Multitape Ordinal Machines and Primitive Recursion

Bernhard Irrgang and Benjamin Seyfferth

University of Bonn, Mathematical Institute
Beringstraße 1, D-53115 Bonn, Germany
irrgang@math.uni-bonn.de, benjamin.seyfferth@gmx.de

Abstract. We introduce a multitape version of the ordinal TURING machines which are defined in [4] as TURING machines computing on tapes of transfinite length in transfinite time. These machines are used to compute the primitive recursive ordinal functions which have a classical theory developed by JENSEN and KARP [3]. Making use of that theory we are able to 1. identify the $\mathbf{\Delta_1}(L_\alpha)$-definable subsets of α as computable (if α is closed with respect to primitive recursive functions) and 2. characterize admissible ordinals by the means of transfinite computations. Similar results linking α-recursion theory and ordinal computability are contained in [6].

Keywords: Ordinal computability, multitape TURING machine, primitive recursive ordinal function, primitive recursive set function, admissible ordinal.

1 Introduction

Ordinal computability studies machine models generalized to perform computations on ordinals. Such machines have been used to compute GÖDEL's hierarchy of constructible sets L ([4], [5]) and can provide a computational approach to α-recursion theory ([6]). The α-TURING machine is a standard TURING program computing on tapes of length a limit ordinal α using a lim inf-rule to determine the machine configuration at limit times. The machine is said to *terminate* if it runs for less than α many steps, otherwise it *diverges.*

In [3] JENSEN and KARP established a connection between the primitive recursive *ordinal* functions ($\mathrm{Prim_O}$, a generalization of the usual primitive recursive functions on natural numbers) and the primitive recursive *set* functions ($\mathrm{Prim_S}$) that are used in the theory of GÖDEL's constructible universe ([2]). The multitape α-TURING machines introduced in this paper are a straightforward multitape version of the α-TURING machine and are well suited to handle the calculus of $\mathrm{Prim_O}$ functions, i.e. every $\mathrm{Prim_O}$ function is also multitape α-computable (Theorem 1).

Using a result from [3] that the $\mathrm{Prim_O}$ functions are exactly the $\mathrm{Prim_S}$ functions that map ordinals to ordinals we prove Theorem 2: If α is an ordinal closed under $\mathrm{Prim_O}$ functions then the $\mathbf{\Delta}_1(L_\alpha[B])$ definable subsets of α are exactly the ones that are multitape α-computable in B. Furthermore we are able to give a characterization of *admissible ordinals* by the means of multitape α-TURING machines (Theorem 3): A limit ordinal is admissible iff there is no multitape α-computable function mapping some $\beta < \alpha$ cofinally into α. Admissible ordinals, which play an important role in generalizing recursion theory, are classically defined by the means of definability over a constructible level L_α or axiomatized through the axioms of KRIPKE-PLATEK set theory. We hence provide an alternative approach to admissibility from a computational perspective.

Similar theorems are already contained in [6] but obtained in a different way: A (single-tape) α-computable truth function for L_α is employed to transfer definability over L_α to the computational context. The computability of this truth function however requires α to be sufficiently closed with respect to ordinal arithmetic. Seeing this in a talk the second author gave in a seminar on ordinal computability in Bonn in November 2007, the first author suggested to make use of the well developed theory of primitive recursive ordinal and set functions and applied them in the proofs of Theorems 2 and 3. We kindly thank PETER KOEPKE for his suggestions and support of this work.

2 Multitape α-TURING Machines

A program P for a standard TURING-machine with k tapes (each with an independent read-write head) can be seen as a finite subset of $\{0,1\}^k \times \omega \times \{0,1\}^k \times \omega \times \{-1,+1\}^k$. An element $(a, s, a', s', d) \in P$ codes the following instruction: If the k-many read write heads read the symbols corresponding to the entries of the vector $a \in \{0,1\}^k$ and the machine is in state $s \in \omega$ then have the heads write the entries of $a' \in \{0,1\}^k$ to the respective tapes, change the machine state to $s' \in \omega$ and move the read-write heads according to the vector $d \in \{-1,+1\}^k$. Similarly to previous studies ([4, 6]) this can be used as a basis for the following notion of a transfinite computation according to P:

At successor times the program P is used as in the standard TURING machine case, with the single exception that when dealing with tapes of transfinite length a convention has to be found what should happen when a read-write-head is being moved left from a cell indexed by a limit ordinal. In this situation we want the head to be reset to the beginning of the tape (cell number '0').

For limit times the TURING-program cannot determine the tape content, head positions and program state so we have to define them in a sensible way. Following the lines of [4] we use inferior limits: We want each single cell of every tape to contain the $\liminf$ of its previous values, the machine state to be the $\liminf$ of the previous machine states and every tape's read/write-head to be located on the cell indexed by the $\liminf$ over the positions it previously assumed in the limit machine state, i.e. the least cell that was read cofinally often while the machine was in the same state as at the limit time.

More formally:

Definition 1. *Let α be a limit ordinal or $\alpha = \text{Ord}$. Let $P \subseteq \{0,1\}^k \times \omega \times \{0,1\}^k \times \omega \times \{-1,1\}^k$ be finite and let $T_0 = (T_{0,0}, T_{1,0}, \ldots, T_{(k-1),0}) \in ({}^{\alpha}\{0,1\})^k$*

be the initial tape content of the k-many tapes of length α. A triple

$$(T_\theta, H_\theta, S_\theta)_{\theta \leq \Theta}$$

*is a k-tape α-*TURING *computation by P on input T_0 if the following conditions hold:*

- $\Theta \leq \alpha$;
- $S_\theta \in \omega$ *for* $\theta \leq \Theta$;
- $H_\theta = (H_{0,\theta}, H_{1,\theta}, \ldots, H_{(k-1),\theta})$
  where $H_{i,\theta} \in \alpha$ *for* $0 \leq i < k$ *and for* $\theta \leq \Theta$;
- $T_\theta = (T_{0,\theta}, T_{1,\theta}, \ldots, T_{(k-1),\theta})$
  where $T_{i,\theta} : \alpha \longrightarrow \{0, 1\}$ *for* $0 \leq i < k$ *and for* $\theta \leq \Theta$;
- $(T_\theta, H_\theta, S_\theta)_{\theta \leq \Theta}$ *is defined recursively in P and the initial tape contents $T_{i,0}$ in the following way:*
  Termination: *Let $\theta \leq \Theta < \alpha$ and let $(T_{\theta'}, H_{\theta'}, S_{\theta'})_{\theta' \leq \theta}$ be already defined. If there is no $(a, s, a', s', d) \in P$ where $(T_{i,\theta}(H_{i,\theta}))_{i<k} = a$ and $S_\theta = s$ then the computation* terminates*, i.e. $\theta = \Theta$.*
  Successor step: *Let $\theta \leq \Theta$, let $(T_{\theta'}, H_{\theta'}, S_{\theta'})_{\theta' \leq \theta}$ be already defined and let there be a $c = (a, s, a', s', d) \in P$ where $(T_{i,\theta}(H_{i,\theta}))_{i<k} = a$ and $S_\theta = s$. Choose c minimally with respect to some fixed well-order on P. As usual we want the configuration $(T_{\theta+1}, H_{\theta+1}, S_{\theta+1})$ to be derived from $(T_\theta, H_\theta, S_\theta)$ according to the instruction c.*
  Let $a' = (a'_0, a'_1, \ldots, a'_{k-1})$. For all $i < k$ we require:

$$T_{i,\theta+1}(\xi) = \begin{cases} a_i & \text{, if } \xi = H_{i,\theta} \\ T_{i,\theta}(\xi) & \text{, else} \end{cases}$$

$$H_{i,\theta+1} = \begin{cases} H_{i,\theta} + 1 & \text{, if } d = +1 \\ H_{i,\theta} - 1 & \text{, if } d = -1 \text{ and } H_{i,\theta} \text{ is a successor ordinal} \\ 0 & \text{, if } d = -1 \text{ and } H_{i,\theta} \text{ is a limit ordinal} \end{cases}$$

$$S_{i,\theta+1} = s'.$$

  Limit step: *Now let $\theta \leq \Theta$ be a limit ordinal and let $(T_{\theta'}, H_{\theta'}, S_{\theta'})_{\theta' < \theta}$ be already defined. For $i < k$ and $\xi < \alpha$ set*

$$S_\theta = \liminf_{\theta' < \theta} S_{\theta'}$$

$$H_{i,\theta} = \liminf_{\theta' < \theta, S_\theta = S_{\theta'}} H_{i,\theta'}$$

$$T_{i,\theta}(\xi) = \liminf_{\theta' < \theta} T_{i,\theta'}(\xi).$$

  Note that the machine configuration at limit times is always defined whenever the configurations at all previous stages are defined. If $\theta = \Theta = \alpha$ we say that the computation diverges.

The primitive recursive ordinal functions we want to compute are n-ary functions mapping ordinals $< \alpha$ to α. So we define:

Definition 2. *Let $B \subseteq \alpha$ and $n \in \omega$. A function $f : \alpha^n \longrightarrow \alpha$ is called* k-α-computable *in B if there is a k-tape* TURING *program P and a finite sequence of ordinal parameters $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_m) \in \alpha^m$ s.t. $k \geq n + m + 2$ and for all $\boldsymbol{\xi} = (\xi_1, \xi_2, \dots, \xi_n) \in \alpha^n$ the k-tape α-*TURING *computation by P with input T_0 $(T_\theta, H_\theta, S_\theta)_{\theta \leq \Theta_{\boldsymbol{\xi}}}$ is of the form*

- *$\Theta_{\boldsymbol{\xi}} < \alpha$, i.e. the computation terminates;*
- *for every $i < n$ there is a read-only tape containing $\chi_{\{\xi_i\}}$;*
- *for every $j < m$ there is a read-only tape containing $\chi_{\{\pi_j\}}$;*
- *there is a read-only tape containing χ_B;*
- *all other tapes are initially empty;*
- *there is a tape that at time $\Theta_{\boldsymbol{\xi}}$ contains $\chi_{\{f(\boldsymbol{\xi})\}}$.*

If $B = \emptyset$ we call f k-α-computable.
A function $g : \alpha^n \longrightarrow \alpha$ is called multitape α-computable *if there is a k such that g is k-α-computable.*

The question arises which closure properties an ordinal α must have so that the notions of α-computable in B defined in [6] and multitape α-computable in B coincide. Clearly this is true for admissible ordinals, but much weaker closure properties will certainly suffice.

Lemma 1. *Let $f : \alpha^n \longrightarrow \alpha$ be k-α-computable by the program P in parameters $\boldsymbol{\pi}$. Let $(T_\theta, H_\theta, S_\theta)_{\theta \leq \Theta_{\boldsymbol{\xi}}}$ be the k-α-computation by P for $f(\boldsymbol{\xi})$. Then the function* $\mathrm{Time}_f : \alpha^n \longrightarrow \alpha$ *which maps $\boldsymbol{\xi} \mapsto \Theta_{\boldsymbol{\xi}}$ is $(k+1)$-α-computable.*

Proof. We extend P to a $(k+1)$-tape TURING program which still computes f but where every instruction in P additionally moves the $(k+1)$-st tape's head to the right. The computation terminates after $\Theta_{\boldsymbol{\xi}}$ many steps, i.e. there is no command for the final configuration. Add a new instruction for this configuration that writes a '1' at the current head position of the $(k+1)$-st tape. Since the $k+1$-st tape's head now points to a '1' instead of a '0' the computation terminates with this new instruction. It follows that Time_f is $(k+1)$-α-computable.

3 Primitive Recursive Functions

The familiar primitive recursive functions on natural numbers are those recursive functions generated by a weak recursion scheme that allows recursive definitions using the supremum over the previous function values. A generalization of this concept to functions operating on the universe of sets has for instance been used in the study of the constructible hierarchy ([2]). In [3] JENSEN and KARP defined the notion of primitive recursiveness for functions mapping ordinals to ordinals and developed their theory. Following [3] we define:

Definition 3. *The class* $\mathrm{Prim_O}(B)$ *of* primitive recursive ordinal functions in $B \subseteq \mathrm{Ord}$ *is defined as the minimal set containing all the functions of type (1) to (5) and closed under the schemes for substitution (a) and (b) and recursion (R).*

(1) $f(\xi) = \chi_B(\xi)$
(2) $\mathrm{pr}_{n,i}(\boldsymbol{\xi}) = \xi_i$, *for all* $n \in \omega$, $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$ *and* $1 \leq i \leq n$.
(3) $f(\xi) = 0$
(4) $f(\xi) = \xi + 1$
(5) $c(\xi, \zeta, \gamma, \delta) = \begin{cases} \xi, \textit{ if } \gamma < \delta \\ \zeta, \textit{ else} \end{cases}$
(a) $f(\boldsymbol{\xi}, \boldsymbol{\zeta}) = g(\boldsymbol{\xi}, h(\boldsymbol{\xi}), \boldsymbol{\zeta})$
(b) $f(\boldsymbol{\xi}, \boldsymbol{\zeta}) = g(h(\boldsymbol{\xi}), \boldsymbol{\zeta})$
(R) $f(\zeta, \boldsymbol{\xi}) = g(\sup\{f(\eta, \boldsymbol{\xi}) \mid \eta < \zeta\}, \zeta, \boldsymbol{\xi})$

We write $\mathrm{Prim_O}$ *instead of* $\mathrm{Prim_O}(\emptyset)$.

If $B \subseteq \alpha$ *and* α *is an ordinal that is closed under* $\mathrm{Prim_O}(B)$ *functions then we call a function* $f : \alpha^n \longrightarrow \alpha$ $\mathrm{Prim_O}(B)$ *iff it is the restriction of a* $\mathrm{Prim_O}(B)$ *function.*

Definition 3 is a special case of the following definition of primitive recursive set functions also taken from [3]:

Definition 4. *Let* X *be a one-place set function. The class* $\mathrm{Prim_S}(X)$ *of* primitive recursive set functions in X *is defined as the minimal set containing all the functions of type (1) to (5) and closed under the schemes for substitution (a) and (b) and recursion (R).*

(1) $F(x) = X(x)$
(2) $Pr_{n,i}(\boldsymbol{x}) = x_i$, *for all* $n \in \omega$, $\boldsymbol{x} = (x_1, \ldots, x_n) \in \alpha^n$ *and* $1 \leq i \leq n$.
(3) $F(x) = 0$
(4) $F(x, y) = x \cup \{y\}$
(5) $C(x, y, u, v) = \begin{cases} x, \textit{ if } u \in v \\ y, \textit{ else} \end{cases}$
(a) $F(\boldsymbol{x}, \boldsymbol{y}) = G(\boldsymbol{x}, H(\boldsymbol{x}), \boldsymbol{y})$
(b) $F(\boldsymbol{x}, \boldsymbol{y}) = G(H(\boldsymbol{x}), \boldsymbol{y})$
(R) $F(y, \boldsymbol{x}) = G(\sup\{F(z, \boldsymbol{x}) \mid z < y\}, y, \boldsymbol{x})$

We write $\mathrm{Prim_S}$ *instead of* $\mathrm{Prim_S}(\emptyset)$.

If B *is a set and* X *is the unary function* $X(x) = x \cap B$ *then we may write* $\mathrm{Prim_S}(B)$ *for* $\mathrm{Prim_S}(X)$.

If $B \subseteq \alpha$ *and* α *is an ordinal that is closed under* $\mathrm{Prim_O}(B)$ *functions then we call a function* $f : \alpha^n \longrightarrow \alpha$ $\mathrm{Prim_S}(B)$ *iff it is the restriction of a* $\mathrm{Prim_S}(B)$ *function.*

An n*-ary relation* $R \subseteq V^n$ *is* $\mathrm{Prim_S}(X)$ *if there is a* $\mathrm{Prim_S}(X)$ *function* $F_R : V^n \longrightarrow V$ *s.t.* $F_R(\boldsymbol{x}) = 0$ *iff* $\boldsymbol{x} \in R$.

Theorem 1. *If* α *is closed under* $\mathrm{Prim_O}$ *functions, then every* $\mathrm{Prim_O}(B)$ *function* $f : \alpha^n \longrightarrow \alpha$ *is multitape* α*-computable in* B. *Furthermore there is a* $\mathrm{Prim_O}$ *function* M_f *that majorizes* Time_f, *i.e.* $\forall \boldsymbol{\xi} \in \alpha^n$ $M(\boldsymbol{\xi}) > \mathrm{Time}_f(\boldsymbol{\xi})$.

Proof. Functions (1) to (4) are easily seen to be multitape α-computable with majorizing functions the maximum input ordinal plus 1.

(5) We define a program to compute C. Move the heads of the tapes containing γ and δ to the right to decide whether $\gamma < \delta$. According to the outcome move the output tape's head together with the head of the tape containing ξ or ζ to the right until a '1' is read. Copy the '1' to the output tape and stop. This program runs at most $M_C(\gamma, \delta, \xi, \zeta) = \max\{\gamma, \delta\} + \max\{\xi, \zeta\} + 1$-many steps. Since α is closed under $\mathrm{Prim}_{\mathrm{O}}$ functions and ordinal addition is $\mathrm{Prim}_{\mathrm{O}}$ the program terminates and C is multitape α-computable.

(a) Let g be k-α-computable by a program P_g with parameters $\boldsymbol{\pi_g}$ and majorizing function M_g, h l-α-computable by P_h with parameters $\boldsymbol{\pi_h}$ and majorized by function M_h. We define a $(k+l)$-tape TURING program that computes f using $\boldsymbol{\pi_g}\hat{}\boldsymbol{\pi_h}$ as parameters. The program first runs P_h to compute $h(\boldsymbol{\xi})$. Note that any program computing f in parameters $\boldsymbol{\pi_g}\hat{}\boldsymbol{\pi_h}$ uses tapes containing the components of $\boldsymbol{\xi}$, $\boldsymbol{\zeta}$, $\boldsymbol{\pi_h}$, $\boldsymbol{\pi_g}$ and one tape containing the characteristic function of B; these can be used as input tapes for P_h and later on for P_g respectively. After resetting all heads to position zero the program continues with running P_g with the output tape of P_h, now containing $h(\boldsymbol{\xi})$, as additional input tape. Resetting of heads takes only finitely many machine steps thanks to the well-foundedness of ordinals (to recognize head position '0' we may assume additional parameter tapes containing $\chi_{\{0\}}$). So the new Program will run in less than α many steps since addition of ordinals is $\mathrm{Prim}_{\mathrm{O}}$ and α is closed under $\mathrm{Prim}_{\mathrm{O}}$ functions. We can set $M_f = (M_g \cdot 2) + (M_h \cdot 2)$.

(b) Similarly to (a).

(R) Let g be k-α-computable by P_g in parameters $\boldsymbol{\pi_g}$ and majorized by M_g. We describe a $(k+2)$-tape TURING program that computes f with parameters $\boldsymbol{\pi_g}$. The algorithm runs through ζ-many *stages*. In stage $\eta < \zeta$ the new $(k+1)$-st tape contains the current value of $\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' < \eta\}$. P_g is called once to compute $f(\eta, \boldsymbol{\xi}) = g(\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' < \eta\}, \eta, \boldsymbol{\xi})$. If necessary the value of $\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\}$ has to be updated. The stage concludes by erasing all the work tapes used by P_g and resetting all heads to zero. We have to ensure that for all $\eta < \zeta$ the following conditions hold:

(i) At the time of the call of P_g to compute $f(\eta, \boldsymbol{\xi})$ the $(k+1)$-st tape contains in fact $\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\}$.
(ii) The number of machine steps the program uses before entering stage η is less than α.

For (i) use the $(k+2)$-nd tape to save the value of $f(\eta, \boldsymbol{\xi}) = \gamma$ not as $\chi_{\{\gamma\}}$ but as $\chi_{\gamma+1}$. At the beginning of every stage use tape $(k+2)$ to write a '1' to the $\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\}$-th cell of tape $(k+1)$ and use this as input to P_g. (i) holds inductively at successor stages. So let η be a limit ordinal. By the lim inf-rule tape $(k+2)$ contains in fact $\chi_{\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\}}$. So (i) holds.

Stage η consists of the following operations:

– Find the first 0 on tape $(k+2)$ and write a 1 to the respective cell of tape $(k+1)$ (note that the the first '0' occurs in cell number $\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' < \eta\}+1$):

As seen above the number of machine steps $\beta_{\text{find}} = \sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' < \eta\} + 2$ is less than α.

- Reset tape $(k+1)$'s head: This needs at most $\beta_{reset(k+1)} = \sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' < \eta$ many steps.
- Run P_g to compute $f(\eta, \boldsymbol{\xi}) = g(\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' < \eta\}, \eta, \boldsymbol{\xi})$, the number of steps $\beta_g = \text{Time}_g(\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' < \eta\}, \eta, \boldsymbol{\xi})$ is $(k+1)$-α-computable therefore $< \alpha$.
- Reset the head of P_g's output tape and update tape $(k+2)$ if necessary: This can be decided and done in at most $\beta_{\text{update}} = f(\eta, \boldsymbol{\xi}) \cdot 2 < \alpha$ many steps.
- Erase the work tapes used by P_g. WLOG P_g maintains a 'timer' tape as in Lemma 1 which we can use to determine up to which cell the tapes have to be erased. Since only cells up to index β_g may have been used by P_g (all heads were at position 0 when P_g was called) this takes again at most $\beta_g \cdot 2$ many steps ($\cdot 2$ since we have to reset the heads before erasing the tapes).
- Reset all heads. This takes at most $\max\{\beta_{find}, \beta_g\}$ many steps.

We define a majorizing function M_f for Time_f by Prim_{O} recursion:

$$\begin{aligned} M_f(\eta) = {} & \sup_{\eta' < \eta} M_f(\eta') \\ & + (\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\} + 2) \cdot 2 \\ & + M_g(\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\}, \eta) \\ & + f(\eta, \boldsymbol{\xi}) \cdot 2 \\ & + M_g(\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\}, \eta) \cdot 2 \\ & + \max\{\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\}, M_g(\sup\{f(\eta', \boldsymbol{\xi}) \mid \eta' \leq \eta\}, \eta, \boldsymbol{\xi})\} \end{aligned}$$

It follows from inductive analysis of the algorithm above that $\text{Time}_f < M_f$. Note that as supremum of the first η many values of a Prim_{O} function $\sup_{\eta' < \eta} M_f(\eta')$ is Prim_{O} and therefore $< \alpha$. So (ii) holds.

4 Applications

The following theorems are similar to Theorem 7 and 9 in [6] which provide a connection between α-recursion theory and ordinal computability. The proofs found in [6] explicitely give a truth function for bounded formulas in L_α which is α-computable if α is sufficiently closed with respect to ordinal arithmetic. Instead we use facts from the classical theory of Prim_{S} functions to obtain these results.

Theorem 2. *If α is an ordinal closed under* Prim_{O} *functions then $A \subseteq \alpha$ is $\boldsymbol{\Delta}_1(L_\alpha[B])$ iff A is multitape α-computable in B.*

Proof. '$\Leftarrow$' follows from the recursion theorem as in [6].
'$\Rightarrow$' Let $A \subseteq \alpha$ be $\boldsymbol{\Delta}_1(L_\alpha[B])$ by

$$\begin{aligned} \gamma \in A &\leftrightarrow L_\alpha[B] \models \exists x \, \phi[x, \gamma, \boldsymbol{p}] \\ \gamma \notin A &\leftrightarrow L_\alpha[B] \models \exists x \, \psi[x, \gamma, \boldsymbol{q}]. \end{aligned}$$

where ϕ, ψ are Σ_0^B formulas. So we have:

$$\begin{aligned} \gamma \in A &\leftrightarrow L_\alpha[B] \models \exists x\, \phi[x, \gamma, \boldsymbol{p}] \\ &\leftrightarrow \exists x \in L_\alpha[B]\ \phi[x, \gamma, \boldsymbol{p}] \end{aligned}$$

In [3], Lemma 3.2, it is shown that there is a one-one Prim_S function N mapping the ordinals onto the constructible sets s.t. $N \upharpoonright \alpha : \alpha \xrightarrow{bij} L_\alpha$. It is easily seen that there is also a one-one $\mathrm{Prim}_\mathrm{S}(B)$ function N' mapping the ordinals onto $L[B]$ s.t. $F = N' \upharpoonright \alpha : \alpha \xrightarrow{bij} L_\alpha[B]$. So we can write:

$$\leftrightarrow \exists \xi \in \alpha\ \phi[F(\xi), \gamma, \boldsymbol{F(\pi)}]$$

Like in [2], Lemma I.2.4, we see that every Σ_0^B relation is $\mathrm{Prim}_\mathrm{S}(B)$. Hence there is a $\mathrm{Prim}_\mathrm{S}(B)$ function G s.t. $G(\boldsymbol{z}) = 0$ iff $\phi[\boldsymbol{z}]$:

$$\begin{aligned} &\leftrightarrow \exists \xi \in \alpha\ G(F(\xi), \gamma, \boldsymbol{F(\pi)}) = 0 \\ &\leftrightarrow \exists \xi \in \alpha\ g(\xi, \gamma, \boldsymbol{\pi}) = 0. \end{aligned}$$

Where

$$g(\xi, \gamma, \boldsymbol{\pi}) = \begin{cases} 0 & \text{, if } G(F(\xi), \gamma, \boldsymbol{F(\pi)}) = 0 \\ 1 & \text{, else.} \end{cases}$$

Since G is $\mathrm{Prim}_\mathrm{S}(B)$ so is g by (a) and (5). g mapping ordinals to ordinals, $B \subset \alpha$, and we see like in Theorem 3.5 in [3] that g is $\mathrm{Prim}_\mathrm{O}(B)$. Similarly we obtain a function h for ψ. Now we can describe the following algorithm which computes the characteristic function of A:

```
WHILE ($g(\xi, \gamma, \boldsymbol{\pi}) = 1$ AND $h(\xi, \gamma, \boldsymbol{\eta}) = 1$) DO $\xi + +$;
IF $g(\xi, \gamma, \boldsymbol{\pi}) = 0$ THEN STOP $= 0$;
IF $h(\xi, \gamma, \boldsymbol{\pi}) = 0$ THEN STOP $= 1$;
```

We analyse the algorithm into its stages $\xi < \alpha$, each consisting mainly of one computation for g and one for h plus some erasing of work tapes and resetting of heads. We have to make sure that this algorithm behaves nicely at limit stages, i.e. actually reaches every limit stage $\delta < \alpha$. Again it will be neccessary to store the counter ξ as $\chi_{\xi+1}$ and to decode this into χ_ξ at the beginning of every stage. Similar to the proof of Theorem 1 we see that the algorithm up to stage ξ uses a number of steps majorized by a Prim_O function $M(\xi)$. Again also $\sup_{\zeta<\delta} M(\zeta)$ for $\delta < \alpha$ is Prim_O so the algorithm reaches every stage.

We can now give a characterization of admissible ordinals solely based on ordinal computations.

Theorem 3. *A limit ordinal α is admissible iff there is no multitape α-computable function mapping some $\beta < \alpha$ cofinally into α.*

Proof. α is admissible iff there is no $\mathbf{\Sigma}_1(L_\alpha)$-definable total function that maps some $\beta < \alpha$ cofinally into α (cf. [1], Lemma II.7.2).

If α is already closed under Prim_O functions then any multitape α-computable function f is $\mathbf{\Delta}_1(L_\alpha)$ (and therefore $\mathbf{\Sigma}_1(L_\alpha)$) by the recursion theorem as in [6] (cf. Theorem 2). Conversely let α be not admissible and let $f : \beta \xrightarrow{cof} \alpha$ be a $\mathbf{\Sigma}_1(L_\alpha)$-definable total function on β. So we have:

$$\begin{aligned} f(\gamma) = \delta &\leftrightarrow L_\alpha \models \exists x\, \phi[x, \gamma, \delta, \boldsymbol{p}] \\ &\leftrightarrow \exists x \in L_\alpha\, \phi[x, \gamma, \delta, \boldsymbol{p}] \end{aligned}$$

Where ϕ is a Σ_0 function. With $F : \alpha \xrightarrow{bij} L_\alpha$ Prim_S from Lemma 3.2 in [3]:

$$\leftrightarrow \exists \xi \in \alpha\, \phi[F(\xi), \gamma, \delta \boldsymbol{F(\pi)}]$$

Since ϕ is Σ_0 it is Prim_S ([2], Lemma I.2.4):

$$\begin{aligned} &\leftrightarrow \exists \xi \in \alpha\, G(F(\xi), \gamma, \delta, \boldsymbol{F(\pi)}) = 0 \\ &\leftrightarrow \exists \xi \in \alpha\, g(\xi, \gamma, \delta, \boldsymbol{\pi}) = 0. \end{aligned}$$

Where

$$g(\xi, \gamma, \delta, \boldsymbol{\pi}) = \begin{cases} 0 & \text{, if } G(F(\xi), \gamma, \delta, \boldsymbol{F(\pi)}) = 0 \\ 1 & \text{, else.} \end{cases}$$

Since G is Prim_S so is g by (a) and (5). g is mapping ordinals to ordinals and is therefore Prim_O by Theorem 3.5 in [3]. The desired algorithm goes through less than α many stages to compute $f(\gamma)$ given γ as input. In every stage $\eta < \alpha$ the algorithm computes the values of $g(\xi, \gamma, \delta, \boldsymbol{\pi})$ for all $\xi, \delta < \eta$. g is Prim_O so Time_g is multitape α-computable majorized by M_g which in turn is Prim_O. So $\sup_{\xi,\delta<\eta} M(\xi, \gamma, \delta, \boldsymbol{\pi})$ is less than α and every stage η is reached by the algorithm. At some stage one computation for g will return 0 and the value $\delta = f(\gamma)$ is found. So f is multitape α-computable as required.

In the case that α is not closed under Prim_O functions we will define $\beta < \alpha$ and a multitape α-computable total function $f : \beta \xrightarrow{cof} \alpha$.

Any limit ordinal is closed under (1)-(4) so assume in the following closure under (1)-(4). If α is not closed under (5) then also for one instance of (5) $f(\zeta_0, \zeta_1, \zeta_2, \zeta_3)$ the canonical algorithm will not terminate in less than α many steps. Analysing the algorithm in the proof of Theorem 1 we can extract two ordinals $\gamma, \delta < \alpha$ with $\gamma + \delta \geq \alpha$ (set $\gamma = \max\{\zeta_2, \zeta_3\}$ and $\delta = \max\{\zeta_0, \zeta_1\}$). So there is a $\beta \leq \delta$ such that the multitape α-computable function $f : \beta \longrightarrow \alpha$, $\xi \mapsto \gamma + \xi$ is cofinal in α.

If α is closed under two functions f and g so it is also closed under $f(\boldsymbol{\xi}, \boldsymbol{\zeta}) = g(\boldsymbol{\xi}, h(\boldsymbol{\xi}), \boldsymbol{\zeta})$ and under $f'(\boldsymbol{\xi}, \boldsymbol{\zeta}) = g(h(\boldsymbol{\xi}), \boldsymbol{\zeta})$. So if α is closed under (1)-(5) but not closed under Prim_O functions it has to be not closed under (R).

Assume α closed under (1)-(5),(a),(b) but not closed under (R). Since the Prim_O functions are defined by recursion, there are Prim_O functions f, g s.t.

$f(\zeta, \boldsymbol{\xi}) = g(\sup\{f(\eta, \boldsymbol{\xi}) \mid \eta < \zeta\}, \zeta, \boldsymbol{\xi})$ is an instance of (R) and α closed under g but not closed under f. Choose β minimally s.t. $f(\beta, \boldsymbol{\xi}) \notin \alpha$. $f(\beta, \boldsymbol{\xi}) = g(\sup_{\gamma<\beta} f(\gamma), \boldsymbol{\xi}), \beta, \boldsymbol{\xi})$ and since α is closed under g we have that $\sup_{\gamma<\beta} f(\gamma, \boldsymbol{\xi}) = \alpha$. Now $f \restriction \beta : \beta \longrightarrow \alpha$ is cofinal in α.

References

1. Keith Devlin. *Constructibility*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1984.
2. Keith Devlin. *Aspects of Constructibility*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, Heidelberg, 1973.
3. Ronald B. Jensen and Carol Karp. *Primitive recursive set functions.* In: Axiomatic Set Theory, Proceedings of Symposia in Pure Mathematics, Volume XIII, Part I. American Mathematical Society, Providence, Rhode Island, 1971.
4. Peter Koepke. *Turing computations on ordinals.* The Bulletin of Symbolic Logic 11 (2005), 377–397.
5. Peter Koepke and Martin Koerwien. *Ordinal computations.* Mathematical Structures in Computer Science (2006), 867–884.
6. Peter Koepke and Benjamin Seyfferth. *Ordinal Machines and Admissible Recursion Theory.* Submitted to: Annals of Pure and Applied Logic, CiE 2007 special volume, to be published 2008.