α-Recursion Theory and Ordinal Computability

by Peter Koepke

University of Bonn

13.2.2007

Abstract

Motivated by a talk of S.D.Friedman at BIWOC we show that the α-recursive and α-recursively enumerable sets of G. Sacks's α-recursion theory are exactly those sets that are recursive and recursively enumerable by an ordinal Turing machines with tapes of length α and time bound α.

1 Introduction.

α-Recursion theory is a branch of higher recursion theory that was developed by G. Sacks and his school between 1965 and 1980. Sacks gave the following characterization [4]:

> α-recursion theory lifts classical recursion theory from ω to an arbitrary Σ_1 admissible ordinal α. Many of the classical results lift to every α by means of recursive approximations and fine structure techniques.

The lifting is based on the observation that a set $A \subseteq \omega$ is recursively enumerable iff it is Σ_1 definable over $(H_\omega, \in)$, the set of all hereditarily finite sets. By analogy, a set $A \subseteq \alpha$ is called *α-recursively enumerable* iff it is $\mathbf{\Sigma}_1(L_\alpha)$, i.e., definable in parameters over $(L_\alpha, \in)$ where L_α is the α-th level of Gödel's constructible hierarchy. Consequently a set $A \subseteq \alpha$ is said to be *α-recursive* iff it is $\mathbf{\Delta}_1(L_\alpha)$. Sacks discusses the "computational character" of $\mathbf{\Sigma}_1(L_\alpha)$-definitions [4]:

> The definition of f can be thought of as a *process*. At *stage* δ it is assumed that all *activity* at previous stages is encapsulated in an α-finite object, $s \restriction \delta$. In general it will be necessary to *search* through L_α for some existential witness ... [emphases by P.K.].

In this note we address the question whether it is possible to base α-recursion theory on some idealized computational model.

Let us fix an admissible ordinal α , $\omega < \alpha \leqslant \infty$ for the rest of this paper. A standard Turing computation may be visualized as a time-like sequence of elementary *read-write-move* operations carried out by "heads" on "tapes". The sequence of actions is determined by the initial tape contents and by a finite Turing *program*. We may assume that the Turing machine acts on a tape whose cells are indexed by the set ω $(=\mathbb{N})$ of *natural numbers* $0, 1, \ldots$ and contain 0's or 1's. A computation takes place in $\omega \times \omega$ "spacetime":

			S P A C E								
		0	1	2	3	4	5	6	7	...	...
	0	1	0	0	1	1	1	0	0	0	0
	1	0	0	0	1	1	1	0	0		
T	2	0	0	0	1	1	1	0	0		
I	3	0	0	1	1	1	1	0	0		
M	4	0	1	1	1	1	1	0	0		
E	⋮										
	n	1	1	1	1	0	1	1	1		
	$n+1$	1	1	1	1	1	1	1	1		
	⋮										

A standard TURING *computation. Head positions are indicated by shading.*

Let us now generalize TURING computations from $\omega \times \omega$ to an $\alpha \times \alpha$ space-time: consider TURING tapes whose cells are indexed by α (= the set of all ordinals $< \alpha$) and calculations which are sequences of elementary tape operations indexed by ordinals $< \alpha$. For successor times, calculations will basically be defined as for standard TURING machines. At limit times tape contents, program states and head positions are defined by *inferior limits*.

		S p a c e α															
		0	1	2	3	4	5	6	7	...	...	ω	...	θ	θ	...	...
	0	1	1	0	1	0	0	1	1	...	...	1	...	1	0	0	0
	1	0	1	0	1	0	0	1	1			1					
T	2	0	0	0	1	0	0	1	1			1					
i	3	0	0	0	1	0	0	1	1			1					
m	4	0	0	0	0	0	0	1	1			1					
e	⋮																
	n	1	1	1	1	0	1	0	1			1					
α	n+1	1	1	1	1	1	1	0	1			1					
	⋮	⋮	⋮	⋮	⋮	⋮											
	ω	0	0	1	0	0	0	1	1	...	...	1					
	$\omega+1$	0	0	1	0	0	0	1	1			0					
	⋮																
	$\theta<\alpha$	1	0	0	1	1	1	1	0	...	...	...	...	0	...	...	...
	⋮			⋮			⋮			⋮	⋮						
	⋮																

A computation of an α-TURING *machine.*

This leads to an α-*computability theory* with natural notions of α-*computable* and α-*computably enumerable* subsets of α. We show that α-computability largely agrees with α-recursion theory:

Theorem 1. *Consider a set $A \subseteq \alpha$. Then*

a) A is α-recursive iff A is α-computable.

b) A is α-recursively enumerable iff A is α-computably enumerable.

One can also define what it means for $A \subseteq \alpha$ to be *α-computable in an oracle* $B \subseteq \alpha$ and develop a theory of *α-degrees*. The reduction by α-computation is coarser than the standard reducibility used in α-recursion theory:

Theorem 2. *Consider sets A, $B \subseteq \alpha$ such that A is weakly α-recursive in B. Then A is α-computable in B.*

The relationship between ordinal TURING machines and the constructible model L was studied before [2]. We shall make use of those results by restricting them to α. It should be noted that we could have worked with ordinal *register* machines instead of TURING machines to get the same results [3]. The present work was inspired by S.D.FRIEDMAN's talk on α-recursion theory at the BIWOC workshop.

2 α -TURING Machines

The intuition of an α-TURING machine can be formalized by restricting the definitions of [2] to α.

Definition 3.

a) A command *is a 5-tuple $C=(s, c, c', m, s')$ where $s, s' \in \omega$ and $c, c', m \in \{0, 1\}$; the natural number s is the* state *of the command C. The intention of the command C is that if the machine is in state s and reads the symbol c under its read-write head, then it writes the symbol c', moves the head left if $m = 0$ or right if $m = 1$, and goes into state s'. States correspond to the "line numbers" of some programming languages.*

b) A program *is a finite set P of commands satisfying the following structural conditions:*

i. If $(s, c, c', m, s') \in P$ then there is $(s, d, d', n, t') \in P$ with $c \neq d$; thus in state s the machine can react to reading a "0" as well as to reading a "1".

ii. If $(s, c, c', m, s') \in P$ and $(s, c, c'', m', s'') \in P$ then $c' = c''$, $m = m'$, $s' = s''$; this means that the course of the computation is completely determined by the sequence of program states and the initial cell contents.

c) For a program P let

$$\text{states}(P) = \{s \mid (s, c, c', m, s') \in P\}$$

be the set of program states.

Definition 4. *Let P be a program. A triple*

$$S\colon\theta\to\omega,\ H\colon\theta\to\alpha,\ T\colon\theta\to({}^{\alpha}2)$$

*is an α-*computation *by P iff the following hold:*

a) *θ is a successor ordinal $<\alpha$ or $\theta=\alpha$; θ is the* length *of the computation.*

b) *$S(0)=H(0)=0$; the machine starts in state 0 with head position 0.*

c) *If $t<\theta$ and $S(t)\notin\mathrm{state}(P)$ then $\theta=t+1$; the machine* stops *if the machine state is not a program state of P.*

d) *If $t<\theta$ and $S(t)\in\mathrm{state}(P)$ then $t+1<\theta$; choose the unique command $(s, c, c', m, s')\in P$ with $S(t)=s$ and $T(t)_{H(t)}=c$; this command is executed as follows:*

$$\begin{aligned} T(t+1)_{\xi} &= \begin{cases} c'\ , & \text{if } \xi=H(t);\\ T(t)_{\xi}\ , & \text{else;}\end{cases}\\ S(t+1) &= s';\\ H(t+1) &= \begin{cases} H(t)+1, & \text{if } m=1;\\ H(t)-1, & \text{if } m=0 \text{ and } H(t) \text{ is a successor ordinal;}\\ 0, & \text{else.}\end{cases}\end{aligned}$$

e) *If $t<\theta$ is a limit ordinal, the machine constellation at t is determined by taking inferior limits:*

$$\begin{aligned} \forall\xi\in\mathrm{Ord}\ T(t)_{\xi} &= \liminf_{r\to t} T(r)_{\xi};\\ S(t) &= \liminf_{r\to t} S(r);\\ H(t) &= \liminf_{s\to t, S(s)=S(t)} H(s).\end{aligned}$$

*The α-computation is obviously recursively determined by the initial tape contents $T(0)$ and the program P. We call it the α-*computation by *P* with input *$T(0)$. If the α-computation stops, $\theta=\beta+1$ is a successor ordinal and $T(\beta)$ is the final tape content. In this case we say that P* computes *$T(\beta)$ from $T(0)$ and write P: $T(0)\mapsto T(\beta)$.*

Sets $A\subseteq\alpha$ may be coded by their characteristic functions $\chi_A\colon\alpha\to 2$, $\chi_x(\xi)=1$ iff $\xi\in A$.

Definition 5. *A partial function $F\colon\alpha\rightharpoonup\alpha$ is α-*computable *iff there is a program P and a finite set $p\subseteq\alpha$ of* parameters *such that for all $\delta<\alpha$:*

- *if $\delta\in\mathrm{dom}(F)$ then the α-computation with initial tape contents $T(0)=\chi_{p\cup\{2\cdot\delta\}}$ stops and P: $\chi_{p\cup\{2\cdot\delta\}}\mapsto\chi_{\{F(\delta)\}}$; note that we use "even" ordinals to code the input δ, the parameter set p would typically consist of "odd" ordinals;*
- *if $\delta\notin\mathrm{dom}(F)$ then the α-computation with initial tape contents $T(0)=\chi_{p\cup\{2\cdot\delta\}}$ does not stop.*

A set $A \subseteq \alpha$ *is* α-computable *iff its characteristic function* $\chi_A: \alpha \to 2$ *is* α*-computable. A set* $A \subseteq \alpha$ *is* α-computably enumerable *iff* $A = \mathrm{dom}(F)$ *for some* α*-computable partial function* $F: \alpha \rightharpoonup 2$.

3 α-computations inside L_α

In general, recursion theory subdivides recursions and definitions into minute elementary computation steps. Thus computations are highly *absolute* between models of (weak) set theories and we get:

Lemma 6. *Let* P *be a program and let* $T(0): \alpha \to 2$ *be an initial tape content which is* Σ_1*-definable in* $(L_\alpha, \in)$ *from parameters. Let* $S: \theta \to \omega$, $H: \theta \to \alpha$, $T: \theta \to ({}^\alpha 2)$ *be the* α*-computation by* P *with input* $T(0)$*. Then:*

- *a)* S, H, T *is the* α*-computation by* P *with input* $T(0)$ *as computed* in the model $(L_\alpha, \in)$.
- *b)* S, H, T *are* Σ_1*-definable in* $(L_\alpha, \in)$ *from parameters.*
- *c)* *If* $A \subseteq \alpha$ *is* α*-recursively enumerable then it is* $\Sigma_1(L_\alpha)$ *in parameters.*
- *d)* *If* $A \subseteq \alpha$ *is* α*-recursive then it is* $\Delta_1(L_\alpha)$ *in parameters.*

So we have proved one half of the Equivalence Theorem 1.

4 The bounded truth predicate for L_α

For the converse we have to analyse KURT GÖDEL's constructible hierarchy using ordinal computability. The inner model L of *constructible sets* is defined as the union of a hierarchy of levels L_δ:

$$L = \bigcup_{\delta \in \mathrm{Ord}} L_\delta$$

where the hierarchy is defined by: $L_0 = \emptyset$, $L_\delta = \bigcup_{\gamma < \delta} L_\gamma$ for limit ordinals δ, and $L_{\gamma+1}$ = the set of all sets which are first-order definable with parameters in the structure $(L_\gamma, \in)$. The standard reference to the theory of the model L is the book [1] by K. DEVLIN. We consider in particular the model

$$L_\alpha = \bigcup_{\gamma < \alpha} L_\gamma$$

To make L_α accessible to an α-TURING machine we introduce a language with symbols (,), {, }, |, $\in$, $=$, $\wedge$, $\neg$, $\forall$, $\exists$ and variables $v_0, v_1, \ldots$. Define (*bounded*) *formulas* and (*bounded*) *terms* by a common recursion on the lenghts of words formed from these symbols:

- the variables $v_0, v_1, \ldots$ are terms;
- if s and t are terms then $s = t$ and $s \in t$ are formulas;

– if φ and ψ are formulas then $\neg\varphi$, $(\varphi \wedge \psi)$, $\forall v_i \in v_j\, \varphi$ and $\exists v_i \in v_j\, \varphi$ are formulas;

– if φ is a formula then $\{v_i \in v_j \mid \varphi\}$ is a term.

For terms and formulas of this language define *free* and *bound variables*:

– $\mathrm{free}(v_i) = \{v_i\}$, $\mathrm{bound}(v_i) = \emptyset$;

– $\mathrm{free}(s = t) = \mathrm{free}(s \in t) = \mathrm{free}(s) \cup \mathrm{free}(t)$;

– $\mathrm{bound}(s = t) = \mathrm{bound}(s \in t) = \mathrm{bound}(s) \cup \mathrm{bound}(t)$;

– $\mathrm{free}(\neg\varphi) = \mathrm{free}(\varphi)$, $\mathrm{bound}(\neg\varphi) = \mathrm{bound}(\varphi)$;

– $\mathrm{free}((\varphi \wedge \psi)) = \mathrm{free}(\varphi) \cup \mathrm{free}(\psi)$, $\mathrm{bound}((\varphi \wedge \psi)) = \mathrm{bound}(\varphi) \cup \mathrm{bound}(\psi)$;

– $\mathrm{free}(\forall v_i \in v_j\, \varphi) = \mathrm{free}(\exists v_i \in v_j\, \varphi) = \mathrm{free}(\{v_i \in v_j \mid \varphi\}) = (\mathrm{free}(\varphi) \cup \{v_j\}) \setminus \{v_i\}$;

– $\mathrm{bound}(\forall v_i \in v_j\, \varphi) = \mathrm{bound}(\exists v_i \in v_j\, \varphi) = \mathrm{bound}(\{v_i \in v_j \mid \varphi\}) =$
$= \mathrm{bound}(\varphi) \cup \{v_i\}$.

For technical reasons we will be interested in terms and formulas in which

– no bound variable occurs free,

– every free variable occurs exactly once.

Such terms and formulas are called *tidy*; with tidy formulas one avoids having to deal with the interpretation of one free variable at different positions within a formula.

An *assignment* for a term t or formula φ is a finite sequence $a: k \to V$ so that for every free variable v_i of t or φ we have $i < k$; $a(i)$ will be the *interpretation* of v_i. The *value* of t or the *truth value* of φ is determined by the assignment a. We write $t[a]$ and $\varphi[a]$ for the values of t und φ under the assignment a.

Concerning the constructible hierarchy L, it is shown by an easy induction on γ that every element of L_γ is the interpretation $t[(L_{\gamma_0}, L_{\gamma_1}, \ldots, L_{\gamma_{k-1}})]$ of some *tidy* term t with an assignment $(L_{\gamma_0}, L_{\gamma_1}, \ldots, L_{\gamma_{k-1}})$ whose values are constructible levels L_{γ_i} with $\gamma_0, \ldots, \gamma_{k-1} < \gamma$. This will allow to reduce bounded quantifications $\forall v \in L_\gamma$ or $\exists v \in L_\gamma$ to the substitution of terms of lesser complexity. Moreover, the truth of (bounded) formulas in L is captured by *tidy* bounded formulas of the form $\varphi[(L_{\gamma_0}, L_{\gamma_1}, \ldots, L_{\gamma_{k-1}})]$.

We shall code an assignment of the form $(L_{\gamma_0}, L_{\gamma_1}, \ldots, L_{\gamma_{k-1}})$ by its sequence of ordinal indices, i.e., we write $t[(\gamma_0, \gamma_1, \ldots, \gamma_{k-1})]$ or $\varphi[(\gamma_0, \gamma_1, \ldots, \gamma_{k-1})]$ instead of $t[(L_{\gamma_0}, L_{\gamma_1}, \ldots, L_{\gamma_{k-1}})]$ or $\varphi[(L_{\gamma_0}, L_{\gamma_1}, \ldots, L_{\gamma_{k-1}})]$. The relevant assignments are thus elements of $\mathrm{Ord}^{<\omega}$.

We define a bounded truth function W for the constructible hierarchy on the class

$$A = \{(a, \varphi) \mid a \in \mathrm{Ord}^{<\omega}, \varphi \text{ is a tidy bounded formula}, \mathrm{free}(\varphi) \subseteq \mathrm{dom}(a)\}$$

of all “tidy pairs” of assignments and formulas. Define the *bounded constructible truth function* $W: A \to 2$ by

$$W(a, \varphi) = 1 \text{ iff } \varphi[a].$$

In [2] we showed:

Lemma 7. *The bounded truth function W for the constructible universe is ordinal computable.*

Restricting all considerations to α yields

Lemma 8. *The bounded truth function $W\upharpoonright L_\alpha$ for L_α is α-computable.*

This yields the Equivalence Theorem 1:

Lemma 9. *If $A\subseteq\alpha$ is $\Sigma_1(L_\alpha)$ in parameters then A is α-computably enumerable. If $A\subseteq\alpha$ is $\Delta_1(L_\alpha)$ in parameters then A is α-computable.*

Proof. Consider a $\Sigma_1(L_\alpha)$-definition of $A\subseteq\alpha$:

$$\xi\in A\leftrightarrow\exists y\in L_\alpha\, L_\alpha\vDash\varphi[\xi,y,\vec{a}]$$

where φ is a bounded formulas. This is equivalent to

$$\xi\in A\leftrightarrow\exists\beta<\alpha\, L_\beta\vDash\exists y\varphi[\xi,y,\vec{a}]$$

and

$$\xi\in A\leftrightarrow\exists\beta<\alpha\, W((\xi,\beta,\vec{a}),\varphi^*)$$

where φ^* is an appropriate tidy formula.

Now A is α-computably enumerable, due to the following “search procedure”: for $\xi<\alpha$ search for the smallest $\beta<\alpha$ such that

$$W((\xi,\beta,\vec{a}),\varphi^*);$$

if the search succeeds, stop, otherwise continue.

For the second part, let $A\subseteq\alpha$ be $\Delta_1(L_\alpha)$ in parameters. Then A and $\alpha\setminus A$ are α-computably enumerable. By standard arguments, A is α-computable. □

5 Reducibilities

The above considerations can all be relativized to a given oracle set $B\subseteq\alpha$. One could, e.g., provide B on an extra input tape. This leads to a natural reducibility

$$A\prec B \text{ iff } A \text{ is } \alpha\text{-computable in } B.$$

Note that so far we have not really used the admissibility of α but only that α is closed under ordinal multiplication. We obtain:

Proposition 10. *$A\prec B$ iff A is $\Delta_1(L_\alpha(B))$ in parameters, where $(L_\delta(B))_{\delta\in\mathrm{Ord}}$ is the constructible hierarchy relativized to B.*

The α-recursion theory of [4] uses the following two reducibilities for subsets of α:

Definition 11.

a) *A is* weakly α-recursive in *B, $A \leqslant_{w\alpha} B$, iff there exists an α-recursively enumerable set $R \subseteq L_\alpha$ such that for all $\gamma < \alpha$*

$$\gamma \in A \text{ iff } \exists H \subseteq B \exists J \subseteq \alpha \setminus B \, (H, J, \gamma, 1) \in R$$

and

$$\gamma \notin A \text{ iff } \exists H \subseteq B \exists J \subseteq \alpha \setminus B \, (H, J, \gamma, 0) \in R.$$

b) *A is* α-recursive in *B, $A \leqslant_\alpha B$, iff there exist α-recursively enumerable sets $R_0, R_1 \subseteq L_\alpha$ such that for all $K \in L_\alpha$*

$$K \subseteq A \text{ iff } \exists H \subseteq B \exists J \subseteq \alpha \setminus B \, (H, J, K) \in R_0$$

and

$$K \subseteq \alpha \setminus A \text{ iff } \exists H \subseteq B \exists J \subseteq \alpha \setminus B \, (H, J, K) \in R_1 .$$

It is easy to see that $A \leqslant_\alpha B$ implies $A \leqslant_{w\alpha} B$. If $A \leqslant_{w\alpha} B$ then an inspection of the conditions an part a) of the definition shows immediately that A is $\Delta_1(L_\alpha(B))$, i.e., $A \prec B$, which proves Theorem 2.

We *conjecture* that POST's problem holds for $\prec$: there are α-computably enumerable sets $A, B \subseteq \alpha$ such that

$$A \not\prec B \text{ and } B \not\prec A.$$

This would immediately yield the SACKS-SIMPSON theorem [5]

$$A \not\leqslant_{w\alpha} B \text{ and } B \not\leqslant_{w\alpha} A$$

which is the positive solution to POST's problem in α-recursion theory.

Bibliography

[1] Keith Devlin. *Constructibility*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1984.

[2] Peter Koepke. Turing computations on ordinals. *The Bulletin of Symbolic Logic*, 11:377–397, 2005.

[3] Peter Koepke and Ryan Siders. Register computations on ordinals. *submitted to: Archive for Mathematical Logic*, 14 pages, 2006.

[4] Gerald E. Sacks. *Higher Recursion Theory*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin Heidelberg, 1990.

[5] Gerald E. Sacks and Stephen G. Simpson. The α-finite injury method. *Annals of Mathematical Logic*, 4:343–367, 1972.