universitätbonn

B I W O C
Bonn International Workshop on Ordinal Computability

January 21-25, 2007, Bonn, Germany

Workshop report

Organised by
Joel David Hamkins from the City University of New York and
Peter Koepke from the University of Bonn, with the aid of the

hausdorff center for mathematics

Report prepared by Ioanna Matilde Dimitriou

Introduction

Ordinal computability theory is based on ordinal numbers, just as standard computability theory is based on natural numbers: we clock the steps of a computation by ordinal numbers instead of natural numbers (*infinite* or *ordinal* "time"), index the cells of a Turing tape by ordinal numbers (*infinite* or *ordinal* "space"), read and write ordinals as register values (ordinal "space") and so on. After sporadic early experiments, a comprehensive and active theory of ordinal computability has begun to emerge since the late 1990's: Infinite time Turing machines (ITTMs) generate a host of results and problems in analogy and contradistinction to classical computability theory; by iterating the trivial Turing operations along the ordinals, we are able to define and analyze complicated and attractive sets of reals numbers. The success of ITTMs has motivated other ordinal generalizations of classical models of computations which may be classified according to the following schema:

standard space & time	standard space, ordinal time	ordinal space, ordinal time
Turing machine	ITTM $\Delta^1_1 \subsetneq$ computable $\subsetneq \Delta^1_2$	Ordinal Turing machine OTM, computable $\leftrightarrow$ constructible
register machine	Infinite time register machine ITRM, computable $\leftrightarrow$ hyperarithmetic	Ordinal register machine ORM, computable $\leftrightarrow$ constructible

Various classes of sets become computable under the different models of computability. Thus ordinal computability theory takes the paradigm of computability to descriptive set theory, constructibility theory and other fields in innovative and fruitful ways.

The aim of the Bonn International Workshop on Ordinal Computability (BIWOC) was to bring together a majority of researchers in ordinal computability as well as interested graduate students and outstanding representatives from neighboring fields. The program consisted of morning talks, which represented the current spectrum of ordinal machine models and identified relevant research questions and projects. Afternoons gave space for informal discussions and collaborations and were concluded by one-hour "informal slots". These consisted of presentations decided at the workshop, a discussion session about the future of ordinal computability, and a problem session. Progress on problems posed was already made during the workshop or shortly afterwards.

This workshop report contains extended abstracts of most of the talks at the workshop, a list of problems, some of them solved in the meantime, and further material inspired by questions from the workshop. Information about the workshop can also be found electronically at `www.math.uni-bonn.de/people/logic/biwoc`.

The workshop was made possible through generous financial support by the new Hausdorff Center for Mathematics. The workshop was administrated professionally and efficiently by Dagmar Böttcher, Luiza Jakuszek and Anke Thiedemann at the office of the Hausdorff Research Institute and by Dorothee Reuther at the office of the Bonn Logic Group. Several student helpers from the Hausdorff Institute provided friendly and unobtrusive catering and technical assistence. Bernhard Irgang, besides participating in the scientific program, perfectly ran and coordinated all technical aspects at the conference site. The conference location was kindly provided by the Mathematical Institute. Holger Hammes, the unfailing Hausmeister of the Institute, helped in several respects. Ioanna Dimitriou edited the workshop booklet and this report competently and swiftly.

We have to thank the Hausdorff Center for Mathematics, the Mathematical Institute and all the helpers mentioned for their magnificent work and support, which led to a very positive and productive workshop atmosphere. We also thank the participants for their contributions and interactions, which made the workshop a scientific success and proved that ordinal computability will continue to thrive, and that it is challenging, interdisciplinary, interactive, and fun.

Joel D. Hamkins and Peter Koepke

March 2007

Contents

Conference photo

List of participants

Klaus Ambos-Spies, Heidelberg	Merlin Carl, Bonn
Sam Coskey, Rutgers University	Barnaby Dawson, Bristol
Ioanna Dimitriou, Bonn	Sy Friedman, Vienna
Gunter Fuchs, Münster	Alexander Gilbers, Bonn
Victoria Gitman, New York	Joel Hamkins, New York
Daisuke Ikegami, Amsterdam	Bernhard Irrgang, Bonn
Marek Karpinski, Bonn	Michael Klein, Bonn
Ansten Klev, Amsterdam	Peter Koepke, Bonn
David Linetsky, New York	Benedikt Löwe, Amsterdam
Russell Miller, New York	Wolfram Pohlers, Münster
Michael Rathjen, Leeds	Jonas Reitz, New York
Alexander Rothkegel, Bonn	Ralf Schindler, Münster
Daniel Seabold, Hofstra University	Benjamin Seyfferth, Bonn
Ryan Siders, Helsinki	Jip Veldman, Bonn
Thomas Waltke, Bonn	Steve Warner, Hofstra University
Philip Welch, Bristol	Joost Winter, Amsterdam

Schedule

Sunday, January 21, 2007

10:00-10:10	Peter Koepke Welcome, Opening Remarks
10:10-11:10	Joel Hamkins A Survey of Infinite Time Turing Machines (page 14)
11:30-12:30	Peter Koepke Ordinalize! (page 21)
13:00-15:00	*Buffet lunch at China City, Clemens-August-Strasse 2*
16:00-16:30	*Afternoon coffee and tea*
16:30-17:30	Informal slot: Philip Welch Turing Unbound: on the extent of computation in Malament -Hogarth spacetimes

Monday, January 22, 2007

09:30-10:45	Sy Friedman Recursion Theory on an Admissible Ordinal (page 11)
11:15-12:30	Philip Welch Descriptive set theoretic aspects of ordinal computability (page 36)
16:00-16:30	*Afternoon coffee and tea*
16:30-17:30	Informal slot: Michael Rathjen Proof theory

Tuesday, January 23, 2007

09:30-10:30	Ralf Schindler The P vs. NP problem for infinite time Turing machines (page 29)
11:00-11:30	Ryan Siders Ordinal register machines.
12:00-12:30	David Linetsky The complexity of quickly decidable ORM sets (page 26)
12:35	*Workshop photograph*
16:00-16:30	*Afternoon coffee and tea*
16:30-17:30	Informal slot: Open discussion Aspects of Ordinal Computability (page 55)

Wednesday, January 24, 2007

09:30-10:30	Wolfram Pohlers Recursion in higher types.
10:45-11:30	Steve Warner Infinite time computable model theory I (page 31)
11:45-12:30	Daniel Seabold Infinite time computable model theory II (page 31)
15:30-16:00	*Afternoon coffee and tea*
16:00-17:30	Informal slot: Joost Winter, Problem session (page 42)
17:45-18:30	*Informal walking tour of Bonn*
18:30-??:??	*Conference dinner at DACAPO, Beethovenhalle*

Thursday, January 25, 2007

09:30-10:00	Barnaby Dawson Extensions of ordinal time computers (page 9)
10:30-11:00	Sam Coskey Infinite time computable equivalence relation theory (page 8)
11:30-12:30	Russell Miller Post's problem for Ordinal Register Machines (page 27)
15:30-16:00	*Afternoon coffee and tea*

Extended abstracts

Infinite Time Decidable Equivalence Relations

Sam Coskey
Rutgers University
Tuesday, 10:45 – 11:15

Introduction. There is a theory of Borel equivalence relations revolving around the following complexity notion. Let E, F be equivalence relations on $\mathbb{R}$. Write that $E \leq_B F$ if there exists a Borel map $f : \mathbb{R} \to \mathbb{R}$ such that for $x, y \in \mathbb{R}$

$$xEy \leftrightarrow f(x)Ff(y)$$

The function f is said to reduce E to F.
We instead consider several weaker notions made possible by the Hamkins-Kidder ITTM model [2]. We say that a function f is (boldface) computable if there is an ITTM program p and a real paramater z such that $f = \phi_p^z$. Now write that $E \leq_C F$ if there is a computable reduction from E to F.
Additionally, we say that a function f is eventually computable if there is a program which on input x eventually writes $f(x)$ on the tape (it need not halt). We say that f is semicomputable if its graph is decidable. The corresponding reduction notions are written $E \leq_{EC} F$ and $E \leq_{SC} F$, respectively.
These are weakenings of the Borel notion; every Borel function is computable (from a real parameter) and every computable function is semicomputable. On the other hand, each of these functions has a $\mathbf{\Delta}_2^1$ graph and so these notions are not too far from Borel.
Our first observation, and a strong motivation for this investigation, is the following. The $\leq_C$ notion offers a very similar picture to the $\leq_B$ theory, while the $\leq_{SC}$ notion collapses nearly every computable equivalence relation to a point under $V = L$. Thus we are right on a boundary; we may be considering the most powerful reductions that can still distinguish important complexity classes under ZFC.
We are presently motivated by the following goals. The first is to discover the exact extent of the analogy between $\leq_B$ and $\leq_C$ on the Borel equivalence relations. The second is to use the notion of computable complexity to distinguish some interesting relations of high complexity (where Borel reducibility is not appropriate).

Borel equivalence relations. We have the following dichotomy.

Meta-theorem. *A large fragment of the Borel complexity picture carries over to the computable case.*

Theorem 1. *If $V = L$ then any computable equivalence relation E semicomputably reduces to $=$.*

The Theorem above appears in [3], but let us discuss the Meta-theorem. For example, consider the countable Borel equivalence relations, for which there is a nice (and hard-won) Borel complexity picture. The least complex countable Borel equivalence relation is $=$. Its immediate successor is called E_0 and the most complex is called E_∞. The remaining ones lie in the interval (E_0, E_∞). It has been shown by Adams-Kechris [1] that every Borel partial ordering embeds into this interval.
We wish to show that the $\leq_C$ picture is similar. Of course every Borel reduction is also a computable reduction, so we need only concern ourselves with showing that the non-reductions are preserved.

But it is the case that many of the non-reduction proofs in the Borel case overshoot and show there is no measurable reduction. And moreover, all computable functions are measurable. So for instance there is no computable reduction from E_∞ to E_0 or from E_0 to $=$. We also obtain for free the analog of the Adams-Kechris theorem. It remains of fundamental importance to answer the following question. Are there Borel equivalence relations E, F such that $E \leq_C F$ but $E \not\leq_B F$?

Relations of high complexity. We now move on to our second aim, that of identifying and classifying interesting equivalence relations of possibly high complexity. First, let us introduce a pair of equivalence relations which are naturally occurring and have a different relationships with respect to the Borel and computable complexity notions. We define that $x \equiv_{WO} y$ if x, y code the same ordinal (by some fixed means). Next define $x \equiv_{CK} y$ if x, y have the same finite-time computable ordinals. Then we have the following:

Theorem 2. *The equivalence relations* $\equiv_{WO}$ *and* $\equiv_{CK}$ *are computably bireducible but Borel incomparable.*

Here are a couple more interesting relations. Write that $x \equiv_{SET} y$ if x and y, thought of as countable sequences of reals, have the same range. This relation lies above every countable Borel equivalence relation. Write that $x \equiv_{HC} y$ if x and y code the same hereditarily countable set. Then each of $\equiv_{SET}$ and $\equiv_{WO}$ computably reduce to $\equiv_{HC}$. We hope to show that these reductions are strict.
Another important relation is the Turing degree relation $\equiv_{\omega_1}$, defined by $x \equiv_{\omega_1} y$ if x is computable from y by an ITTM and vice versa. It is semidecidable but not decidable. So it does not computably reduce to any computable relation, but it is unknown if even $=$ reduces to it.
On the other hand, $\equiv_{\omega_1}$ does eventually reduce to $\equiv_{SET}$. Given x just run all programs and at each stage put the results into a set. This program will eventually converge on the $\equiv_{\omega_1}$ class of x but it will not halt. So $\leq_{EC}$ provides a slightly more stable picture of more complex relations.

References

[1] Adams, Scot and Alexander Kechris. Linear Algebraic Groups and Countable Borel Equivalence Relations. J. AMS 13.4, 2000.
[2] Hamkins, Joel, and Andy Lewis. Infinite time Turing machines. Journal of Symbolic Logic 65.2, 2000.
[3] Hamkins, Joel, et. al. Infinite time computable model theory.

Computability and Constructibility.

Barnaby L Dawson
University of Bristol
Thursday, 9:30 – 10:00

Ordinal time Turing machines have recently become the subject of some study. This paper will look at plausible extensions of ordinal time Turing machines from the perspective of what sets and class functions may become computable with their usage.

Ordinal time Turing machines run for ordinal periods of time and have ordinal amount of memory. Each memory cell may contain a 0 or a 1. At limit stages the state of the machine is fixed using lim-inf operations.

These machines don't allow us to compute the class function giving us beth numbers. In addition other class functions are also non-computable despite having simple definitions. If we want to investigate the model theory and syntax of infinite languages calculating beth numbers come in handy. Finally the notion of constructibility ties in well with ordinal time Turing computers. Is this just an artifact of the precise way they are defined? This is the main question we shall address in the talk.

1. Examples of possible extensions

Here are some examples of possible extensions of our computers:

- In the presence of sharps we could allow an operation which writes to the input tape $A^\sharp$ where A is currently on the output tape.
- We could introduce an operation that moves a read/write head to the next limit ordinal.
- We could introduce an operation that moves a read/write head to the next cardinal as seen from V (or relativize to L)
- We could introduce an operation that writes onto the input tape a bijection B from the current set A on the output tape to its cardinality. Again we could relativize to L.
- We could introduce a clock that resets the current instruction at certain times (perhaps at each cardinal).

2. Ordinal computers with oracles

If we imagine another computer with an extra input tape on which we can have an ordinal sized sequence of 0's and 1's this computer will have extended abilities. All of the above extensions can be simulated by a computer with an oracle input tape added.

However, these computers may have very different complexity theories attached to their operation. In this talk we shall concentrate on what may be computed not how long that might take.

In the talk we shall show that any first order definable notion of computation (respecting AC) can be simulated by an ordinal time Turing computer with an oracle.

We shall show that this notion is highly arbitrary allowing for any set to be considered as computational and any class function to be computational too.

We discuss the problems with this definition and come up with two ways of overcoming them.

3. Definitions of weakly and strongly computational

- Our weak ordering $\leq$ is this: $F \leq G$ if $range(F) \supseteq range(G)$ and G is unbounded.

- Our strong ordering $\prec$ is this: $F \preceq G$ if G majorises F (and $F \neq G$).

- A set S is weakly computational if $\exists F : \forall G \geq F$: S can be computed on a computer with G as an oracle.

- A set S is strongly computational if $\exists F : \forall G \succeq F$: S can be computed on a computer with G as an oracle.

4. The importance of fast growing functions

We shall show how the growth rate of an oracle (defined in the talk) can be the important feature determining what may be computed with that oracle. We will show how the arbitrary nature of extensions is overcome by the above definitions.

In addition we will show that any oracle that grows faster than the cardinals allows us to compute the cardinals (as seen from L).

5. Some interesting questions

What can we say about the weakly/strongly computational sets? Ideally we want to know how these fit in with the constructible universe and with large cardinal hypotheses.

Is every set that is constructible also strongly computational? We remark that this is trivially the case. Is it consistent for a set to be strongly/weakly computational but not constructible? A simple proof is given that it is consistent (assuming the consistency of $0^\sharp$) that there is a non computational but weakly computational set. A similar proof is given to show the same is true for strong computationality

Can a set be strongly/weakly computable in an inner model M but not in the universe V itself? It is shown that $0^\sharp$ is strongly computational in any model containing it.

Further directions of research are considered.

Recursion theory on an admissible ordinal.

Sy-David Friedman
KGRC, University of Vienna
Monday, 9:30 – 10:30

An interesting generalisation of classical recursion theory grew out of the Metarecursion Theory of Kreisel-Sacks, which drew the analogies

$$\text{RE} \approx \Pi^1_1$$
$$\text{Finite} \approx \text{Hyperarithmetic}$$

In α-recursion theory, the natural numbers are replaced by ordinals less than α, computations take place using such ordinals and have length less than α. One way to make this precise is via Kripke's generalisation of Kleene's equation calculus, which I now describe. For the moment, let α be any infinite ordinal.

The α-Equation Calculus

Symbols
A name for β (each $\beta < \alpha$)
Variables $x, y, z, ...$
Function symbols $f, g, h, ...$
$\exists$, $<$

Terms
$f(t_0, ..., t_n)$
$(\exists x < t_0)t_1$

Equations
$t = u$ where t, u are terms

Rules (where e denotes an equation)
From $e(x)$ infer $e(\beta)$
From $e(f(\beta_1, ..., \beta_n))$ and $f(\beta_1, ..., \beta_n) = \beta$ infer $e(\beta)$
(For $\gamma < \beta < \alpha$) From $t(\gamma) = 0$ infer $(\exists x < \beta)t(x) = 0$
(For $\beta < \alpha$) From $t(\gamma) = 1$ for all $\gamma < \beta$ infer $(\exists x < \beta)t(x) = 1$

A partial function F from α to α is *partial α-recursive* iff there is a finite set E of equations and a function symbol f such that:

$$F(\beta) = \gamma \text{ iff } f(\beta) = \gamma \text{ is derivable from } E.$$

A desirable property, called *recursive regularity* or *admissibility*, is:

If F is total α-recursive and β is less than α then the Range of F on β is bounded in α.

Regular infinite cardinals are clearly admissible. In fact all infinite cardinals are admissible. But there are many admissible ordinals which are not cardinals; indeed any uncountable cardinal is a limit of admissible ordinals.

The set-theoretic viewpoint

So far we have worked with just ordinals less than α and functions from α to α. However we must broaden our perspective if we are to properly understand α-RE sets.

A subset of α is *α-RE* iff it is the domain of a partial α-recursive function. A is *α-recursive* iff both A and $\alpha \setminus A$ are α-RE.

In classical recursion theory, an RE set A has an *increasing recursive enumeration*, i.e., A is the union of a sequence $\langle A^s \mid s \in \omega \rangle$ where:

i. Each A^s is finite.
ii. $s < t \to A^s$ is a subset of A^t.
iii. The function $s \mapsto A^s$ is recursive.

To lift this property to α-recursion theory, we must develop a notion of α-finite set of ordinals and α-recursive function from α to α-finite sets. This is best accomplished using Gödel's L-hierarchy. For any ordinal β, L_β denotes the β-th level of this hierarchy.

Theorem 1. *i. α is admissible iff L_α satisfies Σ_1 replacement.*
ii. For admissible α, the partial α-recursive functions are precisely those partial functions from α to α which are Σ_1 definable over L_α.

Suppose now that α is admissible.

Definitions. A partial function F from L_α to L_α is *partial α-recursive* iff its graph is Σ_1 over L_α. A subset A of L_α is *α-RE* iff it is Σ_1 over L_α and is *α-recursive* iff it is Δ_1 over L_α. A set is *α-finite* iff it is an element of L_α.

Theorem 2. *i. A set is α-finite iff it is both α-recursive and bounded, i.e., a subset of some L_β, $\beta < \alpha$.*
ii. If A is α-RE then A is the union of $\langle A^\sigma \mid \sigma < \alpha \rangle$, where each A^σ is α-finite, $\sigma < \tau \to A^\sigma \subseteq A^\tau$ and the function $\sigma \mapsto A^\sigma$ is α-recursive.

All of the basic results of classical recursion theory hold for an admissible ordinal α, with RE, recursive, finite replaced by α-RE, α-recursive, α-finite. In particular there is a universal α-RE set $W(e,x)$, i.e., W is α-RE and every α-RE set A equals $W_e = \{x \mid W(e,x)\}$ for some e.

Turing α-degrees

For $A \subseteq L_\alpha$, the set $N(A)$ of *neighbourhood conditions on A* is defined by

$$N(A) = \{(c,d) \mid c \subseteq A \text{ and } d \subseteq L_\alpha \setminus A\}.$$

Suppose A, B are subsets of L_α. Then A is *α-RE in B* iff for some α-RE set W_e

$$A = W_e^B = \{x \mid \text{There exists } (c,d) \in N(B) \text{ such that } (x,c,d) \in W_e\}.$$

A is *weakly α-recursive in B* iff both A and $L_\alpha \setminus A$ are α-RE in B. (This relation is not transitive in general.) A is *α-recursive in B*, written $A \leq_\alpha B$, iff both $\{c \mid c \subseteq A\}$ and $\{c \mid c \subseteq L_\alpha \setminus A\}$ are α-RE in B. The *α-jump* A' of A is the set $\{e \mid e \in W_e^A\}$. A' has the largest α-degree of a set α-RE in A.

Positive results

The ordinal recursion theory school of Sacks succeeded in lifting many results from classical recursion theory to α-recursion theory. If α is very admissible, e.g., if L_α satisfies Σ_n replacement for $n = 3$, then the proofs from the classical case readily generalise. With more work and sometimes very subtle arguments, the following positive results were obtained.

Theorem 3. *(Sacks-Simpson) For any admissible α, there are α-RE sets of incomparable α-degree.*

Theorem 4. *(Shore) For any admissible α, the α-degrees of α-RE sets are dense.*

Theorem 5. *(Shore) Any α-RE set which is not α-recursive is the union of two α-RE sets with smaller α-degree.*

Unexpected negative results

It was at first thought that with enough work, all results from classical recursion theory would lift to α-recursion theory. This turned out to be dramatically false. An α-RE set M is *maximal* iff for each α-RE set A, either $A \setminus M$ is α-finite or $L_\alpha \setminus (M \cup A)$ is α-finite. If α equals ω then there is a maximal set.

Theorem 6. *(Lerman) If there is a maximal α-RE set then α is countable.*

Thus the cardinality of α is relevant for α-recursion theory. A second negative result reveals the importance of cofinality. For simplicity, assume that $V = L$.

Theorem 7. *(SDF) If α is $\aleph_{\omega_1}$ then α-degrees above $0'$ are well-ordered, with successor given by the α-jump.*

This is certainly a picture which differs radically from the classical case! Thus although recursion theory can be interestingly generalised to many ordinals α greater than ω, some similarity with ω (i.e., regularity) is required for a reasonable theory. Many open questions remain, such as whether there is a minimal α-degree or whether there is a minimal pair of α-RE degrees for each admissible α.

A survey of infinite time Turing machines

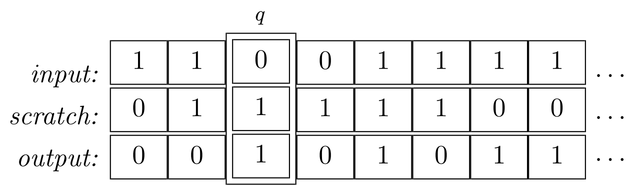
Joel David Hamkins[1]
City University of New York
Sunday, 10:10 – 11:10

Infinite time Turing machines extend the operation of ordinary Turing machines into transfinite ordinal time, thereby providing a natural model of infinitary computability, with robust notions of computability and decidability on the reals, while remaining close to classical concepts of computability. Here, I survey the theory of infinite time Turing machines and recent developments. These include the rise of infinite time complexity theory, the introduction of infinite time computable model theory and the study of the infinite time analogue of Borel equivalence relation theory, via infinite time computable equivalence relations and reductions. The study of infinite time Turing machines increasingly relies on the interaction of methods from set theory, descriptive set theory and computability theory.
Infinite time Turing machines were first considered by Hamkins and Kidder in 1989, with the principal introductory article provided by Hamkins and Lewis [HL00]. The theory has now been extended by many others, including Philip Welch, Benedikt Löwe, Daniel Seabold, Ralf Schindler, Vinay Deolalikar, Russell Miller, Steve Warner, Giacomo Lenzi, Erich Monteleone, Peter Koepke and others. Numerous precursors to the theory include Blum-Shub-Smale machines (1980s), Büchi machines (1960s) and accompanying developments, Barry Burd's model of Turing machines with "blurs" at limits (1970s) and the extensive development of α-recursion and E-recursion theory, a part of higher recursion theory, studied since the 1970s. More recent are Jack Copeland's accelerated Turing machines (1990s), Ryan Bissell-Siders' ordinal machines (1990s), with further recent developments by Peter Koepke (2000s), including ordinal tape infinite time Turing machines and ordinal register machines. The expanding literature involving infinite time Turing machines includes [HL00], [Wel99], [Wel00a], [Wel00b], [L0̈1], [HS01], [HL02], [Sch03], [HW03], [Ham02], [Ham04], [LM04], [DHS05], [HMSW07], [Ham05], [Wel], [Wel05], [Koe05] and others.

[1]Joel David Hamkin's research has been supported in part by grants from the Research Foundation of CUNY and by grants from the Netherlands Organization for Scientific Research (NWO). The author is also grateful to the University of Amsterdam ILLC for the support of a Visiting Professorship during his 2007 sabbatical from CUNY

0. A Brief Review of Infinite time Turing machines

Let us quickly review the basic operation of the machines and some key concepts. An infinite time Turing machine has the same hardware as its classical finite time counterpart, with a head moving on a semi-infinite paper tape, writing 0s and 1s in accordance with the rigid instructions of a finite program having finitely many states. For convenience, we have used a three tape model, with separate tapes for input, scratch work and output. At successor stages of computation, the machine



operates in exactly the classical manner, according to the program instructions. The new part of the computational behavior comes at limit ordinal stages. At any limit stage ξ, the machine is placed into the special *limit* state, one of the distinguished states alongside the *start* and *halt* states; the head is reset to the left-most cell; and the tape is updated by placing in each cell the $\limsup$ of the values previously displayed in that cell. This completely specifies the configuration of the machine at stage ξ, and the computation may continue to stage $\xi + 1$ and so on. Output is given only when the machine explicitly attains the *halt* state, and computation ceases when this occurs.

Since there is plenty of time for the machines to write out and inspect infinite binary strings, the natural context for input and output to the machines is Cantor space 2^ω, which I shall denote by $\mathbb{R}$ and refer to as the reals. The machines therefore provide an infinitary notion of computability on the reals. Program p computes the partial function $\varphi_p : \mathbb{R} \to \mathbb{R}$, defined by $\varphi_p(x) = y$ if program p on input x yields output y. A subset $A \subseteq \mathbb{R}$ is *infinite time decidable* if its characteristic function is infinite time computable. The set A is *infinite time semi-decidable* if the constant function $1 \upharpoonright A$ is computable. This is equivalent to A being the domain of an infinite time computable function (but not necessarily equivalent to A being the range of such a function). Initial results in [HL00] show that the arithmetic sets are exactly those that are decidable in time uniformly less than ω^2 and the hyperarithmetic sets are those that are decidable in time less than some recursive ordinal. Every Π^1_1 set is decidable, and the class of decidable sets is contained in Δ^1_2.

An easy cofinality argument establishes that every computation either halts or repeats by some countable ordinal stage. An ordinal α is *clockable* if there is a computation $\varphi_p(0)$ halting on exactly the α^{th} step. A real x is *writable* if it is the output of a computation $\varphi_p(0)$, and an ordinal is writable if it is coded by such a real. There are of course only countably many clockable and writable ordinals, because there are only countably many programs. Both the clockable and writable ordinals extend through all the recursive ordinals and far beyond; their supremum is recursively inaccessible and more. While the writable ordinals form an initial segment of the ordinals, there are gaps in the clockable ordinals, intervals of non-clockable ordinals below the supremum of the clockable ordinals. The gap structure itself becomes quite complicated, with limits of gaps sometimes being gaps and so on, and ultimately it exhibits the same complexity as the infinite time version of the halting problem. Nevertheless, [Wel00b] established that the supremum of the clockable and writable ordinals is the same. A real x is *eventually writable* if there is a computation $\varphi_p(0)$ for which x appears on the output tape from some

point on (even if the computation does not halt), and x is *accidentally writable* if it appears on any of the tapes at any stage during a computation $\varphi_p(0)$. By coding ordinals with reals, we obtain the notions of eventually and accidentally writable ordinals. If λ is the supremum of the clockable or writable ordinals, ζ is the supremum of the eventually writable ordinals and Σ is the supremum of the accidentally writable ordinals, then [HL00] establishes $\lambda < \zeta < \Sigma$. Welch [Wel00a] showed that $L_\lambda \prec_{\Sigma_1} L_\zeta \prec_{\Sigma_2} L_\Sigma$, and furthermore, these ordinals are characterized as the least example of this pattern.
Many of the fundamental constructions of classical finite time computability theory carry over to the infinite time context. For example, one can prove the infinite time analogues of the *smn*-theorem, the Recursion theorem and the undecidability of the infinite time halting problem, by essentially the classical arguments. Some other classical facts, however, do not directly generalize. For example, it is not true in the infinite time context that if the graph of a function f is semi-decidable, then the function is computable. This is a consequence of the following:

Theorem 1 (Lost Melody Theorem)**.** *There is a real c such that $\{c\}$ is infinite time decidable, but c is not writable.*

The real c is like the lost melody that you can recognize yes-or-no when someone sings it to you, but which you cannot sing on your own; it is a real that exhibits sufficient internal structure that $\{c\}$ is decidable, but is too complicated itself to be writable. If $f(x) = c$ is the function with constant value c, then f is not computable because c is not writable, but the graph is decidable, because we can recognize whether a pair has the form (x, c).
The infinite time analogue of the halting problem breaks into lightface and boldface versions, $h = \{\, p \mid \varphi_p(p)\downarrow \,\}$ and $H = \{\, (p, x) \mid \varphi_p(x)\downarrow \,\}$, respectively. These are both semi-decidable and not decidable, but in the infintary context, they are not computably equivalent.
There are two natural sorts of oracles to be used in oracle computations. First, one can use any real as an oracle in exactly the classical manner, by adjoining an oracle tape on which the values of that real are written out. Second, one naturally wants somehow to use a *set* of reals as oracle; but we cannot expect in general to write such a set out on the tape (perhaps it is even uncountable). Instead, the oracle tape is initially empty, and during the computation the machine may freely write on this tape. Whenever the algorithm calls for it, the machine may make a membership query about whether the real currently written on the oracle tape is a member of the oracle or not. Thus, the machine is able to know of any real that it can produce, whether the real is in the oracle set or not.
The result is a notion of relative computabiliy $\varphi_p^A(x)$, a notion of reduction $A <_\infty B$ and a notion of equivalence $A \equiv_\infty B$, with a rich theory of the infinite time Turing degrees. For any set A, we have the lightface jump $A^\triangledown$ and the boldface jump $A^\blacktriangledown$, corresponding to the two halting problems. One can show $A <_\infty A^\triangledown <_\infty A^\blacktriangledown$, as well as $A^{\triangledown\blacktriangledown} \equiv_\infty A^\blacktriangledown$ and a great number of other interesting interactions. In [HL02], we settled the infinite time analogue of Post's problem, the question of whether there are intermediate semi-decidable degrees between 0 and the jump $0^\triangledown$. The answer cuts both ways:

Theorem 2. *The infinite time analogue of Post's problem has both affirmative and negative solutions.*

(1) *There are no reals z with $0 <_\infty z <_\infty 0^\triangledown$.*
(2) *There are sets of reals A with $0 <_\infty A <_\infty 0^\triangledown$. Indeed, there are incomparable semi-decidable sets of reals $A \perp_\infty B$.*

In other work, Welch [Wel99] found minimality in the infinite time Turing degrees. Hamkins and Seabold [HS01] analyzed one-tape versus multi-tape infinite time Turing machines, and Benedikt Löwe [LÖ1] observed the connection between infinite time Turing machines and revision theories of truth.

1. Survey of recent developments

Let me now discuss some of the recent developments in the theory of infinite time Turing machines, including the rise of infinite time complexity theory, the introduction of infinite time computable model theory and the beginnings of infinite time computable equivalence relation theory.

1.1. Infinite time complexity theory.

Ralf Schindler [Sch03] initiated the study of infinite time complexity theory by solving the infinite time Turing machine analogue of the P versus NP question. To define the polynomial class P in the infinite time context, Schindler observed simply that all reals have length ω and the polynomial functions of ω are bounded by those of the form ω^n. Thus, he defined that a set $A \subseteq \mathbb{R}$ is in P if there is a program p and a natural number n such that p decides A and halts on all inputs in time before ω^n. The set A is in NP if there is a program p and a natural number n such that $x \in A$ if and only if there is y such that p accepts (x, y), and p halts on all inputs in time less than ω^n. Schindler proved $\mathrm{P} \neq \mathrm{NP}$ for infinite time Turing machines in [Sch03], using methods from descriptive set theory to analyze the complexity of the classes P and NP. This has now been generalized in joint work [DHS05] to the following, where the class co-NP consists of the complements of sets in NP.

Theorem 3. $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$ *for infinite time Turing machines.*

Some of the structural reasons behind $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$ are revealed by placing the classes P and NP within a larger hierarchy of complexity classes P_α and NP_α using computations of size bounded below α. We proved, for example, that the classes NP_α are identical for $\omega + 2 \leq \alpha \leq \omega_1^{\mathrm{CK}}$, but nevertheless, $\mathrm{P}_{\alpha+1} \subsetneq P_{\alpha+2}$ for any clockable limit ordinal α. It follows, since the P_α are steadily increasing while the classes $\mathrm{NP}_\alpha \cap \text{co-NP}_\alpha$ remain the same, that $\mathrm{P}_\alpha \subsetneq \mathrm{NP}_\alpha \cap \text{co-NP}_\alpha$ for any ordinal α with $\omega + 2 \leq \alpha < \omega_1^{\mathrm{CK}}$. Thus, $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$. Nevertheless, we attain equality at the supremum ω_1^{CK} with

$$\mathrm{P}_{\omega_1^{\mathrm{CK}}} = \mathrm{NP}_{\omega_1^{\mathrm{CK}}} \cap \text{co-NP}_{\omega_1^{\mathrm{CK}}}.$$

In fact, this is an instance of the equality $\Delta^1_1 = \Sigma^1_1 \cap \Pi^1_1$.
This same pattern of inequality $\mathrm{P}_\alpha \subsetneq \mathrm{NP}_\alpha \cap \text{co-NP}_\alpha$ is mirrored higher in the hierarchy, whenever α lies strictly within a contiguous block of clockable ordinals, with the corresponding $\mathrm{P}_\beta = \mathrm{NP}_\beta \cap \text{co-NP}_\beta$ for any β that begins a gap in the clockable ordinals. In addition, the question is settled in [DHS05] for the other complexity classes P^+, P^{++} and P^f. Benedikt Löwe has introduced analogues of PSPACE.

1.2. Infinite time computable model theory.

Computable model theory is model theory with a view to the computability of the structures and theories that arise. Infinite time computable model theory carries out this program with the notion of infinite time computability provided by infinite time Turing machines. The classical theory began decades ago with such topics as computable completeness (Does every decidable theory have a decidable model?) and computable categoricity (Does every isomorphic pair of computable models have a computable isomorphism?), and the field has now matured into a sophisticated analysis of the complexity spectrum of countable models and theories.

The motivation for a broader context is that, while classical computable model theory is necessarily limited to countable models and theories, the infinitary computability context allows for uncountable models and theories, built on the reals. Many of the computational constructions in computable model theory generalize from structures built on $\mathbb{N}$, using finite time computability, to structures built on $\mathbb{R}$, using infinite time computability. The uncountable context opens up new questions, such as the infinitary computable Löwenheim-Skolem Theorem, which have no finite time analogue. Several of the most natural questions turn out to be independent of ZFC.

In joint work [HMSW07], we defined that a model $\mathcal{A} = \langle A, \ldots \rangle$ is infinite time *computable* if $A \subseteq \mathbb{R}$ is decidable and all functions, relations and constants are uniformly infinite time computable from their Gödel codes and input. The structure $\mathcal{A}$ is *decidable* if one can compute whether $\mathcal{A} \models \varphi[\vec{a}]$ given $\ulcorner\varphi\urcorner$ and $\vec{a}$. A theory T is infinite time *decidable* if the relation $T \vdash \varphi$ is computable in $\ulcorner\varphi\urcorner$. Because we want to treat uncountable languages, the natural context for Gödel codes is $\mathbb{R}$ rather than $\mathbb{N}$.

The initial question, of course, is the infinite time computable analogue of the Completeness Theorem: Does every consistent decidable theory have a decidable model? The answer turns out to be independent of ZFC.

Theorem 4 ([HMSW07])**.** *The infinite time computable analogue of the Completeness Theorem is independent of* ZFC*. Specifically:*

(1) *If $V = L$, then every consistent infinite time decidable theory has an infinite time decidable model, in a computable translation of the language.*

(2) *It is relatively consistent with ZFC that there is an infinite time decidable theory, in a computably presented language, having no infinite time computable or decidable model in any translation of the language.*

The proof of (1) uses the concept of a *well-presented* language $\mathcal{L}$, for which there is an enumeration of the symbols $\langle s_\alpha \mid \alpha < \delta \rangle$ such that from any $\ulcorner s_\alpha \urcorner$ one can uniformly compute a code for the prior symbols $\langle \ulcorner s_\beta \urcorner \mid \beta \leq \alpha \rangle$. One can show that every consistent decidable theory in a well-presented language has a decidable model, and if $V = L$, then every computable language has a well presented computable translation. For (2), one uses the theory T extending the atomic diagram of $\langle WO, \equiv \rangle$ while asserting that f is a choice function on the $\equiv$ classes. This is a decidable theory, but for any computable model $\mathcal{A} = \langle A, \equiv, f \rangle$ of T, the set $\{ f(c_u) \mid u \in WO \}$ is Σ^1_2 and has cardinality ω_1. It is known to be consistent with ZFC that no Σ^1_2 set has size ω_1.

For the infinite time analogues of the Löwenheim-Skolem Theorem, we proved for the upward version that every well presented infinite time decidable model has a proper elementary extension with a decidable presentation, and for the downward version, every well presented uncountable decidable model has a countable decidable elementary substructure. There are strong counterexamples to a full direct generalization of the Löwenheim-Skolem theorem, however, because [HMSW07] provides a computable structure $\langle \mathbb{R}, U \rangle$ on the entire set of reals, which has no proper computable elementary substructure.

Some of the most interesting work involves computable quotients. A structure has an infinite time computable *presentation* if it is isomorphic to a computable structure, and has a computable *quotient presentation* if it is isomorphic to the quotient of a computable structure by a computable equivalence relation (a congruence). For structures on $\mathbb{N}$, in either the finite or infinite time context, these notions are equivalent, because one can computably find the least element of any equivalence class. For structures on $\mathbb{R}$, however, computing such distinguished elements of every equivalence class is not always possible.

Question 5. *Does every structure with an infinite time computable quotient presentation have an infinite time computable presentation?*

In the finite time theory, or for structures on $\mathbb{N}$, the answer of course is Yes. But in the full infinite time context for structures on $\mathbb{R}$, the answer depends on the set theoretic background.

Theorem 6. *The answer to Question 5 is independent of* ZFC*. Specifically,*

(1) *It is relatively consistent with* ZFC *that every structure with an infinite time computable quotient presentation has an infinite time computable presentation.*
(2) *It is relatively consistent with* ZFC *that there is a structure having an infinite time computable quotient presentation, but no infinite time computable presentation.*

Let me briefly sketch some of the ideas appearing in the proof. In order to construct an infinite time computable presentation of a structure, given a computable quotient presentation, we'd like somehow to select a representative from each equivalence class, in a computably effective manner, and build a structure on these representatives. Under the set theoretic assumption $V = L$, we can attach to the L-least member of each equivalence class an escort real that is powerful enough to reveal that it is the L-least member of its class, and build a computable presentation out of these escorted pairs of reals. (In particular, the new presentation is not built out of mere representatives from the original class, since these reals may be too weak; they need the help of their escorts.) Thus, if $V = L$, then every structure with a computable quotient presentation has a computable presentation. On the other side of the independence, we prove statement 2 by the method of forcing. The structure $\langle \omega_1, < \rangle$ always has a computable quotient presentation built from reals coding well orders, but there are forcing extensions in which no infinite time computable set has size ω_1, on descriptive set theoretic grounds. In these extensions, therefore, $\langle \omega_1, < \rangle$ has a computable quotient presentation, but no computable presentation.

1.3. **Infinite time computable equivalence relation theory.** Recently, Sam Coskey and I have introduced the infinite time analogue of Borel equivalence relation theory and reductions. The idea of the classical Borel theory is to provide a structural analysis of the relative complexity of canonical equivalence relations on the reals (or more generally, Polish spaces) by comparing them under many-one Borel reducibility. Since Borel functions are all infinite time computable from their real parameters, it is a slight generalization of this theory to consider infinite time computable reductions. Thus, for any two equivalence relations E and F on $\mathbb{R}$, we say that E computably reduces to F, written $E \leq_c F$, if there is an infinite time computable function f (freely allowing real parameters) such that $x \mathrel{E} y \longleftrightarrow f(x) \mathrel{F} f(y)$. A slightly more generous notion of reduction is $E \leq_{sc} F$ if there is a semi-computable function $f : \mathbb{R} \to \mathbb{R}$, that is, a function whose graph is infinite time decidable (in a real parameter), such that $x \mathrel{E} y \longleftrightarrow f(x) \mathrel{F} f(y)$. An intriguing threshold phenomenon suggests that the distinction between computable reductions and semi-computable reductions is near a critical boundary.

To explain, let me first mention that there are an enormous number of natural equivalence relations on the reals to which this reduction theory applies, including all the Borel relations that have been studied in the classical theory. All the positive Borel reductions, of course, carry over to the infinite time context because all Borel functions are infinite time computable from their real parameters. Furthermore, many of the classical non-reductions in the Borel theory actually establish the lack of an infinite time computable reduction, because they often establish the lack of a measurable reduction, and all infinite time computable functions are measurable. In this way, the infinite time computable reduction theory is tightly interwoven into the classical Borel theory. Sample theorems include:

Theorem 7.

(1) *E_0 and $\equiv_{SET}$ do not computably reduce to $=$.*
(2) *$\equiv_{WO}$ and $\equiv_{SET}$ computably reduce to $\equiv_{HC}$.*
(3) *$\equiv_{HC}$ and $\equiv_{SET}$ do not computably reduce to $\equiv_{WO}$.*
(4) *$\equiv_{ck}$ and $\equiv_{WO}$ are computably bi-reducible.*

Interestingly, we know that it is consistent that the semi-computable reduction theory completely collapses.

Theorem 8. *If $V = L$, then every infinite time computable equivalence relation E on $\mathbb{R}$ semi-computably reduces to the equality relation.*

In this sense, under $V = L$ every computable relation is semi-computably smooth. The proof uses the ideas of Theorem 6, and as in that argument, the reduction functions are not selectors for the relation.
One should not construe this theorem to suggest that the semi-computable reduction relation is trivial, however, since under other set theoretic hypotheses inconsistent with $V = L$, such as a mild determinacy assumption, every semi-computable function is measurable. In this case the semi-computable degrees are definitely not collapsed in this way.

References

[DHS05] Vinay Deolalikar, Joel David Hamkins, and Ralf-Dieter Schindler. $P \neq \mathrm{NP} \cap \text{co-NP}$ for infinite time turing machines. *Journal of Logic and Computation*, 15(5):577–592, October 2005.

[Ham02] Joel David Hamkins. Infinite time turing machines. *Minds and Machines*, 12(4):521–539, 2002. (special issue devoted to hypercomputation).

[Ham04] Joel David Hamkins. Supertask computation. In Boris Piwinger Benedikt Löwe and Thoralf Räsch, editors, *Classical and New Paradigms of Computation and their Complexity Hierarchies*, volume 23 of *Trends in Logic*, pages 141–158. Kluwer Academic Publishers, 2004.

[Ham05] Joel David Hamkins. Infinitary computability with infinite time Turing machines. In Barry S. Cooper and Benedikt Löwe, editors, *New Computational Paradigms*, volume 3526 of *LNCS*, Amsterdam, June 8-12 2005. CiE, Springer-Verlag.

[HL00] Joel David Hamkins and Andy Lewis. Infinite time Turing machines. *J. Symbolic Logic*, 65(2):567–604, 2000.

[HL02] Joel David Hamkins and Andy Lewis. Post's problem for supertasks has both positive and negative solutions. *Archive for Mathematical Logic*, 41(6):507–523, 2002.

[HMSW07] J. D. Hamkins, R. Miller, D. Seabold, and S. Warner. Infinite time computable model theory. In S.B. Cooper, Benedikt Lfwe, and Andrea Sorbi, editors, *New Computational Paradigms: Changing Conceptions of What is Computable.* Springer, 2007.

[HS01] Joel David Hamkins and Daniel Seabold. Infinite time Turing machines with only one tape. *Mathematical Logic Quarterly*, 47(2):271–287, 2001.

[HW03] Joel David Hamkins and Philip Welch. $P^f \neq NP^f$ for almost all f. *Mathematical Logic Quarterly*, 49(5):536–540, 2003.

[Koe05] Peter Koepke. Turing computations on ordinals. *Bulletin of Symbolic Logic*, 11(3):377–397, September 2005.

[LÖ1] Benedikt Löwe. Revision sequences and computers with an infinite amount of time. *Logic Comput.*, 11(1):25–40, 2001.

[LM04] Giacomo Lenzi and Erich Monteleone. On fixpoint arithmetic and infinite time turing machines. *Information Processing Letters*, 91(3):121–128, 2004.

[Sch03] Ralf-Dieter Schindler. P $\neq$ NP for infinite time Turing machines. *Monatshefte für Mathematik*, 139(4):335–340, 2003.

[Wel] Philip Welch. On a question of Deolalikar, Hamkins and Schindler. available on the author's web page at http://www2.maths.bris.ac.uk/∼mapdw/dhs.ps.

[Wel99] Philip Welch. Friedman's trick: Minimality arguments in the infinite time Turing degrees. *in "Sets and Proofs", Proceedings ASL Logic Colloquium*, 258:425–436, 1999.

[Wel00a] Philip Welch. Eventually infinite time Turing machine degrees: Infinite time decidable reals. *Journal of Symbolic Logic*, 65(3):1193–1203, 2000.

[Wel00b] Philip Welch. The lengths of infinite time Turing machine computations. *Bulletin of the London Mathematical Society*, 32(2):129–136, 2000.

[Wel05] Philip Welch. The transfinite action of 1 tape Turing machines. In Barry S. Cooper and Benedikt Löwe, editors, *New Computational Paradigms*, volume 3526 of *LNCS*, Amsterdam, June 8-12 2005. CiE, Springer-Verlag.

Ordinalize!

Peter Koepke
University of Bonn
Sunday, 11:30 – 12:30

0. Introduction

Cantor's ordinals extend the standard natural numbers $\mathbb{N}$ into the transfinite:
$0, 1, 2, 3, \ldots, n, n+1, \ldots$
is continued by
$\omega, \omega+1, \ldots, \omega+n, \ldots, \omega+\omega = \omega\cdot 2, \omega+\omega+1 = \omega\cdot 2+1, \ldots$
$\omega\cdot\omega = \omega^2, \ldots, \omega^3, \ldots, \omega^\omega, \ldots, \omega^{\omega^2}, \ldots, \ldots$
$\omega^{\omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}}, \ldots, \alpha, \alpha+1, \ldots\ldots$
$\aleph_1, \aleph_1+1, \ldots, \aleph_2, \ldots, \aleph_3, \ldots, \aleph_\omega, \ldots\ldots\ldots$
Whereas natural numbers are either 0 or *successors*, by the axiom of infinity there are *limit ordinals* like $\omega, \omega+\omega, \ldots, \omega\cdot\omega, \ldots$. The induction and recursion laws for ordinals extend the corresponding laws for natural numbers by limit laws, where the letter λ is used to denote limit ordinals.

$$\begin{array}{l} A(0) \\ A(n) \rightarrow A(n+1) \\ \hline \forall n A(n) \end{array} \quad \text{is extended to} \quad \begin{array}{l} A(0) \\ A(\alpha) \rightarrow A(\alpha+1) \\ \forall \alpha < \lambda A(\alpha) \rightarrow A(\lambda) \\ \hline \forall \alpha A(\alpha) \end{array}$$

The recursion law

$$\begin{array}{l} F(0) = a_0 \\ F(n+1) = G(F(n), n) \end{array} \quad \text{is extended to} \quad \begin{array}{l} F(0) = a_0 \\ F(\alpha+1) = G(F(\alpha), \alpha) \\ F(\lambda) = G(F \restriction \alpha) \end{array}$$

One can now go through various mathematical theories based on natural numbers and try to extend them to ordinals (*Ordinalize!*). This contribution to BIWOC indicates how computability on the natural numbers may be ordinalized.

1. Ordinal recursive functions

Arithmetic becomes *ordinal arithmetic* with the operations

$$\begin{array}{l} \alpha + 0 = \alpha \\ \alpha + (\beta+1) = (\alpha+\beta)+1 \\ \alpha + \lambda = \bigcup_{\beta<\lambda} (\alpha+\beta) \end{array}$$

and

$$\begin{aligned}&\alpha \cdot 0 = 0\\ &\alpha \cdot (\beta+1) = (\alpha \cdot \beta) + \alpha\\ &\alpha \cdot \lambda = \bigcup_{\beta<\lambda} (\alpha \cdot \beta)\end{aligned}$$

Problem 1. *What can be defined in the structure* $(\mathrm{Ord}, <, +, \cdot, \dots, 0, 1)$*?*

Ordinal arithmetic suggests the following family of ordinal functions:

Definition 2. *The* ordinal recursive functions *form the smallest collection* $\mathcal{R}$ *of functions* $F : \mathrm{Ord}^i \to \mathrm{Ord}$ *such that*

- *the constant functions are in* $\mathcal{R}$
- *the projection functions are in* $\mathcal{R}$
- *the successor function* $\alpha \mapsto \alpha + 1$ *is in* $\mathcal{R}$
- *the indicator function* $I_< : \mathrm{Ord}^2 \to \{0, 1\}$, $I(\alpha, \beta) = 1$ *iff* $\alpha < \beta$
- $\mathcal{R}$ *is closed under functional composition*
- $\mathcal{R}$ *is closed under the following recursion schema, defining* F *from* $G_0, G_{\mathrm{succ}}, G_{\mathrm{lim}}$*:*

$$\begin{aligned}&F(0,\vec{p}) = G_0(\vec{p})\\ &F(\alpha+1,\vec{p}) = G_{\mathrm{succ}}(F(\alpha),\alpha,\vec{p})\\ &F(\lambda,\vec{p}) = \textstyle\bigcup_{\alpha<\lambda} G_{\mathrm{lim}}(F(\alpha),\alpha,\lambda)\end{aligned}$$

Example 3. *The following functions are ordinal recursive:*

- *ordinal arithmetic*
- *propositional logic (`true`∼ 1, `false`∼ 0):*

$$(A \wedge B)(\vec{x}) = A(\vec{x}) \cdot B(\vec{x}), \neg A(\vec{x}) = I_<(A(\vec{x}), 1)$$

- *bounded quantification:*

$$\exists \nu < \alpha A(\nu, \vec{x}) = \bigcup_{\nu<\alpha} A(\nu, \vec{x})$$

- $\max(\alpha, \beta) = \alpha \cdot I_<(\beta, \alpha) + \beta \cdot \neg I_<(\beta, \alpha)$
- *the indicator function for equality* $I_=(\alpha, \beta) = (\neg I_<(\alpha, \beta)) \wedge (\neg I_<(\beta, \alpha))$
- *if* $F : \mathrm{Ord} \to \mathrm{Ord}$ *is ordinal recursive and strictly monotone and other conditions hold then* F^{-1} *is ordinal recursive.*
- *the sum* $S(\alpha)$ *("*$= 2 + 4 + \ldots + \nu \cdot 2 + \ldots$ *for* $\nu < \alpha$*") of* even *ordinals by the recursion*

$$S(0) = 0, S(\alpha + 1) = S(\alpha) + \alpha \cdot 2, S(\lambda) = \bigcup_{\alpha<\lambda} S(\alpha)$$

- *define the* Gödel pairing $\langle .,. \rangle : \mathrm{Ord}^2 \leftrightarrow \mathrm{Ord}$ *by*

$$\langle \alpha, \beta \rangle = S(\max(\alpha, \beta)) + I_<(\alpha, \beta) \cdot \alpha + \neg I_<(\alpha, \beta) \cdot (\alpha + \beta)$$

- *the projections* $\langle \alpha, \beta \rangle \mapsto \alpha$ *and* $\langle \alpha, \beta \rangle \mapsto \beta$ *are ordinal recursive*
- *via* Gödel *pairing and unpairing, ordinals may be seen as finite sequences of ordinals, or as sequences of* symbols

2. An ordinal language

Let the language L_T be appropriate for first-order structures of the type

$$(\alpha, <, G, R)$$

where the Gödel pairing function G is viewed as a ternary *relation* on α and R is a unary relation on α. So the language consists of

- terms v_n and constants c_ξ for $\xi \in \mathrm{Ord}$; c_ξ will be interpreted as ξ;

– atomic formulas $t_1 \equiv t_2$, $t_1 < t_2$, $\dot{G}(t_1, t_2, t_3)$ and $\dot{R}(t_1)$;
– formulas $\neg\varphi$, $(\varphi \vee \psi)$, $\exists v_n < t\varphi$.

Note that all formulas of L_T are bounded. We assume an ordinal computable *Gödelization* such that for $\zeta < \xi$:

$$\varphi\frac{c_\zeta}{v_n} < (\exists v_n < c_\xi \varphi).$$

Define the satisfaction relation $(\mathrm{Ord}, <, G, R) \models \varphi$ for sentences φ as usual. Since φ is bounded,

$$(\mathrm{Ord}, <, G, R) \models \varphi \text{ iff } (\varphi, <, G, R) \models \varphi.$$

Definition 4. *Define the* bounded truth predicate $T \subseteq \mathrm{Ord}$ *by*

$$T(\alpha) \quad \textit{iff} \quad \alpha \textit{ is a bounded } L_T\textit{-sentence and } (\alpha, <, G, T \cap \alpha) \vDash \alpha.$$

In short

$$T(\alpha) \quad \textit{iff} \quad (\alpha, T \cap \alpha) \vDash \alpha.$$

Theorem 5. *The truth predicate T is ordinal recursive.*

Proof. The characteristic function χ_T can be defined by

$$\chi_T(\alpha) = \begin{cases} 1 \text{ iff } \exists \nu < \alpha H(\alpha, \nu, \chi_T(\nu)) = 1 \\ 0 \text{ else} \end{cases}$$

with

$$\begin{array}{lll} H(\alpha, \nu, \chi) = 1 & \text{iff} & \alpha \text{ is an } L_T\text{-sentence and} \\ & & \exists \xi, \zeta < \alpha(\alpha = c_\xi \equiv c_\zeta \wedge \xi = \zeta) \\ & \text{or} & \exists \xi, \zeta < \alpha(\alpha = c_\xi < c_\zeta \wedge \xi < \zeta) \\ & \text{or} & \exists \xi, \zeta, \eta < \alpha(\alpha = \dot{G}(c_\xi, c_\zeta, c_\eta) \wedge \eta = G(\xi, \zeta)) \\ & \text{or} & \exists \xi < \alpha(\alpha = \dot{R}(c_\xi) \wedge \nu = \xi \wedge \chi = 1) \\ & \text{or} & \exists \varphi < \alpha(\alpha = \neg\varphi \wedge \nu = \varphi \wedge \chi = 0) \\ & \text{or} & \exists \varphi, \psi < \alpha(\alpha = (\varphi \vee \psi) \wedge (\nu = \varphi \vee \nu = \psi) \wedge \chi = 1) \\ & \text{or} & \exists n < \omega \exists \xi < \alpha \exists \varphi < \alpha(\alpha = \exists v_n < c_\xi \varphi \wedge \exists \zeta < \xi \nu = \varphi\frac{c_\zeta}{v_n} \wedge \chi = 1). \end{array}$$

$$\chi_T(\alpha) = \begin{cases} 1 \text{ iff } \exists \nu < \alpha H(\alpha, \nu, \chi_T(\nu)) = 1 \\ 0 \text{ else} \end{cases}$$

is equivalent to

$$\chi_T(\alpha) = \bigcup_{\nu < \alpha} H(\alpha, \nu, \chi_T(\nu))$$

and thus χ_T is ordinal recursive. □

3. Constructibility

Definition 6. *The* constructible model L *was defined by* Gödel*:*

$$\begin{array}{rcl} L_0 & = & \emptyset \\ L_{\alpha+1} & = & \mathrm{Def}(L_\alpha) = \textit{the set of first-order definable subsets of } (L_\alpha, \in) \\ L_\lambda & = & \bigcup_{\alpha < \lambda} L_\alpha \\ L & = & \bigcup_{\alpha \in \mathrm{Ord}} L_\alpha \end{array}$$

L is the $\subseteq$-minimal inner model of the Zermelo-Fraenkel axioms ZFC. The bounded truth predicate T is just as strong as the constructible model:

Theorem 7. *For ordinals μ and α define "sections" of the truth predicate by*

$$X(\mu, \alpha) = \{\beta < \mu | T(G(\alpha, \beta))\}.$$

Set $\mathcal{S} = \{X(\mu, \alpha) | \mu, \alpha \in \mathrm{Ord}\}$. Then $\mathcal{S} = \{x \subseteq \mathrm{Ord} \,| x \in L\}$.

Proof. (Sketch for $\supseteq$) Show that $(\mathrm{Ord}, \mathcal{S}, <, =, \in, G)$ satisfies a natural theory of sets of ordinals; mathematics can be done in $(\mathrm{Ord}, \mathcal{S}, <, =, \in, G)$; define a version of GÖDEL's L inside $(\mathrm{Ord}, \mathcal{S}, <, =, \in, G)$; thus every constructible set of ordinals is an element of $\mathcal{S}$. □

Thus ordinal recursive functions lead to an *ordinal recursion theory* where ordinal recursive *sets* are the constructible sets.

Problem 8. *Is there a reasonable recursion theory for the ordinal recursive* classes *with respect to ordinal recursive reducibility? Is that reducibility equivalent to Δ^1_1 reducibility over L?*

4. Ordinal programming languages

The essential recursion

$$\chi_T(\alpha) = \begin{cases} 1 \text{ iff } \exists \nu < \alpha H(\alpha, \nu, \chi_T(\nu)) = 1 \\ 0 \text{ else} \end{cases}$$

can be described in a recursive pseudo language like

```
define T(alpha) by
   input alpha
   let nu=0
   while nu<alpha
      if H(alpha,nu,T(nu))=1 return 1
      nu=nu+1
   return 0
```

Problem 9. *Can one generalize other programming languages or language constructs to the ordinals?*

5. Ordinal machines

5.1. **Ordinal stack machines.** Recursive programs on ordinals as above can be interpreted on machines with finite descending *ordinal stacks*

$$\alpha_0(t) > \alpha_1(t) > \ldots > \alpha_{l(t)-1}(t).$$

The machines works in ordinal time t with the following behaviour at limit ordinals λ:

– if $(\alpha_0(t), \ldots, \alpha_{l-1}(t))$ is eventually konstant before time λ then set

$$(\alpha_0(\lambda), \ldots, \alpha_{l-1}(\lambda)) = (\alpha_0(t), \ldots, \alpha_{l-1}(t))$$

for sufficiently high $t < \lambda$. Also let l be maximal with that property.

– if $\liminf_{t \to \lambda} \alpha_l(t)$ is defined, set $l(\lambda) = l + 1$ and $\alpha_l(\lambda) = \liminf_{t \to \lambda} \alpha_l(t)$

– if $\liminf_{t \to \lambda} \alpha_l(t)$ is undefined, set $l(\lambda) = l$

5.2. **Ordinal Turing machines.**

– use standard TURING programs

– employ lim inf-rules as limit rules for tape contents and state

– The truth predicate T can be calculated by an ordinal TURING machine, writing T successively on one of the tapes.

– Thus: a set of ordinals is ordinal TURING computable iff it is constructible.

5.3. **Ordinal register machines.**

– use standard register programs, i.e., `goto` programs
– employ lim inf-rules for register contents and state
– An ordinal stack can be simulated by an ordinal register machine.
– Thus: a set of ordinals is ordinal register computable iff it is constructible.

5.4. **Nondeterministic computations.** A class $\mathcal{C}$ of sets of ordinals is *nondeterministically ordinal computable* if there is an ordinal Turing machine $\mathcal{M}$ with ordinal parameters such that for $x \subseteq \mathrm{Ord}$

$$x \in \mathcal{C} \text{ iff } \exists y \mathcal{M} \text{ accepts } (x, y)$$

Problem 10. *What is the class*

$$N = \{x \subseteq \mathrm{Ord} \,|\, \{x\} \textit{ is non-deterministically ordinal computable}\}?$$

6. An application: fine structure for the constructible model

Apart from leading to satisfying models of infinitary computation, ordinal computability also starts to have applications in other fields. We indicate how the Jensen fine structure of the constructible hierarchy may be reconstructed within ordinal computability. We base our approach on Silver machines.

Definition 11. *Consider* $M = (\mathrm{Ord}, <, M)$*,* $M : \mathrm{Ord}^{<\omega} \rightharpoonup \mathrm{Ord}$*. For* $\alpha \in \mathrm{Ord}$ *let*

$$M^\alpha = (\alpha, <, M \cap \alpha^{<\omega});$$

for a set $X \subseteq \alpha$ *let* $M^\alpha[X]$ *be the substructure of* M^α *generated by* X*.* M *is a* Silver machine *iff it satisfies*

– Condensation: *for* $\alpha \in \mathrm{Ord}$ *and* $X \subseteq \alpha$ *there is a unique* β *such that* $M^\beta \cong M^\alpha[X]$*;*
– Finiteness property: *for* $\alpha \in \mathrm{Ord}$ *there is a finite set* $z \subseteq \alpha$ *such that for all* $X \subseteq \alpha + 1$
$$M^{\alpha+1}[X] \subseteq M^\alpha[(X \cap \alpha) \cup z] \cup \{\alpha\};$$
– Collapsing property: *if the limit ordinal* β *is singular in* L *then there is* $\alpha < \beta$ *and a finite set* $p \subseteq \mathrm{Ord}$ *such that* $M[\alpha \cup p] \cap \beta$ *is cofinal in* β*.*

Jack Silver defined Silver machines within the constructible model L and used them to give simple proofs of the combinatorial principles $\Box$ and Morass. We can naturally define a Silver machine from the bounded truth predicate T.

Definition 12. *Consider the structure* $(\mathrm{Ord}, <, T)$*. Define a* Skolem *function by*

$$h(\alpha) = \begin{cases} \beta, \textit{ if } \alpha = \exists v_n < c_\xi \varphi \textit{ and } \beta \textit{ is minimal s. th. } (\alpha, <, G, T) \models \varphi\frac{c_\beta}{v_n} \\ 0, \textit{ else} \end{cases}$$

Let G_1, G_2 *be the inverses of the* Gödel *pairing function. Code* h, G_1, G_2 *into a machine function* M *by*

$$M(0, \alpha) = h(\alpha), M(1, \alpha) = G_1(\alpha), M(2, \alpha) = G_2(\alpha).$$

Theorem 13. $M = (\mathrm{Ord}, <, M)$ *as defined in the previous definition is a* Silver *machine.*

Proof. (Sketch)
Condensation: For $\alpha \in \mathrm{Ord}$ and $X \subseteq \alpha$ there is a unique β such that $M^\beta \cong M^\alpha[X]$.
Proof by induction on α: Let $Y = M^{\alpha+1}[X]$. By inductive assumption: $\pi : Y \cap \alpha \cong M^\beta$. If $\alpha \notin Y$, then $Y = Y \cap \alpha \cong M^\beta$. If $\alpha \in Y$, then $Y \cong M^{\beta+1}$; for this one mainly has to show that

$$\pi(h(\alpha)) = h(\beta).$$

Finiteness: Observe that $M^{\alpha+1}[X] \subseteq M^{\alpha}[(X \cap \alpha) \cup \{h(\alpha), G_1(\alpha), G_2(\alpha)\}] \cup \{\alpha\}$ and so $z = \{h(\alpha), G_1(\alpha), G_2(\alpha)\}$ may be taken as the desired finite set.
Collapsing: If the limit ordinal β is singular in L then there is $\alpha < \beta$ and a finite set $p \subseteq \mathrm{Ord}$ such that $M[\alpha \cup p] \cap \beta$ is cofinal in β.
This holds because every constructible set of ordinals including a singularizing cofinal set for β can be decoded from T with the help of h. $\square$

Problem 14. *Can one construe fine structural constructions like the definition of $\square$-sequences as computations of ordinal machines?*

The Complexity of Quickly ORM-Decidable Sets

David Linetsky
City University of New York Graduate Center
Tuesday, 12:00 – 12:30

The speaker will discuss some recent results, proved in collaboration with Joel Hamkins and Russell Miller, addressing the time required by Ordinal Register Machines (ORM's) to decide sets of natural numbers. The main theorem to be presented states that the sets of natural numbers that can be decided by an ORM in time uniformly less than ω^ω are exactly the arithmetic sets.

Theorem 1. *The subsets of ω that are ORM-decidable (with ordinal parameters!) in time uniformly less than ω^ω are exactly the arithmetic sets.*

We contrast this with the analogous result, due to Hamkins and Lewis ([2]) concerning Infinite Time Turing Machines (ITTM's), which states that the truth of any arithmetic statement is ITTM-decidable in time less than ω^2. This result highlights an interesting difference between ORM's and Infinite Time Turing Machines (ITTM's), namely, that ITTM's can decide any arithmetic set in time less than ω^2, while ORM's require times arbitrarily large below ω^ω. This is due to the fact that ITTM's are able to make use of infinite tapes on which they can write out an entire truth predicate for sentences of lower complexity that can in turn be used to quickly decide sentences of higher complexity. ORM's, on the other hand, are unable to take advantage of this strategy as their memory is limited to a finite number of ordinals.
A refinement of the main theorem will also be presented which gives a level-by-level characterization of the arithmetic sets. In particular, it will be shown that:

Theorem 2. *$A \subseteq \omega$ is ORM-decidable in time less than ω^{n+1} iff $A \in \Delta^0_{3n+1}$.*

This is accomplished by representing ordinals below $\omega^\omega \cdot 2$ as finite sequences of natural numbers (viz. the coefficients found in their Cantor normal form) and analyzing the complexity of the relation $R_n(\mathcal{C}, \mathcal{C}', P)$, which holds iff the ORM configuration coded by $\mathcal{C} \in \omega$ leads to the configuration coded by $\mathcal{C}' \in \omega$, under the operation of program P, in exactly ω^n steps. The coding of ordinals as finite sequences of natural numbers allows these relations to be expressed using only quantifiers ranging over natural numbers. Indeed, this is expressed precisely in the following lemma.

Lemma 3. *Given $\mathcal{C}, \mathcal{C}', P, n \in \omega$ and relation R_n as above, the sentence $R_n(\mathcal{C}, \mathcal{C}', P)$ is Π^0_{3n}.*

Furthermore, it will be shown that all of the above results concerning ORM's hold even when allowing arbitrary ordinal parameters. This is accomplished by by the following lemma, which shows that in ORM computations which halt in time less than $\omega^\omega$and output a result less than ω^ω, any parameters greater than ω^ωcan be replaced with ω^ωitself without in anyway affecting the computation.

Lemma 4. *For any ORM program P, any finite sequence $\vec{\beta} = \beta_0, \beta_1, \ldots, \beta_n \in \omega^\omega$, and any $\beta > \omega^\omega$, if $P(\vec{\beta}, \beta) \downarrow < \omega^\omega$ in time $\gamma < \omega^\omega$, then $P(\vec{\beta}, \omega^\omega) \downarrow = P(\vec{\beta}, \beta)$ in time γ.*

Moreover, we show that the uniformity in time found in the main theorem is in fact necessary, i.e., that there are non-arithmetic sets that are ORM-decidable in time less that ω^ω, but not uniformly so. In particular:

Theorem 5. *The set $\emptyset^{(\omega)}$ is ORM-decidable in time less than ω^ω, but not uniformly so.*

Finally we end with conjecture about what can expected beyond ω^ω.

Conjecture/Theorem 6. *The sets that are ORM-decidable in time less that ω_1^{CK} are exactly the hyperarithmetic sets.*

References

[1] Peter Koepke. "Ordinals, Computations, and Models of Set Theory: A Tutorial at Days in Logic, Coimbra, Portugal." Tutorial Material. `http://www.mat.uc.pt/~kahle/dl06/koepke.pdf`; accessed January, 2006.

[2] Joel David Hamkins and Andy Lewis. "Infinite Time Turing Machines." *J. Symbolic Logic*, 65(2):567-604, 2000.

Post's Problem for Ordinal Register Machines

Russell Miller[2]
(joint work with Joel D. Hamkins)
City University of New York
Thursday, 11:00 – 12:00

The original version of Post's Problem applied to finite-time Turing machines. It asked whether there exists a computably enumerable set A which is neither computable nor complete. That is, it required $\emptyset <_T A <_T \emptyset'$, where $\emptyset'$ is the jump of the empty set, or equivalently the Halting Problem for finite-time Turing machines. Post's Program for solving this problem was to discover a nonvacuous property of c.e. sets, expressible using only the containment relation $\subseteq$, which would guarantee that A was incomplete and noncomputable.
Post's Program led to a substantial amount of useful research, which yielded the notions of simple, hypersimple, and hyperhypersimple sets, all of which were properties which Post originally hoped would fulfill his program. In fact, none of these properties implies incompleteness, and Post did not live to see the solution of the problem that bears his name. Post's Program was completed by Harrington and Soare [3] in 1991, but his Problem was solved much earlier, in 1956 and 1957, with the invention of the finite injury priority method (independently) by Friedberg [1] and Muchnik [4]. A good description of this method appears in section VII.2 of [5].

[2]The research of Russell Miller and Joel D. Hamkins has been supported in part by grants from the Research Foundation of CUNY.

In this talk we will ask the analogous question for sets of ordinals under computation by ordinal register machines, or ORM's. Of course, some of the concepts must be adapted to the new setting. For instance, while ORM programs are finite and thus can still be coded by natural numbers, the inputs to these machines are arbitrary ordinals. So, as in the case of infinite-time Turing machines (which were studied by Hamkins and Lewis; see [2]), we have both a jump:

$$\emptyset^{\triangledown} = \{p : \phi_p(0) \downarrow\} \subset \omega$$

in which we diagonalize to build a noncomputable set; and also a halting problem:

$$\emptyset^{\blacktriangledown} = \{\langle p, \alpha\rangle : \phi_p(\alpha) \downarrow\} \subset \omega \times \mathrm{ON}$$

asking the general question of whether a given program halts on a given input. Clearly $\emptyset^{\triangledown}$ is computable from $\emptyset^{\blacktriangledown}$, and in finite-time computability the jump $\emptyset'$ and the halting problem are computably isomorphic, but in this context they are very much distinct.
We consider semidecidable sets, or equivalently, ORM-enumerable sets: sets of ordinals which form the domain (equivalently, the range) of the function computed by some ordinal register machine. (In general, neither the function computed by an ORM, nor the class of ordinals on which it halts, nor the class of ordinals which it outputs need be a set, of course.)
Our version of Post's Problem then asks whether there exist noncomputable incomplete semidecidable sets of ordinals (where *incomplete* means that these sets should not be able to compute $\emptyset^{\triangledown}$). As was the case in [2] for infinite-time Turing machines, these specifications allow both a positive and a negative solution. If we consider only subsets of ω, then there is no noncomputable incomplete semidecidable set. However, if we consider sets of ordinals (which we may interpret to include subsets of $\omega \times \mathrm{ON}$), then there is such a set.

Theorem 1. *There exist incomparable semidecidable sets A and B of ordinals. It follows that $\emptyset <_{ORM} A <_{ORM} \emptyset^{\triangledown} (<_{ORM} \emptyset^{\blacktriangledown})$, and similarly for B. However, no semidecidable subset $C \subseteq \omega$ satisfies $\emptyset <_{ORM} C <_{ORM} \emptyset^{\triangledown}$.*

The proof of the negative solution, for subsets of ω, runs along the same lines as that in [2]. The proof of the positive solution is more involved, and also dissimilar from the corresponding proof for infinite-time Turing machines by Hamkins and Lewis. Therefore, the bulk of the talk will be devoted to this proof.
For the positive solution, we use the Friedberg-Muchnik approach, fixing a witness element x for each e such that $x \in B$ iff $\phi_e^A(x)$ halts and equals 0, thereby guaranteeing that ϕ_e^A does not compute B. Doing so for all e (and also with A and B interchanged) will make A and B Turing-incomparable, yet semidecidable, since our procedure will be computable by an ORM and will only add elements to A and B, without ever removing them. A standard finite-injury priority construction is used to make sure that once $\phi_e^A(x)$ has converged to 0, we do not make any changes to the portion of A used by this oracle computation, so that the disagreement between ϕ_e^A and B is preserved thereafter, unless a higher-priority requirement acts to change it.
The difficulty is that, with only finitely many ordinal registers, it is not clear how we should remember the values of the witness elements for all the different requirements, or keep track of which have entered A or B already, or which requirements have been satisfied or injured at various stages of the computation. Our approach is that, whenever we need to know the answers to such a question (of the specific form: "did the ordinal x enter A, or B, at stage $\sigma + 1$?"), we go back and simulate the entire computation up to that stage, and thus determine the answer. Of course, in the computation up to that stage, we had often asked similar questions

about earlier stages. Therefore, our process for answering such a question is in the form of a routine that is allowed to call itself. For this purpose, *stack machines*, a special form of ORM developed by Koepke and Siders, will be essential. In a stack machine, a register can hold not just a single ordinal, but a decreasing (hence finite) sequence of ordinals, always allowing us to pop the smallest ordinal off the top of the stack, or to push a new smaller ordinal onto the stack. Since our routine always asks about smaller and smaller stages of the computation, a stack machine adapts precisely to our purposes. Our proof of the Friedberg-Muchnik result for ORM's therefore consists simply of a routine answering the question "did x enter A at stage $\sigma + 1$?" for arbitrary x and σ, and a similar routine for B, which are allowed to call themselves or each other, but always asking about smaller stages than before. (This will require a secondary routine, to which the same analysis applies, which checks whether x was chosen as a witness element at stage $\sigma + 1$.) Now A is semidecidable, being the domain of the ORM-computable function which asks this question about its input x for ever-increasing ordinals σ and halts whenever it receives a positive answer. (All these results hold symmetrically for B as well, of course.) This much would make A computable from $\emptyset^{\blacktriangledown}$. We will go further, however, and answer Post's Problem for $\emptyset^{\triangledown}$ as well. For this, the key is to ensure that every ordinal x which ever acts as a witness element will be a writable ordinal, i.e. $x = \phi_p(0)$ for some p. Hence, using a $\emptyset^{\triangledown}$-oracle, we can compute whether $x \in A$ for arbitrary x, as follows. First we run the program for x stages and see whether x is chosen as a witness element by stage x. If not, then it never will be so chosen, so it never entered A. If so, then x must be writable, so we find a program which writes it on input 0: just ask the oracle (for each $p < \omega$) whether $\phi_p(0)$ halts, and if so, check whether it outputs x. Then we find a program q which, on input 0, runs program p to write x and then halts iff that x ever enters A. The $\emptyset^{\triangledown}$-oracle will tell us whether ϕ_q ever halts on input 0, and that answers the question.

References

[1] R.M. Friedberg; Two recursively enumerable sets of incomparable degrees of unsolvability, *Proc. Nat. Acad. Sci. (USA)* **43** (1957), 236-238.

[2] J.D. Hamkins & A. Lewis; Post's problem for supertasks has both positive and negative solutions, *Arch. Math. Logic* **41** (2002), 507-523.

[3] L. Harrington & R. I. Soare; Post's Program and Incomplete Recursively Enumerable Sets, *Proc. Nat. Acad. Sci. (USA)* **88** (1991), 10242-10246.

[4] A.A. Muchnik; On the Unsolvability of the Problem of Reducibility in the Theory of Algorithms, *Dokl. Akad. Nauk SSSR*, N.S. **109** (1956), pp. 194-197 (Russian).

[5] R. I. Soare; *Recursively Enumerable Sets and Degrees* (New York: Springer-Verlag, 1987).

The P vs. NP problem for infinite time Turing machines

Ralf Schindler
Universität Münster
Tuesday, 9:30 – 10:30

Abstract. We state different versions of the P vs. NP–problem for infinite time Turing–machines.

The running time of a classical Turing–machine T is a function $f : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, for all inputs w of length $\leq n$, T stops after at most $f(n)$ steps. In the case of infinite time Turing–machines, all typical inputs have the same length, namely ω. We may therefore construct the running time of such a machine as a constant ordinal number. We arrive at:

Definition 1. *Let $A \subset {}^{\omega}2$. We say that A is decidable in polynomial time, or $A \in P$, if there are a Turing–machine T and some $m < \omega$ such that*
- (a) *T decides A (i.e., $x \in A$ iff T accepts x), and*
- (b) *T halts on all inputs in fewer then ω^m many steps.*

Definition 2. *Let $A \subset {}^{\omega}2$, and let $\alpha \leq \omega_1 + 1$. We say that A is in P_α if there is a Turing–machine T and some $\beta < \alpha$ such that*
- (a) *T decides A (i.e., $x \in A$ iff T accepts x), and*
- (b) *T halts on all inputs in fewer than β many steps.*

Of course $P = P_{\omega^\omega}$. Moreover, P_{ω_1+1} is just the class of all $A \subset {}^{\omega}2$ which are decided by some Turing–machine.

Lemma 3. *Let $A \subset {}^{\omega}2$. Then $A \in P_{\omega^2}$ if and only if A is an arithmetic set.*

Lemma 4. *Let $A \subset {}^{\omega}2$. Then $A \in P_{\omega+2}$ if and only if A is a lightface G_δ–set.*

Lemma 5. *Let $A \subset {}^{\omega}2$. Then $A \in P_{\omega_1^{CK}}$ if and only if A is a hyperarithmetic set. If $A \in P_{\omega_1}$ then A is a Borel–set.*

Classically a problem A is in NP iff there is a Turing–machine T such that for all w, $w \in A$ iff there is some v such that T accepts $w \oplus v$, and the running time of T on an input of the form $w \oplus v$ is polynomial in the length of w.

Definition 6. *Let $A \subset {}^{\omega}2$. We say that A is verifiable in polynomial time, or $A \in NP$, if there is a Turing–machine T and some $m < \omega$ such that*
- (a) *$x \in A$ if and only if $(\exists y T$ accepts $x \oplus y)$, and*
- (b) *T halts on all inputs in fewer than ω^m many steps.*

Definition 7. *Let $A \subset {}^{\omega}2$, and let $\alpha \leq \omega_1 + 1$. We say that A is in NP_α, if there are a Turing–machine T and some $\beta < \alpha$ such that*
- (a) *$x \in A$ if and only if $(\exists y T$ accepts $x \oplus y)$, and*
- (b) *T halts on all inputs in fewer than β many steps.*

Again, $NP = NP_{\omega^\omega}$. NP_α is the class of all projections of sets in P_α.
It is not hard to show the following.

Theorem 8. *$NP_{\omega+2} \setminus P_{\omega_1} \neq \emptyset$. In particular, $P \neq NP$.*

A second version of the P vs. NP–problem counts an input w as having length ω_1^w. (This had been a suggestion of Philip Welch.)

Definition 9. *Let $A \subset {}^{\omega}2$. We say that $A \in P^+$ if there is a Turing–machine T such that*
- (a) *$x \in A$ if and only if T accepts x, and*
- (b) *T halts on all inputs x in fewer than ω_1^x many steps.*

Definition 10. *Let $A \subset {}^{\omega}2$. We say that $A \in NP^+$ if there is a Turing–machine T such that*
- (a) *$x \in A$ if and only if $(\exists y T$ accepts $x \oplus y)$, and*
- (b) *T halts on all inputs $x \oplus y$ in fewer than ω_1^x many steps.*

Again we'll have that $P \neq NP$.

Theorem 11. *$P^+ = P_{\omega_1^{CK}} = \Delta^1_1$.*

Corollary 12. *$NP^+ \setminus P^+ \neq \emptyset$.*

The results so far appeared in [2]. There is yet another interesting version.

Definition 13. *Let $A \subset {}^{\omega}2$. We say that $A \in P^{++}$ if there is a Turing–machine T such that*

(a) *$x \in A$ if and only if T accepts x, and*
(b) *T halts on all inputs x in fewer than $\omega_1^x + \omega + 1$ many steps.*

Theorem 14. (Hamkins, Welch) $P^{++} \neq NP^{++}$.

The paper [1] studies the pointclass $NP \cap co - NP$ and its relation to P and to NP.

References

[1] **V. Deolalikar, J. Hamkins, R. Schindler**, *$P \neq NP$ for infinite time Turing–machines*, Journal of Logic and Computation, 15 (2005), pp. 577 – 592

[2] **R. Schindler**, *$P \neq NP$ for infinite time Turing–machines*, Monatshefte für Mathematik, 139 (2003), pp. 335 – 340

An Introduction to Infinite time computable model theory

Daniel Seabold and Steve Warner
Hofstra University

Wednesday, 10:45 – 11:30 part I
and 11:45 – 12:30 part II

0. Introduction

We report on joint work with Joel Hamkins and Russell Miller [HMSW07]. Infinite time Turing machines provide a natural context for studying the computability of functions that have real numbers as inputs and outputs. By carrying out the program of computable model theory using these machines, we are therefore able to consider uncountable structures and theories built from reals. Moving computability theory from the finite-time to the infinite-time context vastly increases the computation power but complicates the domain of discourse. Our results therefore include both strengthenings of the classical (finite-time) theory and striking departures from it. Several natural statements are independent of ZFC.
Our first concern is to ensure that the machine can recognize and parse any code for a formula in the language. We therefore restrict our attention to *computably presented* languages: each function, relation, and constant symbol s must be assigned a real code $\ulcorner s \urcorner$ in such a way that the set of codes is decidable and we can uniformly determine from any code $\ulcorner s \urcorner$ what sort of symbol s is and what its arity is. All other symbols, including variable symbols, are coded by integers. A model $\mathcal{A} = (A, \ldots)$ in a computably presented language is *computable* if the underlying set $A \subseteq \mathbb{R}$ is decidable and each function, relation, and constant is uniformly computable from its input and the code for its symbol. If a model is computable then its atomic diagram is decidable. The model is *decidable* if its full elementary diagram is decidable. A structure is *computably presentable* if it isomorphic to a computable model. Both the structure $(\omega_1^L, <)$ and the real line with its standard operations are computable presentable.
Before considering the theory in detail indeed we pause to present a few key ideas which drive many of the results.

(1) The full power of the machines is on display when the elements of the structure and language are coded in a "searchable" set such as $\mathbb{N}$. For example, a computable model whose underlying set is $\mathbb{N}$ is also decidable since an infinite time Turing machine can inductively carry out the Tarskian definition of truth by searching for witnesses to existential statements. Similarly, a theory coded in $\mathbb{N}$ with a decidable set of axioms will itself be decidable.
(2) Even when the structure or language is not coded in a searchable set, positive results may be obtained for the constructible universe by relying on a decidable set of unique codes for elements of L_{ω_1}. We use these L-codes, which generalize the "lost melody reals" of [HL00], to nicely enumerate languages when proving the computable completeness theorem in L.
(3) Several negative results follow from the existence of a function on $\mathbb{R}$ with a decidable graph that is not computable. It implies, for example, that a decidable model need not be computable.
(4) Other negative results—such as the consistent failure of the computable completeness theorem—rely on the fact from descriptive set theory that no decidable set can have cardinality strictly between ω_1^L and the continuum.
(5) When we say a model is decidable, we mean that a question about the model can be answered given codes for the formula and parameters. Through these codes, we can effectively smuggle in oracles to consult during the decision process. Decidability questions are thus highly sensitive to the presentation of the model and its language.

1. Examples from L-codes

The construction of unique codes for elements of L_{ω_1} generalizes a technique developed by Hamkins and Lewis to prove the so-called Lost Melody Theorem. Suppose that a is an element of L_{ω_1}. Let β be least such that $a \in L_\beta$ and β is countable in $L_{\beta+1}$. It follows that L_β is countable in $L_{\beta+1}$, so there is some L-least real c coding a relation E such that $\langle N, E\rangle \cong \langle L_\beta, \in\rangle$. The set a is represented by some natural number n with respect to E. The *L-code* of a is the pair $\langle n, c\rangle$. It follows that the set of L-codes is decidable. We can also computably decide the relation $\in^*$ induced on the codes by $\in$. As an immediate application, we obtain computable presentations for the structures $\langle \omega_1^L, \in\rangle$ and $\langle L_{\omega_1{}^L}, \in\rangle$ by identifying each element of the structure with its L-code.
Recall that a real is *writable* if it is the output of an infinite time Turing machine on input zero. Hamkins and Lewis prove that if c is the L-code of an ordinal that exceeds the lengths of all computations on zero input, then c is not writable but $\{c\}$ is decidable.[3] The existence of such a real produces several basic counterexamples for infinite time computability theory.

- The constant function $f(x) = c$ has a decidable graph but is not computable.
- The structures $\langle \mathbb{R}, c\rangle$ and $\langle \mathbb{R}, f\rangle$ are decidable models which are not computable.
- The function with domain $\{c\}$ mapping c to 0 is computable but its inverse is not.

2. Computable Quotient Presentations

Although a particular decidable presentation of a model need not be computable, we may still ask whether every decidable model has *some* computable presentation.

[3]Such a real is called a "lost melody real", since it is like a melody which you cannot hum yourself, but which you can recognize when hummed by someone else.

This is true if we relax our notion of a presentation to include quotients. Suppose that $\mathcal{A}$ is an infinite time decidable structure. Augment the language by adding a constant symbol for every element of $\mathcal{A}$, and let $\mathcal{A}^*$ be the structure whose underlying set is the terms of this expanded language, with all function and relation symbols having their obvious interpretations. The functions are computable and the relations are decidable. Define $t_1 \equiv t_2$ if $\mathcal{A} \models t_1 = t_2$. This equivalence relation is computable because $\mathcal{A}$ is decidable. Since $\mathcal{A}^*/\equiv$ is isomorphic to $\mathcal{A}$, we have constructed an infinite time *computable quotient presentation* for $\mathcal{A}$.

Theorem 1. *Every decidable model has a computable quotient presentation.*

We are thus naturally led to ask:

Question 2. *Does every structure with an infinite time computable quotient presentation have an infinite time computable presentation?*

As a test case we consider the structure $\langle \omega_1, < \rangle$. Although this structure has a computable quotient presentation, namely $\langle \text{WO}, <, \equiv \rangle$, the existence of a computable presentation hinges on set-theoretic assumptions. If $\omega_1 = \omega_1^L$, then the structure can be computably presented using L-codes. If $\omega_1 > \omega_1^L$ and CH fails, then there are no decidable sets of size ω_1, and hence no computable presentation for this structure exists. The latter result is a consequence of the Mansfield-Solovay Theorem, which states that every Σ^1_2 set of reals not contained in L contains a perfect subset. Since every decidable set of reals is Σ^1_2, each such set has size at most ω_1^L or else size continuum. Thus if $\omega_1 > \omega_1^L$ and CH fails, there are no decidable sets of cardinality ω_1.
This independence result can be generalized.

Theorem 3. *The statement "every structure with a computable quotient presentation has a computable presentation" is independent of* ZFC.

The consistent failure of the statement has been given. We prove the consistency of the statement by proving it in L. Given a computable quotient structure $\mathcal{A} = \langle A, \ldots, \equiv \rangle$, the idea is to build a model by taking the L-least representative of each equivalence class. To recognize that a given real x is the L-least representative of its equivalence class, we attach to each such x an escort y coding an ordinal sufficiently large to allow us to computably verify that x is the L-least representative of its class. We then build the computable presentation from these escorted pairs $\langle x, y \rangle$.
Specifically, let B be the set of pairs $\langle x, y \rangle$ such that y is an L-code for the least ordinal α such that x is an element of L_α, and L_α satisfies:

- that $x \in A$,
- that x is the L-least real that is equivalent to x, and
- that "ω_1 exists".

The key point is that since L_α has all Turing programs and thinks "ω_1 exists", all the computations using reals in L_α either halt or repeat before stage α. So L_α has access to the full, correct computations from reals in L_α. The set B is decidable: given a pair $\langle x, y \rangle$ we can compute from y a code for L_α, so questions of satisfaction in this structure will be decidable, and all facts which must be checked can be expressed in L_α using, when necessary, the programs that compute A and $\equiv$.
Finally, we put a structure on B. For a relation symbol R, let

$$R^{\mathcal{B}}(\langle x_0, y_0 \rangle, \ldots, \langle x_n, y_n \rangle)$$

hold if and only if $R^{\mathcal{A}}(x_0, \ldots, x_n)$. For a function f, define

$$f^{\mathcal{B}}(\langle x_0, y_0 \rangle, \ldots, \langle x_n, y_n \rangle) = \langle x, y \rangle,$$

where x is the L-least member of $f^{\mathcal{A}}(x_0, \ldots, x_n)$ and y is the L-code for which $\langle x, y\rangle \in B$. The structure $\mathcal{B}$ is computable. Hence the statement "every structure with a computable quotient presentation has a computable presentation" is consistent. The statement "every decidable model has a computable presentation" is consistent, but open. It is true for any sufficiently simple language such as one having only relations and a single unary function.

3. Completeness Theorem

Our central question is the infinite time computable analogue to the Completeness Theorem: does every consistent decidable theory have a decidable model? The answer is independent of ZFC.

Theorem 4. *The infinite time computable Completeness Theorem is independent of* ZFC*.*

1. *If $V = L$, then every consistent infinite time decidable theory has an infinite time decidable model, in a computable translation of the language.*
2. *It is relatively consistent with* ZFC *that there is an infinite time decidable theory, in a computably presented language, having no infinite time computable or decidable model in any translation of the language.*

We first consider the positive result. When the language is presented simply enough, the classical Henkin construction is effective.

Theorem 5. *If T is a consistent theory in a computable language coded in $\mathbb{N}$ and T is decidable, then T has a decidable and computable model.*

Corollary 6. *There are infinite time decidable computable models of* PA*,* ZFC*, etc., provided that these theories are consistent.*

The key to obtaining (1) above is that every computably presented language in L can be enumerated nicely enough to carry out the Henkin construction. A language is *well presented* if there is an enumeration $\langle s_\alpha \mid \alpha < \delta\rangle$ of the function, relation and constant symbols of the language, such that from any $\ulcorner s_\alpha \urcorner$, we can compute a code for the sequence of prior symbols. Given a well-presented language, we can similarly compute the sequence of prior formulas from the code for any formula. We can extend any decidable theory T in a well-presented language to a complete Henkin theory $\overline{T}$ by the usual argument. The theory $\overline{T}$ is decidable because, given a Gödel code for a formula ϕ_α, we can construct the codes for all prior formulas and computably reconstruct $\overline{T}$ up to α to see whether ϕ_α is added to the theory. We then construct a decidable model from the constants. Thus every consistent decidable theory in a computably well-presented language has a decidable model.
Now assume $V = L$ and let $\mathcal{L}$ be a computably presented language. Let $S = \langle s_\alpha \mid \alpha < \delta\rangle$ be the sequence of Gödel codes for symbols of the language under the L-ordering. Since this enumeration may not be computable, we mimic the argument used to obtain a computable presentation from a computable quotient presentation by attaching to each α an escort. Specifically, we represent the symbol coded by s_α by the L-code t_α for the pair $\langle \alpha, \gamma_\alpha\rangle$ where γ_α is the least ordinal above α such that L_{γ_α} thinks that s_β exists for all $\beta < \alpha$ and that ω_1 exists. The language $\mathcal{L}$ is well-presented by the enumeration $\langle t_\alpha \mid \alpha < \delta\rangle$. Result (1) above follows. Note however that this enumeration is a translation of the original presentation.
Part (2) of the theorem relies again on the fact that no Σ^1_2 set can have cardinality strictly between ω_1^L and the continuum. Let the theory T be the atomic diagram of the structure $\langle \text{WO}, \equiv\rangle$ where $\equiv$ is the relation of coding the same ordinal, together with an axiom asserting that f is a choice function on the equivalence classes. This theory is decidable. Suppose that a model $\mathcal{M}$ of T is decidable, or even computable.

Then the graph of $f^{\mathcal{M}}$ and the relation $z = c_x^{\mathcal{M}}$ are decidable, and hence Δ^1_2. It follows that the range of $f^{\mathcal{M}}$ restricted to $\{c_x^{\mathcal{M}} \mid x \in \text{WO}\}$ is a Σ^1_2 set of size ω_1:

$$y \in \{f^{\mathcal{M}}(c_x^{\mathcal{M}}) \mid x \in \text{WO}\} \text{ iff } \exists z, x(x \in \text{WO} \land z = c_x^{\mathcal{M}} \land \langle z, y\rangle \in f^{\mathcal{M}})$$

Since it is consistent that no Σ^1_2 set has size ω_1, (2) follows.

4. Other results

There are many infinite time analogues to the Löwenheim-Skolem Theorem, and many open questions. We obtain positive results when the language of the elementary diagram of a decidable model is well-presented. For example the infinite time computable upward Löwenheim-Skolem theorem holds in L. Our analysis leaves open, however, the question of whether it is consistent with ZFC that there could be a decidable countable model having no size continuum decidable elementary extension. If so, the infinite time computable upward Löwenheim-Skolem theorem is independent of ZFC.
We produce a strong violation of the downward Löwenheim-Skolem Theorem by coding the infinite-time halting problem into a model.

Theorem 7. *There is an infinite time computable structure with underlying set* $\mathbb{R}$ *having no infinite time computable proper elementary substructure.*

In the classical computability theory, the inverse of a computable injection is also computable since once can effectively search the domain for the inverse image of any given point in the range. In the infinite time theory, where we cannot in general search the domain, some computable injective functions lack computable inverses. As an application of these ideas, we consider the infinite-time Cantor-Schröeder-Bernstein Theorem. The most natural statement would be:

> Let A and B be sets with computable injections $f : A \longrightarrow B$ and $g : B \longrightarrow A$. Then there is a computable bijection $h : A \longrightarrow B$ with computable inverse.

In fact, two additional assumptions are provably necessary: the inverses of f and g must be computable and the ranges of f and g must be decidable. Note that A and B are semi-decidable (which is sufficient) since each is the domain of a computable function.
We conclude by stating an odd fact regarding decidable presentations of *transitive* models of ZFC. The smallest transitive model of ZFC, if it exists, has a decidable, computable presentation. But building this model (provably) requires a trick: smuggling in an L-code for the model either as part of the model's underlying set or as part of the language.

5. Future Directions

There are several natural questions left unanswered by our inquiry, and new areas have opened to which little or no thought has yet been given. The most fundamental open question within the existing scope of our inquiry is whether every decidable model can be computably presented. This is true in L. There are also many open questions relating to the Löwenheim-Skolem Theorem.
Since formulas from the infinitary language $L_{\omega_1,\omega}$ can be coded by reals, infinite time Turing machines may be robust enough to develop a corresponding computable model theory. In addition, it may be possible to extend the program of computable model theory to infinite time Turing machines with an alternate architecture.

There has been recent progress on infinite time computable equivalence relations under computable reductions viewed as an analogue to the theory of Borel equivalence relations under Borel reducibility. Some of these ideas are implicit in the analysis of the computable quotient presentation in [HMSW07].

References

[EGNR98] Yuri L. Ershov and Sergey S. Goncharov and Anil Nerode and Jeffrey B. Remmel. *Handbook of Recursive Mathematics, Volume 1: Recursive Model Theory*, volume 138 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, 1998.

[HL00] Joel David Hamkins and Andy Lewis. Infinite time Turing machines. *J. Symbolic Logic*, 65(2):567–604, 2000.

[HMSW07] J. D. Hamkins, R. Miller, D. Seabold, and S. Warner. Infinite time computable model theory. New Computational Paradigms: Changing Conceptions of What is Computable. In S.B. Cooper and Benedikt Löwe and Andrea Sorbi, editors, *New Computational Paradigms: Changing Conceptions of What is Computable*, Springer 2007.

Descriptive set theoretic aspects of transfinite computation

Philip D. Welch
University of Bristol

Monday, 11:15 – 12:30

I Prologue: Kleene Recursion, a thumbnail sketch

II Infinite Time Turing Machines

I Prologue: Kleene Recursion

• An equational calculus for developing the notion of *recursion on a higher type*.

$$x \in A \simeq \{e\}(x, y, B, {}^2E) \downarrow 1 \quad A, B \subseteq \mathbb{R}(= 2^{\mathbb{N}})$$

Intention: to parallel the equational calculus developed for recursion on the integers.

It can be characterised by a model of computation in which a computational device had a

(i) *countably infinite memory,* and

(ii) *an ability to manipulate (search through, write to) that memory in finite time;* optionally

(iii) *an ability to quiz an oracle (for B) about its entire memory contents.*

We may think of this as a TM with one (or more) infinite tapes on which reals (= infinite sequences of 0's,1's are written) and the ability to ask the oracle at any stage of the computation as to whether the current real under consideration is in some "oracle set" $B \subseteq \mathbb{R}$.

• This is not to be conceived as a computation that runs in transfinite segments of discrete time, but rather as one that makes calls for values from *subcomputations*; a computation thus has a *wellfounded finite path tree* structure.

• The course of computation may evolve its own tree structure as it progresses according to its instruction set; we may also view a "machine" as having a previously determined tree structure as part of its "instructions" or program. In short the machine may be viewed as determined by a (finite) program together with a (code for) an infinite finite path-tree.

What is this mathematically?

Kleene degrees: Let $A, B \subseteq \mathbb{R}$; we say that

$A \leq_K B$ iff there is some comp'l arrangement P such as above so that for any $x \in \mathbb{R}$ ($x \overset{\in}{\underset{\notin}{}} \iff P^B(x) \downarrow \frac{1}{0}$)
iff there are Σ_1-formulae in $\mathcal{L}_{\in,\dot{X}}$ φ_1, φ_2, there is $y \in \mathbb{R}$ so that for any $x \in \mathbb{R}(x \in A \iff L_{\omega_1^{B,y,x}}[B,y,x] \models \varphi_1[B,y,x]$
$\iff L_{\omega_1^{B,y,x}}[B,y,x] \models \neg\varphi_2[B,y,x]$)
(here $\omega_1^{B,y,x}$ is the least (B,y,x)-admissible ordinal).
0_K contains $\varnothing, \mathbb{R}$, and in fact consists of the *Borel sets.*

$0'_K$ (the K-degree of a complete Kleene semi-recursive set of reals) contains WO the set of reals coding wellorders, and so a complete Π^1_1 set of reals. In fact it consists of the co-analytic, so Π^1_1sets.

• (Solovay)[7] $AD \Rightarrow$ K-degrees are wellordered. Indeed a K-degree forms a boldface pointclasses being closed under continuous preimages.
•(Harrington-Steel) [8], [4] Det(Bool(Π^1_1)) $\iff \neg\exists A(0 <_K A <_K \text{WO})$
(Simpson) [5] $V = L \Longrightarrow$the opposite.

II Infinite Time Turing Machine Computations.

We turn to the Infinite Time Turing Machine construction of [3].

Definition 1. $\lambda =_{\text{df}} \sup\{\alpha | \alpha = \|x\| \exists e \in \omega P_e(0) \downarrow x\}$*;*

$\gamma =_{\text{df}} \sup\{\alpha | \exists e \in \omega P_e(0)^\alpha \downarrow x\}$.
$\zeta =_{\text{df}} \sup\{\alpha | \alpha = \|x\| \exists e \in \omega x$ *is the eventual contents of the output tape (OT) of* $P_e(0)$ *from point on*$\}$.
$\Sigma =_{\text{df}} \sup\{\alpha | \alpha = \|x\| \exists e \in \omega x$ *appears on some tape of* $P_e(0)$ *at some time* $\nu\}$.

Theorem 2. *If ([10] Thm.1.1)* $\gamma = \lambda$. *By relativisation:* $\gamma^x = \lambda^x$ *for* $x \in 2^{\mathbb{N}}$.

Proof: To show: $\gamma \leq \lambda$. Suppose $P_e(0) \downarrow^\eta$. So we wish to show that $\eta < \lambda$. It suffices to show:
(1) $\eta < \zeta$
(As $\eta < \zeta$ there is f so that $P_f(0)$ eventually has a code y for an ordinal μ with $\eta < \mu < \zeta$ on its OT. Consider the algorithm that simulates "P_f" now & then pausing and performing a run of P_e along the ordinal μ' coded by the current y' on the "P_f" OT. Eventually μ' will be long enough to see that "P_e" halts. So we halt the overall simulation putting y' on the overall OT. But $\|y'\| > \eta$.)
Let $C_i(\nu) \in \{0,1\}$ be the value of the i'th cell at time ν in the course of computation of P_e. Let, for Lim(ν), $\delta_i(\nu) =_{\text{df}} \sup\{\nu' < \nu | \nu = 0 \vee C_i(\nu+1) \neq C_i(\nu)\}$. So $\delta_i(\nu) \leq \nu$.
It suffices to show:
(2) $\forall i \delta_i(\Sigma) < \Sigma \longrightarrow \delta_i(\Sigma) < \zeta$.
(If true, then a) the "snapshot" of the cell values of the computation of $P_e(0)$ at time ζ is exactly that at Σ **and** b) any C_i stabilized at time ζ does not change its value, eg $0 \to 1 \to 0$, in (ζ, Σ). Hence $\eta \notin (\zeta, \Sigma)$. Hence if P_e has not halted by time ζ it enters a permanent loop. So $\eta \not\geq \Sigma$. Hence (1) holds.)
Proof of (2). Fix i. Let $\mathcal{U}$ be the universal machine. Let $\tau(\nu)$= grand sum of all $\|y\|$ where $y \in \text{WO} \wedge y$ appears somewhere on one of $\mathcal{U}$'s "tapes" at time ν.
By definition of Σ:
(3) $\tau(\nu)$becomes unbounded in Σ.
We simulate " $\mathcal{U}$", at fixed times pausing and: (i) calculating the current "$\tau(\nu)$" ; (ii) simulate a run of "P_e" along "$\tau(\nu)$" and calculate $\delta_i(\tau(\nu))$. We check & compare that to our current "best estimate" of $\delta_i(\Sigma)$, $\delta_i(\tau(\nu'))$, which we obtained at some

earlier stage ν', an d have stored. If, and only if, it is longer, do we replace (the code for) $\delta_i(\tau(\nu'))$ by that for $\delta_i(\tau(\nu))$. By (3) we *eventually* have a code for the true $\delta_i(\Sigma)$ on our overall OT. But then $\delta_i(\Sigma) < \zeta$. QED (2) and Thm.

Remark 3. *If we replaced P_e in the above by the universal program itself, the argument at (2) would prove: the snapshot of the* universal machine *at time ζ is the same as that at time Σ: hence the universal machine enters its permanent loop at time ζ. As ζ is the supremum of the eventually writable ordinals it is easy to argue that it does not enter a permanent loop at any earlier stage.*

This enables us to prove:

Theorem 4. *(Normal Form Theorem* [12]*)* $\forall e \exists e' \forall x \in 2^{\mathbb{N}}$
$P_e(x)\downarrow \longrightarrow [P_{e'}(x)\downarrow y$ *where* $y \in 2^{\mathbb{N}}$ *codes a wellordered course-of-computation sequence for* $P_e(x)\downarrow]$.
Moreover the map $e \longrightarrow e'$ *is effective (in the usual Turing sense).*

Theorem 5. ***(The*** λ, ζ, Σ***-Theorem)*** [9], [10] *Any ITTM Type 1 computation* $P_e(x)$ *which halts does so by time* λ^x *where*
$\zeta^x =_{\mathrm{df}} \mu\zeta[\ \exists\,\text{least}\,\Sigma = \Sigma^x > \zeta L_\zeta[x] \prec_{\Sigma_2} L_{\Sigma^x}[x]]$ *and*
$\lambda^x =_{\mathrm{df}} \mu\lambda[\ L_\lambda[x] \prec_{\Sigma_1} L_{\zeta^x}]$.

Remark: ζ^x can be characterised as that ordinal at which the universal ITTM $\mathcal{U}$ on input $x \in 2^{\mathbb{N}}$starts to cycle.
Proof: (Sketch). We take $x = \varnothing$, and show simply that (a) $L_\zeta \prec_{\Sigma_2} L_\Sigma$ and (b) (ζ, Σ) is the lexicographically least pair satisfying (a). To do (b) first, simply note that we may run the universal machine $\mathcal{U}$ inside L. Since the limsup rules for cell values *etc.* can be expressed essentially in a Σ_2 way, we have that for any ζ', Σ' satisfying (a) we should have the snapshots at ζ' and Σ' being identical. But we have already remarked above that (ζ, Σ) are the first two points on the permanent looping cycle of $\mathcal{U}$.
For (a) we mimic the argument of Theorem 2. Suppose $\varphi \equiv \exists u \forall v \psi(u, v, \xi)$ where $\xi < \zeta$ and $L_\Sigma \models \varphi$. Let $u_0 \in L_\Sigma$ be such that $L_\Sigma \models \forall v \psi(u_0, v, \xi)$. Set $\rho = \rho_L(u_0)$ to be the L-rank of u_0. Let $\delta(\Sigma, \varphi, \xi) =_{\mathrm{df}} \inf\{\delta < \Sigma | \forall \delta' \in (\delta, \Sigma]\ L_{\delta'} \models \exists u \forall v \psi(u, v, \xi)\}$. Then observe $\delta(\Sigma, \varphi, \xi) \leq \rho$.
Claim: $\delta(\Sigma, \varphi, \xi) < \zeta$.
We first note that since $\xi < \zeta$ ξ is eventually written by some program $P_g(0)$. We let $P_e(0)$ be the program that (i) computes grand sum ordinals $\sigma(\nu)$ as in Theorem 2; (ii) from those ordinals computes a code for $L_{\sigma(\nu)}$; (iii) by simulating $P_g(0)$, calculates an approximation $\xi(\nu)$ to the latter's output tape after $\sigma(\nu)$ many stages; (iv) calculates

$$\delta(\sigma(\nu), \varphi, \xi(\nu)) =_{\mathrm{df}} \inf\{\delta < \sigma(\nu) | \forall \delta' \in (\delta, \sigma(\nu) L_\delta \models \exists u \forall v \psi(u, v, \xi(\nu))\}$$

(v) it then writes this value $\delta(\sigma(\nu), \varphi, \xi(\nu))$ to a reserved area of tape, R say, but only *after* inspecting the current contents of R, and if that contents codes an ordinal δ' say, it checks that $\delta' < \delta(\sigma(\nu), \varphi, \xi(\nu))$. (Thus if δ' is a larger ordinal than $\delta(\sigma(\nu), \varphi, \xi(\nu))$ δ' is left on R.)
The first point of this is that as ξ is eventually writable, for some $\mu < \zeta$ as long as $\sigma(\nu) \geq \mu$ we shall have that $\xi = \xi(\nu)$. But this condition is easily met as $\sigma(\nu)$ will eventually become unbounded in Σ and stay above ζ. (Hence the correct computation of ξ becomes virtually a side issue.) The second point is that once $\sigma(\nu) > \rho, \mu$ we shall have a code for some ordinal $\delta' \geq \delta(\sigma(\nu), \varphi, \xi)$ written to R, but there will never be any cause for it to be overwritten at a later time ν'. However that implies it is eventually writable and hence less than ζ. By definition of ρ this yields the Claim.

By the definition of $\delta(\Sigma, \varphi, \xi)$ we see that $L_\zeta \models \varphi(\xi)$ so we are done. QED

• From the above analysis it transpires that the ordinal assignment $x \rightarrowtail \lambda^x$ (or ζ^x) satisfies a *Spector Criterion:*

$$x \leq_\infty y \longrightarrow (x^\nabla \leq_\infty y \longleftrightarrow \lambda^x < \lambda^y).$$

• Moral: Even on just sets of integers, this is not going to be like Turing degrees.

Corollary 6. *(i)* $0^\nabla \equiv_1 \Sigma_1\text{-Th}(L_\lambda) \equiv_1 \Sigma_1\text{-Th}(L_\zeta)$.
(ii) The infinite time decidable sets of integers $= L_\lambda \cap 2^{\mathbb{N}}$.

Let F be the class of "quickly computable" reals: $x \in F \iff x \in L_{\lambda^x}$ (Think of the "quickly constructible reals $Q = \{x | x \in L_{\omega_1^x}\}$.)

Lemma 7. [10] $\mathcal{EW} \subseteq F$.

Proof: Let $x \in L_\zeta$. Let $\rho = \rho_L(x) < \zeta \leq \zeta^x$. We run a computation $P_g(x)$ that simulates a run of $\mathcal{U}$ until it finds a code for an ordinal τ and L_τ that shows $x \in L_\tau$. It then halts with output τ. Hence $\tau < \lambda^x$. QED

• The complete AQI-set of integers $\equiv_1 \Sigma_2\text{-Th}(L_\zeta) \equiv_1 S_\zeta$ - the "snapshot" of $\mathcal{U}$ at time ζ. (AQI="*arithmetically quasi-inductive*" -Burgess)

Minimality in the ITTM degrees of reals.

We define the notion of *minimal degree* of $d \in 2^{\mathbb{N}}$ as

$$0_\infty <_\infty d \wedge \forall e[0_\infty <_\infty e \leq_\infty d \longrightarrow d \leq_\infty e].$$

Theorem 8. [11] *There are continuum many reals of minimal $\leq_\infty$-degree.*

Proof: An H. Friedman style double fusion argument of $\leq_\infty$-pointed and L_δ-pointed trees. QED

• Moral: This is not going to be like Δ^1_1-degrees either. It's firmly Δ^1_2-degree like. *cf:*

Theorem 9. *(Hamkins-Lewis)* [2] *There is no real z with $0_\infty <_\infty z <_\infty 0^\nabla$.*

Proof: $0 <_\infty z \longrightarrow z \notin L_\lambda$. As $z \in F$ (Lemma 7), $z \in L_{\lambda^z}$; hence $\lambda^z > \lambda$. But then $0^\nabla \leq_\infty z$. QED

For *higher type* ITTM computations it seems appropriate to talk again about boldface semi-decidable sets of reals. We thus again talk about programs together with an "information" real y:

Definition 10. $A \leq_\infty B$ *iff for some* $e \in \omega$,*some* $y \in \mathbb{R} : \forall x (x \overset{\in}{\underset{\notin}{}} A \iff P_e^{y,B}(x) \downarrow \frac{1}{0})$
iff there are Σ_1-formulae in $\mathcal{L}_{\in,\dot{X}}$ φ_1, φ_2, and $y \in \mathbb{R}$, so that

$$\forall x \in \mathbb{R}(x \in A \iff L_{\zeta^{B,y,x}}[B,y,x] \models \varphi_1[B,y,x]$$
$$\iff L_{\zeta^{B,y,x}}[B,y,x] \models \neg\varphi_2[B,y,x])$$

The $\leq_\infty$-degrees so formed are then boldface pointclasses within the Wadge hierarchy, with 0 and $0^\blacktriangleleft$ as the (degrees of the) ∞-recursive, and ∞-semirecursive sets of reals, respectively.

Definition 11. *Let Γ_0 be the class of ∞-semi-decidable sets of reals.*

Theorem 12. *(V=L)* $\quad 0 <_\infty \underline{F} <_\infty 0^\blacktriangleleft$

Theorem 13. *(Det(Bool(Γ_0)))* *There is no $A \subseteq \mathbb{R}$ with* $0 <_\infty \underline{A} <_\infty 0^\blacktriangleleft$.

Question: How strong is $\mathrm{Det}(\Gamma_0)$?

II.2 Bounding Lemmata for computation times

Definition 14. [1], [6] *(i) $A \in P^{\alpha}$ if $\exists\beta < \alpha$ there is an infinite time Turing machine deciding each $x \in A$ in fewer than β many steps.*
(ii) Let $f : \mathbb{R} \longrightarrow \mathrm{On}$ be a Turing invariant function. (i) $A \in P^f$ if there is an infinite time Turing machine deciding each $x \in A$ in fewer than $f(x)$ many steps.

Theorem 15 (Deolalikar-Hamkins-Schindler)**.** [1] *Let $f_0(x) = \omega^x_{1\,\mathrm{ck}}$; then $P^{f_0} = P^{\omega_{1\,\mathrm{ck}}}$.*

At first sight this looks surprising: as they say

> "...because it means that although the computations deciding $x \in A$ for $A \in P^{f_0}$ are allowed to compute up to $\omega^x_{1\,\mathrm{ck}}$, in fact there is an algorithm needing uniformly fewer than $\omega_{1\,\mathrm{ck}}$ many steps. So the difference between $\omega^x_{1\,\mathrm{ck}}$ and ω_1, which can be substantial, gives no advantage at all in computation. An affirmative answer to the following question would explain this phenomenon completely.
>
> **Question 6** *Suppose an algorithm halts on each input x in fewer than $\omega^x_{1\,\mathrm{ck}}$ steps. Then does it halt uniformly before $\omega_{1\,\mathrm{ck}}$?* "

Theorem 16. *(Uniform Bounding Lemma) Let $F : \mathbb{R} \longrightarrow \mathbb{R}$ be ITTM-computable and total as witnessed by P_e. If $\forall x P_e(x) \downarrow^{<\omega^x_1}$ (meaning halts in ¡ω^x_1 steps) then $\exists\gamma$¡$\omega_{1\,\mathrm{ck}}$ $\forall x P_e(x) \downarrow^{<\gamma}$.*

Proof: Essentially an application of Σ^1_1-Bounding.

Theorem 17 (D-H-S)**.** [1] $P^{\omega_{1\,\mathrm{ck}}+1} = P^{\omega_{1\,\mathrm{ck}}}$.

Theorem 18. *(Bounding Lemma) Suppose β be admissible. Let F be ITTM-computable, total so that $\forall x P_e(x) \downarrow^{\leq\beta}$ where P_e computes F. Then $\exists\gamma < \beta$ $\forall x P_e(x) \downarrow^{<\gamma}$.*

Proof: Ditto: Barwise Compactness.

Definition 19. *$A \in NP^f$ if $\exists e$ $\forall z P_e(z) \downarrow^{<f(z)}$ with $x \in A \Longleftrightarrow \exists y \in 2^{\mathbb{N}}[P_e(\langle x \oplus y\rangle) \downarrow 1^{<f(x)}$*

Lemma 20. *(*[13] *Lemma 2.5) If α is a clockable ordinal, then every ordinal less than the next admissible ordinal beyond α is writable in time $\alpha + \omega$.*

Proposition 21. *Let $\beta \leq \lambda$ be such that β is an admissible limit of admissibles but is not interior to any gap in the clockables (i.e., it is a limit of clockables). Then $P^{\beta} \cap \mathcal{P}(\mathbb{N}) = NP^{\beta} \cap \mathcal{P}(\mathbb{N})$.*

Proof. Let $A \in NP^{\beta} \cap \mathcal{P}(\mathbb{N})$. Let P_e witness this: $\forall n, y\ P_e(n, y) \downarrow^{<\beta}$ and $\forall n[n \in A \Longleftrightarrow \exists y P_e(n, y) \downarrow 1]$. The Bounding Lemma shows that there is a smaller bound $\gamma_0 < \beta$ for the lengths of all these computations. Hence if $n \in A$ then there is a y witnessing this, with $P_e(n, y) \downarrow 1$ and converging in $\leq \gamma_0$. steps. Let $u \in L_\beta \cap \mathrm{WO}$ have rank γ_0. Set:

$$B_n = \{z : \exists y(z \text{ codes a wellfounded computation witnessing } P_e(n, y) \downarrow^{\|u\|} 1\ )\}$$

Again $\varnothing \neq B_n \in \Sigma^1_1(u)$. As above, appealing to the Kleene Basis theorem again, there are witnessing $z, y_0 \in L_{\gamma_0^+ +1}$ if $n \in A$ (where γ_0^+ is next admissible above γ_0.) In other words to test for membership in A all we have to do is search through potential NP-witnesses y in $L_{\gamma_0^+ +1} \in L_\beta$. But this puts $A \in \Delta_1^{L_\beta}(\{\gamma_0\})$. By our assumption on β, by Lemma 20, γ_0 is itself writable by some program P_f in time $< \gamma_0^+$. Putting this together $A \in \Delta_1^{L_\beta}$, so $A \in P_\beta$. □

References

[1] V. Deolalikar, J.D. Hamkins, and R-D. Schindler. $P \neq NP \cap co - NP$ for the infinite time Turing machines. *Journal of Logic and Computation*, 15:577–592, October 2005.

[2] J. D. Hamkins and A. Lewis. Post's problem for supertasks has both positive and negative solutions. *Archive for Mathematical Logic.*

[3] J. D. Hamkins and A. Lewis. Infinite time Turing machines. *Journal of Symbolic Logic*, 65(2):567–604, 2000.

[4] L. Harrington. Analytic determinacy and $0^{\#}$. *JSL*, 43(4):684–693, 1978.

[5] K.Hrbacek and S.Simpson. On Kleene degrees of analytic sets. In H.J.Keisler J.Barwise and K.Kunen, editors, *Proceedings of the Kleene Symposium*, Studies in Logic, pages 347–352. North-Holland, 1980.

[6] R-D. Schindler. $P \neq NP$ for infinite time Turing machines. *Monatsheft für Mathematik*, 139(4):335–340, 2003.

[7] R.M. Solovay. Determinacy and type 2 recursion. *Journal of Symbolic Logic*, 36:374, 1971.

[8] J. R. Steel. Analytic sets and Borel isomorphisms. *Fundamenta Mathematicae*, 108:83–88, 1980.

[9] P. D. Welch. Minimality arguments in the infinite time Turing degrees. In S.B.Cooper and J.K.Truss, editors, *Sets and Proofs: Proc. Logic Colloquium 1997, Leeds*, volume 258 of *London Mathematical Society Lecture Notes in Mathematics*. C.U.P., 1999.

[10] P. D. Welch. Eventually infinite time Turing degrees: infinite time decidable reals. *Journal for Symbolic Logic*, 65(3):1193–1203, 2000.

[11] P. D. Welch. The length of infinite time Turing machine computations. *Bulletin of the London Mathematical Society*, 32:129–136, 2000.

[12] P. D. Welch. Post's and other problems in higher type supertasks. In B. Löwe B. Piwinger T. Räsch, editor, *Classical and New Paradigms of Computation and their Complexity hierarchies, Papers of the Conference Foundations of the Formal Sciences III*, volume 23 of *Trends in logic*, pages 223–237. Kluwer, Oct 2004.

[13] P.D. Welch. Arithmetical quasi-inductive definitions and the transfinite action of one tape Turing machines. *typescript*, 2004.

Open questions

(1) For which ordinals α, $P_\alpha^{HK} \subsetneq PSPACE_\alpha^{HK}$?
Asked by Joost Winter. This is true for all successors and some limits.

(2) How can we show that a set A is not in $PSPACE_f^{HK}$?
Asked by Joost Winter.

(3) Can there be incomparable "lost melody" reals? That is, can there be two reals c, d such that $\{c\}, \{d\}$ are each ITTM-decidable but c is not d-writable and d is not c-writable?
Asked by Joel D. Hamkins, \$5 for the answer.
Answered by Philip Welch: no such pairs exist.

(4) What is the lower bound for the number of registers (k) needed by an ORM to compute L?
Asked by Peter Koepke, €(25−k) for the answer. By ORM it is meant an Ordinal Register Machine, under the definition in "Register computations in ordinals" to be found in ArXiv.

(5) What are the provably in KP_i clockable ordinals/halting programmes?
Asked by Michael Rathjen.

(6) For ITTMs, is it true that $PSPACE \neq P$, $PSPACE^+ \neq P^+$ and/or $PSPACE^{++} \neq P^{++}$?
Asked by Benedikt Löwe, €1 per answer.

(7) Does every decidable model have a computable presentation?
Asked by Daniel Seabold and Russel Miller.

(8) Is it consistent that every decidable theory has a computable/decidable model, without computably translating the language?
Asked by the New York team.

(9) Can we characterise the gaps in the clockable ordinals of OTMs, e.g., is every gap-starting ordinal admissible?
Asked by the New York team. This is true for ITTMs; Philip D. Welch.
Subquestions to this:
 a) For example, can every gap-starting ordinal be something even better, like "strongly admissible"? (Define strongly admissible?)
 Asked by Jonas Reitz, €5 for the answer.
 Answered by Jonas Reitz and Steve Warner: no admissible ordinal is OTM clockable.
 b) Is ω_1^{ck}, OTM clockable?
 Asked by Joel D. Hamkins, \$1 for the answer.
 Answered by Jonas Reitz and Steve Warner: no admissible ordinal is OTM clockable.

(10) Does every countable collection of ITTM-degrees have a minimal upper bound?
Asked by Philip D. Welch, €50 for the answer. It is true for degrees represented by accidentally writable reals. Harvey Friedman showed it also for Δ_2^1 degrees.

(11) What is the structure of the ITTM-degrees (of reals) represented by a real? In particular, can you invert the jump, i.e., if $0^\triangledown \leq_\infty c$ then is there a d such that $d^\triangledown =_\infty c$)? Also, to what degree is there minimality, density, etc.?
Asked by Joel D. Hamkins, \$5 for the inversion of the jump.

(12) Work out a theory of ordinal stack machines and their complexity.
Asked by Peter Koepke.

(13) For ITTMs, for $z \in \mathbb{R}$ if $z \leq_\infty 0^\blacktriangledown$, then is z eventually writable?

Asked by Joel D.Hamkins, \$5 for the answer.

Answered by Philip Welch: no, because Σ is $0^\blacktriangledown$-writable.

a) A related question, what is $\lambda^{0^\blacktriangledown}$, i.e., the supremum of $0^\blacktriangledown$-writable ordinals? Is it ζ, i.e., the supremum of the eventually writable ordinals?

Again by Joel D.Hamkins, also \$5 for the answer.

Answered by Philip Welch: it is larger than Σ and hence ζ.

(14) In analogy with the fact that every countable admissible ordinal is $\omega^x_{1,ck}$, i.e., the supremum of the x-recursive ordinals, for some real x. Can you characterise the ordinals of the form λ^x, i.e., the supremum of the x-writable ordinals? Also, the same for ζ^x.

Asked by Joel D.Hamkins, \$5 for the answer.

(15) Is there a real z such that $\zeta = \lambda^z$?

Asked by Joel D. Hamkins, \$5 for the answer.

Answered by Philip Welch: no. In fact for any A, $\zeta \neq \lambda^A$.

(16) Find equalities/inequalities/famous open questions from standard complexity theory, define their analogues for ITTM/OTM/ORM and prove/disprove them!

Asked by Benedikt Löwe.

(17) Suppose $f : 2^\omega \to \omega$ is Turing invariant (i.e., $x \leq_T y$ implies that $f(x) \leq_T f(y)$) and that for all $x \in 2^\omega$, $f(x)$ is an admissible limit of gap-starting and gap-ending ordinals. Then does $P^f = NP^f$?

Asked by Philip D.Welch. Background: Deolalikar, Hamkins and Schindler show for many $g : 2^\omega \to \omega$ that $P^g \cap \mathcal{P}(\mathbb{N}) \neq NP^g \cap \mathcal{P}(\mathbb{N})$, but for f as above, we have that $P^f \cap \mathcal{P}(\mathbb{N}) = NP^f \cap \mathcal{P}(\mathbb{N})$. The question is whether equality holds at the higher type.

(18) Investigate the structure of the "eventually decidable" (also called "arithmetically quasi-inductive") sets.

Definitions:

A set A is in AQI (is arithmetically quasi-inductive), iff there is a $\phi \in \Sigma_2$ such that for all x, $x \in A \leftrightarrow L_{\zeta^x}[x] \models \phi(x)$ (this might be called the "eventually semi-decidable" sets but the "AQI" is due to Burgess).

We write $A \leq_{ev.\infty} B$ iff there are $\phi_1, \phi_2 \in \Sigma_2$ and $y \in 2^\omega$ such that for every x,

$$x \in A \leftrightarrow L_{\zeta^{x,y,B}}[x, y, B] \models \phi_1[x] \leftrightarrow L_{\zeta^{x,y,B}}[x, y, B] \models \neg\phi_2[x].$$

C.f. analogous definition from Kleene degrees. The "AQI" class forms a Spector class and satisfies Prewellordering.

Asked by Philip D.Welch.

(19) Let Γ_0 be the pointclass of the ∞-semi-decidable sets, i.e., ITTM semi-decidable sets. Determine the strength of $Det(\Gamma_0)$.

Asked by Philip D.Welch. Background: $Det(\Gamma_0)$ implies that there is an inner model of a strong cardinal.

(20) Are there Borel equivalence relations E and F such that E computably reduces to F, but does not Borel reduce? Definitions: *E computably reduces to F* if there is an infinite time computable function f (computable from a real parameter) such that $x \; E \; y \iff f(x) \; F \; f(y)$; it Borel reduces if there is a Borel such function.

Asked by Sam Coskey and Joel Hamkins.

Further material

Two observations regarding infinite time Turing machines

Sy-David Friedman
KGRC, University of Vienna[4]
and
Philip D. Welch
University of Bristol

Abstract. We observe: (I) There is a *"Theory Machine"* that can write down the Σ_2-Theories of the levels of the J-hierarchy up to Σ (the least Σ such that some smaller L_ζ is Σ_2 elementary in L_Σ) in a uniform way. Moreover, below Σ these theories are all distinct. This yields information about the halting times of ITTM's. (II) The ITTM degrees of the semi-recursive singletons are well-ordered in order type the least stable, i.e., the least σ such that L_σ is Σ_1 elementary in L.

The *Theory Machine* generates theories of initial segments of the J-hierarchy. This machine can be used to prove the "ζ-Σ theorem" and analyse the halting times of ITTM's.

The idea of the Theory Machine is to write down the theory of $(J_\alpha, \in)$ (appropriately Gödel-numbered) on the output tape at computation stage $\omega^2 \cdot (\alpha + 1)$, for as long as possible. This will be easy to arrange for successor α, as long as a code for the structure $(J_\alpha, \in)$ can be read off from its theory. For limit α, the machine performs a liminf operation, resulting in a theory T_α; we show that the Σ_2 theory of $(J_\alpha, \in)$ is recursive in the Turing jump of T_α, uniformly in α. Provided a code for the structure $(J_\alpha, \in)$ can be read off from its Σ_2 theory, this will enable the machine to write down the theory of $(J_\alpha, \in)$ at stage $\omega^2 \cdot \alpha + \omega^2$. A fine-structural analysis shows that as long as α is less than the least Σ such that J_ζ is Σ_2 elementary in J_Σ for some $\zeta < \Sigma$, a code for $(J_\alpha, \in)$ can indeed be read off from its Σ_2 theory, uniformly. Therefore the machine will produce distinct theories of structures $(J_\alpha, \in)$ for $\alpha < \Sigma$, and then at stage Σ repeat what it wrote on the output tape at stage ζ.

A corollary is that, up to a "small" error, the halting times of ITTM's are exactly the ordinals $\alpha < \Sigma$ where sentences become true for the first time in the J-hierarchy, i.e., such that some sentence φ of set theory holds in $(J_\alpha, \in)$ but not in $(J_\beta, \in)$ for any $\beta < \alpha$.

The following two claims are crucial to verifying properties of the theory machine.

Lemma 1. *For a limit λ, let T denote the set of Σ_2 sentences that are true in $(J_\alpha, \in)$ for sufficiently large $\alpha < \lambda$. Then the Σ_2 theory of $(J_\lambda, \in)$ is RE in T. Moreover an index for this RE reduction is uniform in λ.*

[4]The first author was supported by FWF Grants P16334-NO5 and P16790-NO4.

Proof. Let φ be a Σ_2 sentence and write φ as $\exists x \psi(x)$ where $\psi(x)$ is Π_1. Also let $h_1(n, x)$ denote the canonical Σ_1 Skolem function; h_1 has a parameter-free Σ_1 definition and for any α, h_1 interpreted in J_α is a partial function from $\omega \times J_\alpha$ into J_α whose range on $\omega \times [A]^{<\omega}$ is the Σ_1 Skolem hull of A in $(J_\alpha, \in)$ (i.e., the universe of the least Σ_1 elementary submodel of $(J_\alpha, \in)$ containing A), for any $A \subseteq J_\alpha$. We say that an ordinal α is Σ_1 *stable* (in the universe) iff every true Σ_1 sentence with parameters from J_α is true in $(J_\alpha, \in)$.

We have the following equivalence:

$(J_\lambda, \in)$ satisfies φ iff
For some n, the following holds in $(J_\alpha, \in)$ for large enough $\alpha < \lambda$: There is a β which is either 0 or Σ_1 stable such that either φ holds in $(J_\beta, \in)$ or $h_1(n, \beta)$ is defined and $\psi(h_1(n, \beta))$ holds.

This equivalence is verified as follows:

Suppose that $(J_\lambda, \in)$ satisfies φ. If $(J_\beta, \in)$ satisfies φ for some β which is Σ_1 stable in λ (i.e., $\beta < \lambda$ and $(J_\beta, \in)$ is Σ_1 elementary in $(J_\lambda, \in)$), then for all α between β and λ, φ will also hold in $(J_\alpha, \in)$, as β is also Σ_1 stable in α. So the right half of the equivalence holds in this case. Otherwise let β be the largest β which is Σ_1 stable in λ (or 0 is there is no β which is Σ_1 stable in λ). Then every element of J_λ is of the form $h_1(n, \beta)$ for some n (as the Σ_1 Skolem hull of $\{\beta\}$ in $(J_\lambda, \in)$ is all of J_λ). Choose n so that $\psi(h_1(n, \beta))$ holds in J_λ. Then for sufficiently large $\alpha < \lambda$, $h_1(n, \beta)$ is defined in $(J_\alpha, \in)$, and $\psi(h_1(n, \beta))$ holds in $(J_\alpha, \in)$ as ψ is Π_1. So the right half of the equivalence also holds in this case.

Conversely, suppose that the right half of the equivalence holds and choose n to witness that. First suppose that the Σ_1 stables in λ are cofinal in λ. Then apply the right half of the equivalence to some α which is Σ_1 stable in λ. Then either φ holds in $(J_\beta, \in)$ for some β which is Σ_1 stable in α or $\psi(h_1(n, \beta))$ holds in $(J_\alpha, \in)$ for some β; in the former case φ holds in $(J_\lambda, \in)$ as β is Σ_1 stable in λ and in the latter case this holds as α is Σ_1 stable in λ. Now suppose that the Σ_1 stables in λ are bounded in λ and let β be the largest Σ_1 stable in λ (or 0 is there is no β which is Σ_1 stable in λ). Choose α to be sufficiently large in the sense of the right hand side of the equivalence and also such that there are no α-stables greater than β. (For example, choose n so that $h_1(n, \beta)$ is large enough and let α be least so that $h_1(n, \beta)$ is defined in $(J_\alpha, \in)$.) Then applying the right hand side of the equivalence to α, there is a β' which is either 0 or Σ_1 stable in α such that either φ holds in $(J_{\beta'}, \in)$ or $\psi(h_1(n, \beta'))$ holds in $(J_\alpha, \in)$. In the former case, β' is at most β and therefore is Σ_1 stable in λ; it follows that φ holds in $(J_\lambda, \in)$. In the latter case, argue as follows: If β' is less than β, then $h_1(n, \beta')$ in fact belongs to J_β and $\psi(h_1(n, \beta'))$ holds in $(J_\beta, \in)$, implying that φ holds in $(J_\lambda, \in)$. If β' equals β then $\psi(h_1(n, \beta))$ holds in J_α, and as α can be chosen arbitrarily large, $\psi(h_1(n, \beta))$ holds in $(J_\lambda, \in)$; it follows that φ holds in $(J_\lambda, \in)$, as desired.

The equivalence shows that the Σ_2 theory of $(J_\lambda, \in)$ is RE in T. And this RE definition is independent of λ. $\square$

Lemma 2. *Let Σ be least so that some $\zeta < \Sigma$ is Σ_2 stable in Σ (i.e., $(J_\zeta, \in)$ is Σ_2 elementary in $(J_\Sigma, \in)$). Let T^α be the Σ_2 theory of the structure $(J_\alpha, \in)$. Then there is a real code for this structure which is recursive in T^α. Moreover, the reduction of this code to T^α is uniform in α.*

Proof. It suffices to show that there is a partial function f from ω onto J_α which is Σ_2 definable over $(J_\alpha, \in)$ without parameter (uniformly in $\alpha < \Sigma$). For given this, consider the set of n such that $f(n)$ is defined, modulo the equivalence relation $n \sim m$ iff $f(n) = f(m)$, together with the binary relation nEm iff $f(n) \in f(m)$. This yields an isomorphic copy of $(J_\alpha, \in)$.

Let $\psi_n(x)$ be the n-th Π_1 formula with free variable x. Define:

$f'(n) = (m, \beta)$ iff the following hold in $(J_\alpha, \in)$:
i. β is 0 or Σ_1 stable.
ii. $h_1(m, \beta) = x$ is defined.
iii. $\psi_n(x)$ holds.
iv. $\psi_n(x')$ fails for any x' of the form $h_1(m', \beta')$, $\beta' < \beta$, $m' < \omega$.
v. $m' < m \rightarrow h_1(m', \beta)$ is undefined or $\psi_n(h_1(m', \beta))$ fails.

Clauses i-iii are easily seen to be either Σ_1 or Π_1. Clause v is equivalent to a disjunction of a Σ_1 formula and a Π_1 formula. Clause iv is vacuous if β is 0 and otherwise holds in $(J_\alpha, \in)$ iff it holds in $(J_\beta, \in)$; it follows that clause iv is Σ_1. And f' is single-valued and therefore a partial function from ω into J_α which is Σ_2 definable over $(J_\alpha, \in)$ without parameter.

Now define $f(n) = h_1(f'(n))$ and let A be the Σ_1 Skolem hull of the range of f. Then A is the range of a partial function g from ω into J_α which is Σ_2 definable over $(J_\alpha, \in)$ without parameter. (Define $g(n) = h_1(n_0, \langle f(n_1), \ldots, f(n_k)\rangle)$, if n codes the sequence $(n_0, \ldots, n_k)$.)

We claim that $(A, \in)$ is Σ_2 elementary in $(J_\alpha, \in)$: Clearly $(A, \in)$ is Σ_1 elementary in $(J_\alpha, \in)$, as it is the Σ_1 Skolem hull of the range of f. Write $g(n) = x$ iff $(J_\alpha, \in) \models \exists y \psi(n, x, y)$, where ψ is Π_1. Now suppose that there exists x in J_α such that $(J_\alpha, \in) \models \gamma(x, g(n_1), \ldots, g(n_k))$, where γ is Π_1. Then there exists $\langle x, x_1, y_1, \ldots, x_k, y_k\rangle$ in J_α such that the following Π_1 formula holds in $(J_\alpha, \in)$:

$$\gamma(x, x_1, \ldots, x_k) \wedge \psi(n_1, x_1, y_1) \wedge \ldots \wedge \psi(n_k, x_k, y_k).$$

By the definition of f, there exists such a sequence $\langle \bar{x}, \bar{x}_1, \bar{y}_1, \ldots, \bar{x}_k, \bar{y}_k\rangle$ in the range of f. Also $\bar{x}$ belongs to A and $\bar{x}_i$ equals $g(n_i)$ for each i, $1 \leq i \leq k$, and therefore $\gamma(\bar{x}, g(n_1), \ldots, g(n_k))$ holds in $(J_\alpha, \in)$. As $(A, \in)$ is Σ_0 elementary in $(J_\alpha, \in)$, it follows that $\gamma(\bar{x}, g(n_1), \ldots, g(n_k))$ holds in $(A, \in)$ for some $\bar{x}$, proving that $(A, \in)$ is Σ_2 elementary in $(J_\alpha, \in)$.

Finally, every element of J_α is countable in $(J_\alpha, \in)$, as otherwise there would be a Σ less than α such that some $\zeta < \Sigma$ is Σ_2 stable in Σ. It follows that A is transitive, as by Σ_1 elementarity, A contains an injection of any of its elements into ω. We have assumed that α is less than Σ, so in fact A equals all of J_α, and therefore there is a partial function g from ω onto J_α which is Σ_2 definable over $(J_\alpha, \in)$ without parameter, as desired. □

Now we are ready to describe the Theory Machine. When we say that the machine writes a theory T on its output tape at stage α, we mean that at stage α, the n-th cell of the output tape has a 1 written in it iff the n-th sentence (via a fixed Gödel numbering) belongs to T. Now the Theory Machine runs as follows: On input 0, the machine uses the first ω^2 stages to ensure that the theory of $(J_0, \in)$ is written on the output tape at stage ω^2. (In fact the machine could arrange this in fewer stages, but we prefer for this to occur at stage ω^2). Inductively, suppose that the theory of $(J_\alpha, \in)$ is written on the output tape at stage $\omega^2 \times (\alpha + 1)$. If α is less than Σ, then by Lemma 2, the machine can compute a code for $(J_\alpha, \in)$ by stage $\omega^2 \times (\alpha+1) + \omega$. Then the machine uses the theory of $(J_\alpha, \in)$ to compute a code for $(J_{\alpha+1}, \in)$ by stage $\omega^2 \times (\alpha+1)+\omega+\omega$ and the next ω^2 steps to write the theory of $(J_{\alpha+1}, \in)$ on its output tape at stage $\omega^2 \times (\alpha+1) + \omega + \omega + \omega^2 = \omega^2 \times (\alpha+2)$. The machine must however never write a 0 in the n-th cell of its output tape (at a stage between $\omega^2 \times (\alpha+1)$ and $\omega^2 \times (\alpha+2)$) if the n-th sentence is true in both $(J_\alpha, \in)$ and $(J_{\alpha+1}, \in)$.

The last requirement ensures that at a stage $\omega^2 \times \lambda$, λ limit, what is written on the output tape is the liminf of the theories of the $(J_\alpha, \in)$, $\alpha < \lambda$, i.e. the theory $T = \{\varphi \mid \varphi \text{ is true in } (J_\alpha, \in) \text{ for sufficiently large } \alpha < \lambda\}$. By Lemma 1, the machine can compute the Σ_2 theory of $(J_\lambda, \in)$ by stage $(\omega^2 \times \lambda) + \omega$ and by Lemma 2 it can compute a code for $(J_\lambda, \in)$ by stage $(\omega^2 \times \lambda) + \omega + \omega$, if λ is less than Σ. Then the machine uses the next ω^2 stages to write the theory of $(J_\lambda, \in)$ on its output tape, again never writing a 0 in the n-th cell of its output tape if the n-th sentence belongs both to T and to the theory of $(J_\lambda, \in)$.

This completes the description of the Theory Machine. The machine is capable of writing the theory of $(J_\alpha, \in)$ on its output tape at stage $\omega^2 \times (\alpha+1)$ provided α is less than Σ. The following corollaries easily follow, where λ is the least Σ_1 stable in Σ and ζ is the least Σ_2 stable in Σ:

On input 0:

Every ITTM either halts or repeats itself by stage Σ.
There is a machine that first repeats itself at stage Σ.
The supremum of the halting times of ITTM's is λ.
The reals that appear on the output tape of an ITTM are the reals in $J_\Sigma = L_\Sigma$.
The reals that appear on the output tape of a halting ITTM are the reals in $J_\lambda = L_\lambda$.
The reals that appear on the output tape of an ITTM from some stage onwards are the reals in $J_\zeta = L_\zeta$.

Also, if Σ^x, ζ^x, λ^x are the relativisations of Σ, ζ, λ to the real x:

A is an ITTM-semirecursive set of reals iff for some Σ_1 formula φ, we have: x belongs to A iff $L_{\lambda^x}[x] \vDash \varphi(x)$.

One can say a bit more about the halting times of ITTM's. Say that α is an *infinite power of* ω iff it is of the form ω^β, where β is infinite. An *infinite power of* ω *interval* is an interval $[\alpha, \beta)$ where $\alpha < \beta$ are adjacent infinite powers of ω. For any sentence φ of set theory let $\alpha(\varphi)$ denote the least α, if any, such that $(J_\alpha, \in)$ satisfies φ.

Corollary 3. *Let I be an infinite power of ω interval. Then the following are equivalent.*
i. I contains the halting time of an ITTM.
ii. I is below Σ and contains $\alpha(\varphi)$ for some sentence φ.

Proof. Suppose that α is the halting time of an ITTM. Then this can be expressed in $(J_{\alpha+1}, \in)$ and therefore $\alpha + 1 = \alpha(\varphi)$ for some φ. Conversely, suppose that $\alpha = \alpha(\varphi)$ for some φ and α is less than Σ. Then there is an ITTM that imitates the Theory Machine but halts when it sees that φ is true in $(J_\alpha, \in)$, at a stage less than $\omega^2 \times (\alpha+1) + \omega^2 = \omega^2 \times (\alpha+2)$. As the latter is less than the least infinite power of ω greater than α, it follows that $\alpha(\varphi)$ and α belong to the same infinite power of ω interval. $\square$

The previous corollary easily yields results about gaps in the set of halting times of ITTM's.

Our second observation concerns Γ-singletons, where Γ is the lightface pointclass of semirecursive sets of reals.

Theorem 4. *Suppose that x is a Γ-singleton, i.e., $\{x\}$ belongs to Γ. Then x is an element of L_{λ^x}.*

Proof. Let x be the unique x such that $L_{\lambda^x}[x] \vDash \varphi(x)$, where φ is Σ_1. Let c be a real which is generic over L_{Σ^x} for the Lévy collapse of λ^x to ω. By absoluteness, there is a real y in $L_{\Sigma^x}[c]$ such that $\varphi(y)$ holds in $L_{\lambda^x}[y]$ and λ^x is less than Σ^y. It follows that $\varphi(y)$ holds in $L_{\Sigma^y}[y]$, therefore in $L_{\lambda^y}[y]$ and therefore y equals x. As c is an arbitrary generic code for λ^x, x belongs to L_{Σ^x} and therefore to L_{λ^x}. □

Corollary 5. *The ITTM-degrees of Γ-singletons are wellordered in ordertype δ^1_2, the supremum of the lengths of Δ^1_2 wellorderings of ω, with successor given by ITTM-jump.*

Proof. If $\lambda^x \leq \lambda^y$ and x is a Γ-singleton then x belongs to L_{λ^y} and therefore is recursive in y. If $\lambda^x < \lambda^y$ then as the ITTM-jump of x is definable over $L_{\lambda^x}[x] = L_{\lambda^x}$, it follows that the ITTM-jump of x is recursive in y. The Γ-singletons include the Π^1_1-singletons, which are cofinal in $L_{\delta^1_2}$, and therefore the length of the wellordering of the ITTM-degrees of Γ-singletons is also δ^1_2. □

Remarks. i. In fact the ITTM-degrees of Δ-singletons are cofinal in those of the Γ-singletons, where Δ is the lightface pointclass of recursive sets of reals. This is because each Π^1_1-singleton is a Δ-singleton.
ii. There are reals with ITTM-degree incomparable with $0'$ = the ITTM-jump of 0; for example, consider a real Cohen generic over L_Σ. But this cannot happen for reals in L_Σ, as such a real x belongs to L_{λ^x} and therefore is either ITTM-recursive or ITTM above $0'$. By using Sacks forcing one obtain a continuum of minimal ITTM-degrees over 0.

α-Recursion Theory and Ordinal Computability

Peter Koepke
University of Bonn

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.

0. 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 $\Sigma_1(L_\alpha)$, 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 $\Delta_1(L_\alpha)$. Sacks discusses the "computational character" of $\Sigma_1(L_\alpha)$-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.].

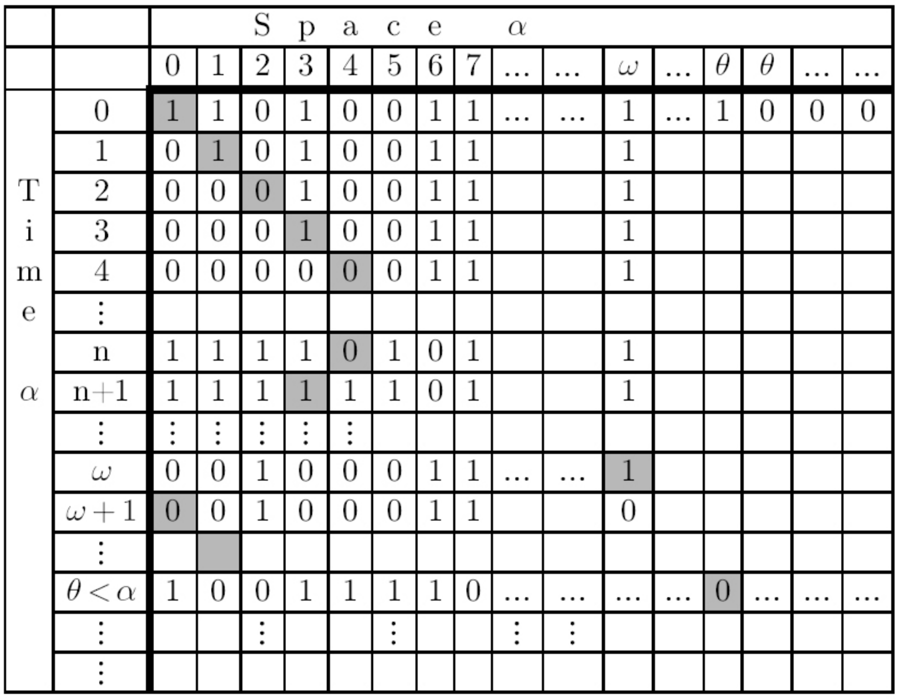
In this note we address the question whether it is possible to base α-recursion theory on some idealized computational model.

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 $\alpha \times \alpha$ spacetime: 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*

(1) *A is α-recursive iff A is α-computable.*
(2) *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.

1. α -Turing Machines

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

Definition 3.

(1) *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.

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

(3) *For a program P let*

$$\mathrm{states}(P) = \{s | (s, c, c', m, s') \in P\}$$

be the set of program states.

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

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

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

(1) *θ is a successor ordinal $< \alpha$ or $\theta = \alpha$; θ is the* length *of the computation.*
(2) *$S(0) = H(0) = 0$; the machine starts in state 0 with head position 0.*
(3) *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.*
(4) *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:*

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

(5) *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 $\chi_A : \alpha \to 2$, $\chi_x(\xi) = 1$ iff $\xi \in A$.

Definition 5. *A partial function $F : \alpha \rightharpoonup \alpha$ 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* $F : \alpha \rightharpoonup 2$.

2. 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:*

(1) S, H, T *is the* α*-computation by* P *with input* $T(0)$ *as computed* in the model $(L_\alpha, \in)$.
(2) S, H, T *are* Σ_1*-definable in* $(L_\alpha, \in)$ *from parameters.*
(3) *If* $A \subseteq \alpha$ *is* α*-recursively enumerable then it is* $\Sigma_1(L_\alpha)$ *in parameters.*
(4) *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.

3. 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 $\{v_i \in v_j | \varphi\}$ is a term.

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

- $\mathrm{free}(v_i) = \{v_i\}, \mathrm{bound}(v_i) = \emptyset$;
- $\mathrm{free}(s = t) = \mathrm{free}(s \in t) = \mathrm{free}(s) \cup \mathrm{free}(t)$;
- $\mathrm{bound}(s = t) = \mathrm{bound}(s \in t) = \mathrm{bound}(s) \cup \mathrm{bound}(t)$;

- $\mathrm{free}(\neg\varphi) = \mathrm{free}(\varphi), \mathrm{bound}(\neg\varphi) = \mathrm{bound}(\varphi)$;
- $\mathrm{free}((\varphi \wedge \psi)) = \mathrm{free}(\varphi) \cup \mathrm{free}(\psi), \mathrm{bound}((\varphi \wedge \psi)) = \mathrm{bound}(\varphi) \cup \mathrm{bound}(\psi)$;
- $\mathrm{free}(\forall v_i \in v_j \varphi) = \mathrm{free}(\exists v_i \in v_j \varphi) = \mathrm{free}(\{v_i \in v_j | \varphi\}) = (\mathrm{free}(\varphi) \cup \{v_j\}) \setminus \{v_i\}$;
- $\mathrm{bound}(\forall v_i \in v_j \varphi) = \mathrm{bound}(\exists v_i \in v_j \varphi) = \mathrm{bound}(\{v_i \in v_j | \varphi\}) =$
  $= \mathrm{bound}(\varphi) \cup \{v_i\}$.

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 and 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

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

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 $W \restriction L_\alpha$ 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 $A \subseteq \alpha$:

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

where φ is a bounded formulas. This is equivalent to

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

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. □

4. 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.

(1) *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 $\gamma < \alpha$*

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

and

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

(2) *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 $K \in L_\alpha$*

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

and

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

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 and part (1) 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.

References

[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.

Aspects of Ordinal Computability

Discussion session

Tuesday, 16:30-17:30

This is a list of topics covered in the open discussion session.

1. Machine Models

1.1. Basic Models of Computation

1.1.1. Turing Machines

1.1.2. Register Machines

1.1.3. Stack Machines

1.1.4. Random Access Machines

1.1.5. Determinism / Non-determinism

1.1.6. ...

1.2. Resources provided for Computations

1.2.1. Space ω / space $\alpha \in \text{Ord}$ / space Ord

1.2.2. Time ω / space $\alpha \in \text{Ord}$ / space Ord

1.2.3. Resources dependent on the input size (Complexities)

1.2.4. ...

1.3. Specific Issues

1.3.1. Write once / write finitely often

1.3.2. Monotone headpositions / Register contents / ...

1.3.2. Tapes / Alphabets / Usage of tapes

1.3.3. Limit rules

1.3.3. ...

2. Programs

2.1. Structuring of programs

3. Relations to other theories

3.1. Descriptive Set Theory

3.2. Constructibility Theory

3.2.1. Coding L by ordinal machines

3.2.2. Applications to L

3.2.2.1. Construction of Silver machines by ordinal computability

3.2.2.2. Proof of $\Box$ via Silver machines

3.2.2.3. Reconstructing L-constructions as ordinal computations

3.3. Recursion Theory

3.3.1. α-Recursion Theory

3.3.1.1. Reconstructing results by ordinal computations

3.3.2. E-recursion

3.3.3. Aiming at weak machine models

3.4. Proof Theory

3.4.1. Background theories for ordinal computations

3.4.1.1. Admissible set theory

3.4.1.2. Constructive background theories

3.4.2. Self-creation and -verification of ordinals required for computations

3.5. Model Theory

3.5.1. Infinite Time Computable Model Theory

3.5.2. Computable Equivalence Relations

3.6. Borel Programming

4. Relations to Physics

4.1. Hypercomputations