Z$ with $|\sigma_n| > n$ such that $Y\restriction n = J^{\omega^\alpha}(\sigma_n)$.*

Proof. We first prove (1) $\implies$ (2). When $n = 0$ let $\sigma_0 = Z\restriction 1$, which works by (P$^{\omega^\alpha}$1). Let $\sigma_n = K^{\omega^\alpha}(Y\restriction n)$. Then $\sigma_n \subseteq \mathcal{K}^{\omega^\alpha}(Y) = Z$, and $Y\restriction n = J^{\omega^\alpha}(\sigma_n)$. We get that $|\sigma_n| > n$ by applying (P$^{\omega^\alpha}$5) n times to $\sigma_n = K_n^{\omega^\alpha}(\langle Y(n-1)\rangle)$.

The proof of (2) $\implies$ (1) is similar to the proof of Lemma 5.11 but uses transfinite induction. By (P$^{\omega^\alpha}$7), for all m and n, $J_m^{\omega^\alpha}(\sigma_n) \subseteq J_m^{\omega^\alpha}(\sigma_{n+1})$, and hence we can consider $\bigcup_n J_m^{\omega^\alpha}(\sigma_n) \in \mathbb{N}^{\mathbb{N}}$. Then, by the induction hypothesis, $\bigcup_n J_m^{\omega^\alpha}(\sigma_n) = \mathcal{J}_m^{\omega^\alpha}(Z)$, and hence $J_m^{\omega^\alpha}(\sigma_n)(0) = \mathcal{J}_m^{\omega^\alpha}(Z)(0)$ for all $m < n$. It follows that for every m and $n > m$

$$\mathcal{J}^{\omega^\alpha}(Z)(m) = \mathcal{J}_{m+1}^{\omega^\alpha}(Z)(0) = J_{m+1}^{\omega^\alpha}(\sigma_n)(0) = J^{\omega^\alpha}(\sigma_n)(m) = Y(m). \qquad \square$$

We are now able to show the intended connection between the ω^α-Jump Tree and the ω^α-Jump operator.

Lemma 6.9. *For every tree T, $[\mathcal{JT}^{\omega^\alpha}(T)] = \{\, \mathcal{J}^{\omega^\alpha}(Z) : Z \in [T] \,\}$.*

Proof. To prove $\{\, \mathcal{J}^{\omega^\alpha}(Z) : Z \in [T] \,\} \subseteq [\mathcal{JT}^{\omega^\alpha}(T)]$ we can argue as in the proof of Lemma 4.9, using Lemma 6.8 in place of Lemma 4.5.

To prove the other inclusion, fix $Y \in [\mathcal{JT}^{\omega^\alpha}(T)]$. Arguing as in the proof of Lemma 5.14, we first let $\sigma_n = K^{\omega^\alpha}(Y\restriction n) \in T$. Let $Z = \mathcal{K}^{\omega^\alpha}(Y) = \bigcup_{n\in\mathbb{N}} \sigma_n \in [T]$. We get that $Y = \mathcal{J}^{\omega^\alpha}(Z)$ from Lemma 6.8. □

6.2. **Jumps versus Veblen.** First, we need to iterate the Jump Tree operator along a finite string.

Definition 6.10. If T is a tree and $\tau \in \mathcal{JT}^{\omega^\alpha}(T)$ we define

$$\mathcal{JT}^{\omega^\alpha}_\tau(T) = \{\, J^{\omega^\alpha}_{|\tau|+1}(\sigma) : \sigma \in T \wedge \tau \subsetneq J^{\omega^\alpha}(\sigma) \,\}.$$

Lemma 6.11. *For $\tau \in \mathcal{JT}^{\omega^\alpha}(T)$,*

$$\mathcal{JT}^{\omega^\alpha}_\emptyset(T) = \mathcal{JT}^{\omega^{\alpha_0}}(T)$$
$$\mathcal{JT}^{\omega^\alpha}_{\tau^\frown\langle c\rangle}(T) = \mathcal{JT}^{\omega^{\alpha_{|\tau|+1}}}(\mathcal{JT}^{\omega^\alpha}_\tau(T)_{\langle c\rangle}).$$

($T_{\langle c\rangle}$ was defined in 2.10.)

Proof. Straightforward induction on $|\tau|$. □

Lemma 6.12. *Given a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$, $\tau \in \mathbb{N}^{<\mathbb{N}}$, and $c \in \mathbb{N}$*

$$\tau^\frown\langle c\rangle \in \mathcal{JT}^{\omega^\alpha}(T) \iff \langle c\rangle \in \mathcal{JT}^{\omega^\alpha}_\tau(T).$$

Proof. Follows from the definitions of $\mathcal{JT}^{\omega^\alpha}(T)$ and $\mathcal{JT}^{\omega^\alpha}_\tau(T)$. □

We now generalize the construction of Definition 5.18, by defining an operator that converts a function with domain $\mathcal{JT}^{\omega^\alpha}(T)$ and values in $\mathcal{X}$ into a function with domain T and values in $\boldsymbol{\varphi}(\alpha, \mathcal{X})$. We will show in Lemma 6.14 that this operator preserves monotonicity.

Definition 6.13. By transfinite recursion, we build, for each computable ordinal α, an operator h^{ω^α} such that given a linear ordering $\mathcal{X}$ and a function

$$g\colon \mathcal{JT}^{\omega^\alpha}(T) \to \mathcal{X},$$

it returns

$$h^{\omega^\alpha}_g : T \to \boldsymbol{\varphi}(\alpha, \mathcal{X}).$$

For $\alpha = 0$, we let $h^{\omega^\alpha} = h$ of Definition 5.1. For $\alpha > 0$ we first define simultaneously for each $\tau \in \mathcal{JT}^{\omega^\alpha}(T)$ a function

$$f_\tau\colon \mathcal{JT}^{\omega^\alpha}_\tau(T) \to \boldsymbol{\varphi}(\alpha, \mathcal{X})$$

by recursion on $|\sigma|$:

$$f_\tau(\sigma) = \begin{cases} \varphi_{\alpha, g(\tau)} & \text{if } \sigma = \emptyset; \\ \\ h^{\omega^{\alpha_n}}_{f_{\tau'}}(\sigma) & \text{if } \sigma \neq \emptyset, \text{ where } \tau' = \tau^\frown\langle\sigma(0)\rangle \text{ and } n = |\tau'|. \end{cases}$$

We then define

$$h^{\omega^\alpha}_g = h^{\omega^{\alpha_0}}_{f_\emptyset} : T \to \boldsymbol{\varphi}(\alpha, \mathcal{X}).$$

Lemma 6.14. *If $g\colon \mathcal{JT}^{\omega^\alpha}(T) \to \mathcal{X}$ is total and $(\supset, <_\mathcal{X})$-monotone, then $h^{\omega^\alpha}_g : T \to \boldsymbol{\varphi}(\alpha, \mathcal{X})$ is also total and $(\supset, <_{\boldsymbol{\varphi}(\alpha,\mathcal{X})})$-monotone. Moreover, $h^{\omega^\alpha}_g$ is computable in g.*

Proof. We say that a partial function e on a tree T is $(n, T, \mathcal{X})$-*good* if e is defined on all strings of length less than or equal to n, it takes values in $\mathcal{X}$, and is $(\supset, <_\mathcal{X})$-monotone on strings of length less than or equal to n.

By transfinite induction on α we will show that for every $n \in \mathbb{N}$ and every $(n, \mathcal{JT}^{\omega^\alpha}(T), \mathcal{X})$-good partial function g, we have that $h^{\omega^\alpha}_g$ is $(n+1, T, \boldsymbol{\varphi}(\alpha, \mathcal{X}))$-good.

For $\alpha = 0$, this follows from the proof of Lemma 5.2: recall that $h_g(\sigma)$ is a finite sum of terms of the form $\omega^{g(J(\sigma)\upharpoonright i)}$, and $|J(\sigma)| < |\sigma|$ by (P5). Thus to compute and compare h_g on strings of length $\leq n+1$, we need only g to be defined and $(\supset, <_{\mathcal{X}})$-monotone on strings of length $\leq n$.

Now fix $\alpha > 0$ and suppose that g is $(n, \mathcal{JT}^{\omega^\alpha}(T), \mathcal{X})$-good. Since $h_g^{\omega^\alpha} = h_{f_\emptyset}^{\omega^{\alpha_0}}$, by the induction hypothesis it is enough to show that $f_\emptyset$ is $(n, \mathcal{JT}_\emptyset^{\omega^\alpha}(T), \boldsymbol{\varphi}(\alpha, \mathcal{X}))$-good. Notice that if $f_\emptyset$ takes values in $\boldsymbol{\varphi}(\alpha, \mathcal{X})$, then $h_g^{\omega^\alpha}$ takes values in $\boldsymbol{\varphi}(\alpha_0, \boldsymbol{\varphi}(\alpha, \mathcal{X})) = \boldsymbol{\varphi}(\alpha, \mathcal{X})$ (by Definition 2.7). We will prove by induction on $m \leq n$ that for every $\tau \in \mathcal{JT}^{\omega^\alpha}(T)$ of length $n-m$, f_τ is $(m, \mathcal{JT}_\tau^{\omega^\alpha}(T), \boldsymbol{\varphi}(\alpha, \mathcal{X}))$-good. When $m = 0$, all we need to observe is that $f_\tau(\emptyset) = \varphi_{\alpha, g(\tau)} \in \boldsymbol{\varphi}(\alpha, \mathcal{X})$, and $g(\tau)$ is defined because $|\tau| = n$. Consider now $\tau \in \mathcal{JT}^{\omega^\alpha}(T)$ of length $n-(m+1)$. If $\sigma = \emptyset$, then $f_\tau(\emptyset)$ is correctly defined as in the case $m = 0$. For $\sigma \in \mathcal{JT}_\tau^{\omega^\alpha}(T)$ with $0 < |\sigma| \leq m+1$, let $\tau' = \tau^\frown\langle\sigma(0)\rangle$. We first need to check that $f_\tau(\sigma) = h_{f_{\tau'}}^{\omega^{\alpha_{n-m}}}(\sigma)$ is defined. By the subsidiary induction hypothesis $f_{\tau'}$ is $(m, \mathcal{JT}_{\tau'}^{\omega^\alpha}(T), \boldsymbol{\varphi}(\alpha, \mathcal{X}))$-good. By Lemma 6.11, $\mathcal{JT}_{\tau'}^{\omega^\alpha}(T) = \mathcal{JT}^{\omega^{\alpha_{n-m}}}(\mathcal{JT}_\tau^{\omega^\alpha}(T)_{\langle\sigma(0)\rangle})$. By the transfinite induction hypothesis (since $\alpha_{n-m} < \alpha$) $h_{f_{\tau'}}^{\omega^{\alpha_{n-m}}}$ is $(m+1, \mathcal{JT}_\tau^{\omega^\alpha}(T)_{\langle\sigma(0)\rangle}, \boldsymbol{\varphi}(\alpha, \mathcal{X}))$-good. Therefore $f_\tau(\sigma)$ is defined. Now we need to show f_τ is $(\supset, <_{\boldsymbol{\varphi}(\alpha,\mathcal{X})})$-monotone on strings of length less than or equal to $m+1$. Take $\sigma' \subset \sigma \in \mathcal{JT}_\tau^{\omega^\alpha}(T)$ with $|\sigma| \leq m+1$. Again let $\tau' = \tau^\frown\langle\sigma(0)\rangle$. By the transfinite induction hypothesis, we know that $h_{f_{\tau'}}^{\omega^{\alpha_{n-m}}}$ is $(m+1, \mathcal{JT}_\tau^{\omega^\alpha}(T), \boldsymbol{\varphi}(\alpha, \mathcal{X}))$-good. Furthermore $f_{\tau'}$ is $(\supset, <_{\boldsymbol{\varphi}(\alpha,\mathcal{X})})$-monotone and takes values in $\boldsymbol{\varphi}(\alpha, \mathcal{X}) \upharpoonright (\varphi_{\alpha, g(\tau')} + 1)$, because $f_{\tau'}(\emptyset) = \varphi_{\alpha, g(\tau')}$. Therefore $h_{f_{\tau'}}^{\omega^{\alpha_{n-m}}}$ takes values below $\varphi_{\alpha_{n-m}}(\varphi_{\alpha, g(\tau')} + 1)$. When $\sigma' = \emptyset$, $f_\tau(\sigma') = \varphi_{\alpha, g(\tau)} > \varphi_{\alpha_{n-m}}(\varphi_{\alpha, g(\tau')} + 1) > f_\tau(\sigma)$. When $\sigma' \neq \emptyset$, we use the monotonicity of $h_{f_{\tau'}}^{\omega^{\alpha_{n-m}}}$. □

Theorem 6.15. *For every computable ordinal α and $Z \in \mathbb{N}^\mathbb{N}$, there exists a Z-computable linear ordering $\mathcal{X}$ such that the jump of every descending sequence in $\mathcal{X}$ computes $Z^{(\omega^\alpha)}$, but there is a Z-computable descending sequence in $\boldsymbol{\varphi}(\alpha, \mathcal{X})$.*

Proof. Let $\mathcal{X} = \langle \mathcal{JT}^{\omega^\alpha}(T_Z), \leq_{\mathrm{KB}} \rangle$ where T_Z is the tree $\{\, Z \upharpoonright n : n \in \mathbb{N} \,\}$. By Lemma 6.4, $\mathcal{X}$ is Z-computable. By Lemma 6.9, $\mathcal{J}^{\omega^\alpha}(Z)$ is the unique path in $\mathcal{JT}^{\omega^\alpha}(T_Z)$. Therefore, by Lemma 2.12, the jump of every descending sequence in $\mathcal{X}$ computes $\mathcal{J}^{\omega^\alpha}(Z)$ and hence, by Lemma 6.7, computes $Z^{(\omega^\alpha)}$.

Let g be the identity on $\mathcal{X}$, which is $(\supset, <_{\mathcal{X}})$-monotone. By Lemma 6.14, h_g^α is $(\supset, <_{\boldsymbol{\varphi}(\alpha,\mathcal{X})})$-monotone and computable. Thus $\{\, h_g^\alpha(Z \upharpoonright n) : n \in \mathbb{N} \,\}$ is a Z-computable descending sequence in $\boldsymbol{\varphi}(\alpha, \mathcal{X})$. □

6.3. **Reverse mathematics results.** In this section, we work in the weak system RCA_0. Therefore, again, we do not have an operation that given $Z \in \mathbb{N}^\mathbb{N}$, returns $\mathcal{J}^{\omega^\alpha}(Z)$ but the predicate with three variables Z, Y and α that says $Y = \mathcal{J}^{\omega^\alpha}(Z)$ is arithmetic as witnessed by Lemma 6.8. Notice that if if we have that condition (2) of Lemma 6.8 holds, then RCA_0 can recover all the $\mathcal{J}_m^{\omega^\alpha}(Z)$ and show that $\mathcal{J}^{\omega^\alpha}(Z)$ is as defined in Definition 6.5. We can then prove Lemma 6.7 in RCA_0: if $Y = \mathcal{J}^{\omega^\alpha}(Z)$, then Y can compute $Z^{(\omega^\alpha)}$, and $Z^{(\omega^\alpha)}$ can compute a real Y such that $Y = \mathcal{J}^{\omega^\alpha}(Z)$. Therefore, we get that $\Pi^0_{\omega^\alpha}$-CA_0 is equivalent to $\mathsf{RCA}_0 + \forall Z \exists Y (Y = \mathcal{J}^{\omega^\alpha}(Z))$ and that ATR_0 is equivalent to $\mathsf{RCA}_0 + \forall \alpha \forall Z \exists Y (Y = \mathcal{J}^{\omega^\alpha}(Z))$.

Theorem 6.16. *Let α be a computable ordinal. Over RCA_0, $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varphi}(\alpha, \mathcal{X}))$ is equivalent to $\Pi^0_{\omega^\alpha}$-CA_0.*

Proof. We already showed that $\Pi^0_{\omega^\alpha}$-$\mathsf{CA}_0 \vdash \mathsf{WOP}(\mathcal{X} \mapsto \varphi(\alpha, \mathcal{X}))$ in Corollary 3.7.

The proof of the other direction is just the formalization of Theorem 6.15 exactly as we did in Theorem 5.23. □

We now give a new, purely computability-theoretic, proof of Friedman's theorem.

Theorem 6.17. *Over* RCA_0*,* $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varphi}(\mathcal{X}, 0))$ *is equivalent to* ATR_0*.*

Proof. We already showed that ATR_0 proves $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varphi}(\mathcal{X}, 0))$ in Corollary 3.8.

For the reversal, we argue within RCA_0. Let α be any ordinal. Notice that relative to the presentation of α, all the constructions of this section can be done as if α were any computable ordinal. Therefore, by the previous theorem it is enough to show that $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varphi}(\alpha, \mathcal{X}))$ holds. Let $\mathcal{X}$ be a well-ordering. We now claim that $\boldsymbol{\varphi}(\alpha, \mathcal{X})$ embeds in $\boldsymbol{\varphi}(\alpha + \mathcal{X}, 0)$, which would imply that $\boldsymbol{\varphi}(\alpha, \mathcal{X})$ is well-ordered too as needed to show $\mathsf{WOP}(\mathcal{X} \mapsto \boldsymbol{\varphi}(\alpha, \mathcal{X}))$.

Define $f\colon \boldsymbol{\varphi}(\alpha, \mathcal{X}) \to \boldsymbol{\varphi}(\alpha + \mathcal{X}, 0)$ by induction on the terms of $\boldsymbol{\varphi}(\alpha, \mathcal{X})$, setting

- $f(0) = 0$,
- $f(\varphi_{\alpha,x}) = \varphi_{\alpha+x}(0)$,
- $f(t_1 + t_2) = f(t_1) + f(t_2)$,
- $f(\varphi_a(t)) = \varphi_a(f(t))$.

The proof that f is an embedding is by induction on terms. Consider $t, s \in \boldsymbol{\varphi}(\alpha, \mathcal{X})$. We want to show that $t \leq_{\boldsymbol{\varphi}(\alpha,\mathcal{X})} s \iff f(t) \leq_{\boldsymbol{\varphi}(\alpha+\mathcal{X},0)} f(s)$. By induction hypothesis, assume this is true for pairs of terms shorter than $t + s$. Suppose that $t \leq s$. Using the induction hypothesis, it is not hard to show that $f(t) \leq f(s)$. Suppose now that $t \not\leq s$. Then, none of the conditions of Definition 2.7 hold. If $t = 0, t = t_1 + t_2$, or $t = \varphi_a(t_1)$, then we can apply the induction hypothesis again and get that none of the conditions of Definition 2.7 hold for $f(t)$ and $f(s)$ either and hence $f(t) \not\leq f(s)$. The case $t = \varphi_{\alpha,x}$ is the only one that deserves attention. In this case we have that for no $y \geq x$, $\varphi_{\alpha,y}$ appears in s. It then follows that $f(s) \in \boldsymbol{\varphi}(\alpha + \mathcal{X}\restriction x, 0)$. Since $f(t) = \varphi_{\alpha+x}(0)$ is greater than all the elements of $\boldsymbol{\varphi}(\alpha + \mathcal{X}\restriction x, 0)$ we get $f(t) \not\leq f(s)$ a wanted. □

References

[Afs08] Bahareh Afshari. *Relative computability and the proof-theoretic strength of some theories.* PhD thesis, University of Leeds, 2008.

[AR09] Bahareh Afshari and Michael Rathjen. Reverse mathematics and well-ordering principles: A pilot study. *Ann. Pure Appl. Logic*, 160(3):231–237, 2009.

[BHS87] Andreas R. Blass, Jeffry L. Hirst, and Stephen G. Simpson. Logical analysis of some theorems of combinatorics and topological dynamics. In Stephen G. Simpson, editor, *Logic and combinatorics*, pages 125–156. American Mathematical Society, Providence, R.I., 1987.

[Can97] Georg Cantor. Beiträge zur Begründung der transfiniten Mengenlehre. *Math. Ann.*, 49(2):207–246, 1897.

[Fef64] Solomon Feferman. Systems of predicative analysis. *J. Symbolic Logic*, 29:1–30, 1964.

[FMS82] Harvey Friedman, Kenneth McAloon, and Stephen G. Simpson. A finite combinatorial principle which is equivalent to the 1-consistency of predicative analysis. In George Metakides, editor, *Patras Logic Symposion*, pages 197–230. North-Holland, Amsterdam, 1982.

[FMW] Harvey Friedman, Antonio Montalbán, and Andreas Weiermann. A characterization of ATR_0 in terms of a Kruskal-like tree theorem. unpublished draft.

[Gen36] Gerhard Gentzen. Die Widerspruchsfreiheit der reinen Zahlentheorie. *Math. Ann.*, 112(1):493–565, 1936.

[Gir87] Jean-Yves Girard. *Proof theory and logical complexity.* Bibliopolis, Naples, 1987.

[Hir94] Jeffry L. Hirst. Reverse mathematics and ordinal exponentiation. *Ann. Pure Appl. Logic*, 66(1):1–18, 1994.

[McA85] Kenneth McAloon. Paris-Harrington incompleteness and progressions of theories. In Anil Nerode and Richard A. Shore, editors, *Proceedings of the AMS-ASL summer institute held in Ithaca, N.Y., June 28–July 16, 1982*, volume 42 of *Proceedings of Symposia in Pure Mathematics*, pages 447–460. American Mathematical Society, Providence, RI, 1985.

[MM09] Alberto Marcone and Antonio Montalbán. On Fraïssé's conjecture for linear orders of finite Hausdorff rank. *Ann. Pure Appl. Logic*, 160(3):355–367, 2009.

[Rat91] Michael Rathjen. The role of parameters in bar rule and bar induction. *J. Symbolic Logic*, 56(2):715–730, 1991.

[Rat06] Michael Rathjen. The art of ordinal analysis. In *International Congress of Mathematicians. Vol. II*, pages 45–69. Eur. Math. Soc., Zürich, 2006.

[RW] Michael Rathjen and Andreas Weiermann. Reverse mathematics and well-ordering principles. to appear in "Computability in Context: Computation and Logic in the Real World", S.B. Cooper and A. Sorbi (eds.), Imperial College Press.

[Sch77] Kurt Schütte. *Proof theory*. Springer-Verlag, Berlin, 1977. Translated from the revised German edition by J. N. Crossley, Grundlehren der Mathematischen Wissenschaften, Band 225.

[Sho06] Richard A. Shore. Invariants, Boolean algebras and ACA_0^+. *Trans. Amer. Math. Soc.*, 358(3):989–1014, 2006.

[Sim99] Stephen G. Simpson. *Subsystems of second order arithmetic*. Springer, 1999.

[Veb08] Oswald Veblen. Continuous increasing functions of finite and transfinite ordinals. *Trans. Amer. Math. Soc.*, 9(3):280–292, 1908.

Dipartimento di Matematica e Informatica, Università di Udine, 33100 Udine, Italy
E-mail address: `alberto.marcone@dimi.uniud.it`

Department of Mathematics, University of Chicago, Chicago, IL 60637, USA
E-mail address: `antonio@math.uchicago.edu`