The Length of Infinite Time Turing Machine Computations*

P. D. Welch

Abstract

We show that the halting times of infinite time Turing Machines (considered as ordinals coded by sets of integers) are themselves all capable of being halting outputs of such machines. This gives a clarification of the nature of "supertasks" or infinite time computations. The proof further yields that the class of sets coded by outputs of halting computations coincides with a level of Gödel's constructible hierarchy: namely that of L_λ where λ is the supremum of halting times. A number of other open questions are thereby answered.

1 Introduction: Infinite Time Turing Machines

Hamkins and Lewis in [4] give an account of the construction of these machines (first developed by Kidder and Hamkins in 1989) and develop the basic theory of this notion of computability. The reader should refer to this paper for a clear and full account of their basic properties, from which all the results and definitions of this introduction are taken. But to summarise: an infinite time Turing machine has the hardware of an ordinary Turing machine, except that transfinite stages of computation are allowed (the action at successor stages being just that specified as usual by a finite program with a finite number of possible states). The action at transfinite limit stages ν of "time" is very definite: the machine head always returns to the same specific cell on the tape, and enters a special limit state. The value in the n'th cell C_n at time ν (denoted $C_i(\nu) \in \{0,1\}$) is specified as the limsup of the values at all previous "times". This specifies precisely a behaviour of a machine, and like ordinary Turing machines, such a device may "halt" with a specific output or may loop forever. The infinite amount of time available means however that such machines can accept an infinite string of integers as input (and can output the same). This implies that a machine can decide whether an input is a code of a wellfounded relation and indeed much more. Programs differ from regular Turing programs only in the addition of a named limit stage state q_L. We may thus as usual consider an enumeration $\langle P_n \,|\, n \in \omega\rangle$ of all programs, coded by integers. A straightforward cardinality/cofinality argument shows that for any computation, it has "settled down" (i.e., is either looping or has halted) by some countable ordinal length of time.

The relation $\phi_p^f(x)\downarrow$ (in α-steps) ("the p'th program halts on input $x \in {}^\omega 2$ using oracle $f \in {}^\omega 2$" (additionally, "in α-steps")) is Δ^1_2 relation of f, p, x (and α). (We consider input strings $f, x, \ldots$ as reals, or as members of Cantor space ${}^\omega 2$.) Similarly $\phi_p^f(x)\uparrow$ is also Δ^1_2. (It is only necessary to have any or all wellordered sequence(s) of ω-sequences of cell values coding the course of a looping computation to see this.) (*cf.* [4] 2.5)(i) .)

If $\mathcal{W} =_{df} \{x \in {}^\omega 2 \,|\, x$ is the output of $\phi_p(0)\downarrow\}$, then the "writable" ordinals, (those ordinals α so that there is $x \in \mathcal{W} \cap WO$ with $||x|| = \alpha$) form an initial segment of the countable

*AMS Class: 03D10 03D60 03E45

ordinals, λ. (*cf.* [4] 3.7, 8.3.) We use the notation $\mathcal{W}^f, \lambda^f \dots$ for the notions relativised to a real. They define the *"clockable"* ordinals as those α so that for some p $\phi_p(0)\downarrow$ in exactly α steps. They let $\gamma =_{df} \sup\{\alpha \mid \alpha \text{ clockable}\}$. Not every ordinal less than γ is clockable however (admissibles for example are not clockable, and there are arbitrarily long gaps below γ without clockable ordinals), but they show that the order type of the "clockables" is precisely λ. Perhaps the main open question from [4] was:

Question 1 ([4] Q3.1) Is $\lambda < \gamma$?

A negative answer to this question about the relationship between ordinal times of computations and possible ordinal outputs would simplify several proofs. We shall show this is indeed the case:

Theorem 1.1 $\lambda = \gamma$ *(and by relativisation* $\forall f \in {}^{\omega}2 \quad \lambda^f = \gamma^f$*).*

The ordinal λ was shown in [4] to have fairly considerable closure properties, see the Fact below, but moreover, that λ was indescribable by any semi-decidable property. This meant that if x was a code for L_λ, then for any property P semi-decidable by such machines, which was true of x, (meaning that there was a program number q, so that $\phi_q(x)\downarrow 1$ if $P(x)$ held, undefined otherwise) then there was a y for which $P(y)$ was true, and y coded L_α for an $\alpha < \lambda$. They asked:

Question 2 ([4], Q8.4) Is λ the least ordinal indescribable by semi-decidable properties?

This is also the case:

Lemma 3.7 Every ordinal less than λ is describable by decidable properties.

The class $\mathcal{EW}$ of *eventually writable* reals, are those reals that remain written on the output tape of any computation of the form $\phi_p(0)$ for any $p \in \omega$, after it has either halted, or has entered a permanent loop. (In the latter case we write $\phi_p(0)\uparrow x$.) In other words, $x \in \mathcal{EW}$ if for some p, and for all sufficiently large times ν, x is the real on the output tape of $\phi_p(0)$. The class $\mathcal{AW}$ of *accidentally writable* reals are those that appear on any tape at any stage at all of any computation of the form $\phi_p(0)$. Clearly $\mathcal{E} \subseteq \mathcal{EW} \subseteq \mathcal{AW}$. [4] shows the inclusions are strict, indeed that $\lambda < \zeta < \Sigma$ where:

Definition 1.1 $\zeta =_{df} \sup\{||x|| \mid x \in \mathcal{EW} \cap WO\}$; $\Sigma =_{df} \sup\{||x|| \mid x \in \mathcal{AW} \cap WO\}$.

Fact ([4] Thm. 8.3, 8.6) The classes $H(\lambda), H(\zeta)$ whose elements are coded by reals in $\mathcal{W}$ (respectively $\mathcal{EW}$) are transitive admissible sets, closed under hyperjump (and more).

It is natural to ask:

Question 3 Is $H(\lambda) = L_\lambda$? Is $H(\zeta) = L_\zeta$?

By virtue of the Main Proposition in §2 establishing Theorem 1.1, we shall see:

Corollary 3.3(i) $H(\lambda) = L_\lambda$. (again for any $f \in {}^\omega 2 \quad H^f(\lambda^f) = L_{\lambda^f}[f]$).
(ii) (Corollary 3.1) $H(\zeta) = L_\zeta$; and similarly (Corollary 3.5) $H(\Sigma) = L_\Sigma$.

(We have defined $H(\Sigma)$ analogously to $H(\zeta)$. The relativised notions are also defined *mutatis mutandis*.) It might be thought that L_Σ is an admissible set, but this is not so (Corollary 3.4 below).

They consider a natural notion of jump (the "strong jump") $A^\blacktriangledown$ for a set of reals A (defined below). Let $\mathcal{C} =_{df} \{y \in {}^\omega 2 \mid y \text{ is computable from } O^\blacktriangledown\}$. They ask:

Question 4 ([4] 6.16) Is $\mathcal{AW} \subseteq \mathcal{C}$? Or *vice versa*?

Corollary 3.6 $\mathcal{AW} \subsetneq \mathcal{C}$.

Theorem 1.1 is proven at the end of section 2. The other corollaries are deduced in section 3. For any notions of constructible sets, the reader may find them in [3]. For the basic structure of Turing Machines, see, for example [2]; for admissibility theory either [3] or [1].

We should like to warmly thank Joel Hamkins for introducing us to the subject by asking us some of these questions, giving some patient explanations, and for his suggestions.

2 The Main Proposition

The Corollaries and the Theorem 1.1 mentioned are all those to (the proof of) the main proposition below, concerning the stabilization times of the values of cells on tapes during computations of the form $\phi_p(0)$. For the machine computing $\phi_p(0)$ let all the cells on the tapes be enumerated as C_i^p (for $i < \omega$); and let the value of the cell $C_i = C_i^p$ at time ν be $C_i(\nu) \in \{0, 1\}$.

Definition 2.1 *For any time ν let $\delta_i(\nu) = \sup\{\nu' < \nu \mid \nu' = 0 \vee C_i(\nu') \neq C_i(\nu' + 1)\} \leq \nu$. For Lim($\nu$) we say C_i has* (locally) stabilized *at stage ν, if $\delta_i(\nu) < \nu$.*

Of course, at a later stage $\overline{\nu} > \nu$ C_i may become destabilized. Some cells may never be stabilized. The main proposition says that for those cells that have stabilized at Σ by a time that is the length of an accidentally writable ordinal, then in fact that ordinal is eventually writable.

Main Proposition $\forall i\ \delta_i(\Sigma) < \Sigma \longrightarrow \delta_i(\Sigma) < \zeta$.

Proof: We think of a universal machine $\mathcal{U}$ as just another of [4]'s machines, and such machines have their tape divided into three parts: 3 ω sequences of cells, one for input values, one for scratch computation, and one for output values. $\mathcal{U}$ is the machine that simulates, by cutting up its scratch tape into infinitely many slices, all the computations $\phi_q(0)$ for each q simultaneously.

Definition 2.2 *Let $\sigma(\nu) =_{df} \Sigma\{||x|| : x \in \mathcal{AW} \cap WO \wedge x$ is on one of $\mathcal{U}$'s tapes at stage $\nu\}$.*

$\sigma(\nu)$ is thus a "grand sum" of everything in sight on $\mathcal{U}$'s tapes at stage ν. We shall use this grand sum as providing us with an ordinal along which we may perform certain computations.

One should note

(1) As $\sigma(\nu) \in \mathcal{AW}$, it must be the case that, by the definition of $\sigma(\nu)$, that as $\mathcal{U}$ progresses $\sigma(\nu)$ becomes unbounded in Σ.
(2) Once ν is sufficiently large, all the tapes of $\mathcal{U}$ that are destined to be eventually written to, will have received those eventual values; hence at some earliest moment $\overline{\nu}$ say, $\sigma(\overline{\nu}) \geq \zeta$, and this will be true at all later stages of time after $\overline{\nu}$.
(3) On the other hand it is easy to see that $\sigma(\nu)$ never becomes a fixed value (if so we could arrange it to be written to an output tape of some machine and then we should have $\sigma(\nu) \in \mathcal{EW}$!). This will contradict (2).

We describe now a machine $\mathcal{M} = \mathcal{M}(i,p)$, whose scratch tape is divided into at least three infinite parts S_0, S_1, S_2. $\mathcal{M}$ is simulating the universal program $\mathcal{U}$ on S_0 (by a further recursive slicing up of S_0 into infinitely many pieces). Let T denote M's output tape. $\mathcal{M}$ is to behave as follows at time ν.
(i) $\mathcal{M}$ computes a current value for "$\sigma(\nu)$" by looking at the cells of its "$\mathcal{U}$";
(ii) on its scratch tape S_1, it writes a code for the whole course-of-computation of the first $\sigma(\nu)$ steps of $\phi_p(0)$ (it may do this easily given the infinite amount of time available and using the computability of arithmetic properties of reals enjoyed by these machines (see [4] 4.9 for more details - indeed if we wished we could consider an $\mathcal{M}$ which was writing a code for $L_{\sigma(\nu)}$ on S_1 and inspecting the run of $(\phi_p(0))_{L_{\sigma(\nu)}}$).
Let $\delta = \delta_i(\sigma(\nu))$ the current stabilization value for cell C_i.
(iii) Then $\mathcal{M}$ inspects T: the contents of T is a sequence $x \in {}^{\omega}2$, it tests (using S_2) if $x \in WO$ (which it can do as "y codes a wellorder" is decidable by such machines); if so, $\mathcal{M}$ runs $||x||$-many steps of $\phi_p(0)$, and computes $\delta_i(||x||)$; if $\delta_i(||x||) \neq ||x||$, or if x was not a code for a wellorder, or if $||x|| < \delta$, then x on T is overwritten by (a code for) δ (which $\mathcal{M}$ has available); otherwise T is left untouched.
(iv) $\mathcal{M}$ runs another ω steps in $\mathcal{U}$'s computations. And then returns to stage (i) now at time $\nu + 1$.

Claim If $\delta_i(\Sigma) < \Sigma$ then $\delta_i(\Sigma)$ is the content of T, and is in $\mathcal{EW}$, and hence $\delta_i(\Sigma) < \zeta$.

Proof: Let $\tilde{\delta} =_{df} \delta_i(\Sigma)$ and suppose first $\tilde{\delta} < \Sigma$. At various stages ν, the current $\sigma(\nu)$ is calculated, and $\delta_i(\sigma(\nu))$ written to T. When $\mathcal{M}$ inspects T at (iii) above, it may find T's contents x do indeed code some wellorder, but this ordinal may be bogus arising out of some "chance" operation of the machine by taking limsups in the cells at some earlier limit stage. Or it might find that the contents of T is just garbage resulting from infinitely many previous overwrites, and the limsup operation. But in any case the x that is left on T satisfies $\delta_i(||x||) = ||x|| \geq \delta$.

Eventually however (by (2) above) $\sigma(\nu) > \zeta$ and perhaps later by (1) becomes sufficiently large below Σ that it is greater than the true $\tilde{\delta}$. At this time $\tilde{\delta}$ is written to T. But note the content of T remains fixed hereafter: there will not be any reason to overwrite it, irrespective of the fact that the value of $\sigma(\nu)$ fluctuates, and may very well be less than $\tilde{\delta}$ at some future time. Hence $\tilde{\delta} \in \mathcal{EW}$.

Q.E.D.Claim & Proposition

We could show conversely that $\delta_i(\Sigma) = \Sigma$ implies the contents of T is not in $\mathcal{EW}$, but we have no need for this direction.

Corollary 2.1 *Theorem 1.1 holds, i.e.* $\lambda = \gamma$.

Proof: Fix an arbitrary $p \in \omega$. We have just seen that for any $i < \omega$ if C_i is stabilized at Σ, then the point of stabilization is below ζ. The value of all the other, unstabilized, cells is perforce 1, by the limsup operation, at times ζ *and* Σ. Hence a), the "snapshot" of the state of the computation (meaning the distribution of 0's and 1's in the cells C_i^p, the state of the machine, etc.) is exactly the same at time ζ and Σ; *and moreover* b), for no $i < \omega$ for which $C_i(\zeta) = 0$, is $C_i(\xi) = 1$ for any $\zeta < \xi < \Sigma$. The conclusion to be drawn from b) is that either $\phi_p(0)\downarrow$, and has halted at time $\mu < \zeta$, or the snapshots at ζ and Σ are from a permanently looping cycle. Thus $\phi_p(0)\uparrow$, and has started to loop, and at the latest by time ζ. In the former case we see that μ is "clockable", i.e. $\mu < \gamma$. If this is so:
Claim $\mu < \lambda$.
Proof: Suppose $q \in \omega$ is a program number so that $\phi_q(0)$ eventually writes an ordinal $\mu' \geq \mu$ on its output tape. We specify a machine $\mathcal{N}$ that will witness that $\mu \in \mathcal{W}$. $\mathcal{N}$ simulates on one scratch tape $\phi_q(0)$, and using the ordinals being written by ϕ_q to its (simulated) output tape, $\mathcal{N}$ also simulates runs of computation of $\phi_p(0)$ along these ordinals, on the other scratch tape. Eventually $\phi_q(0)$ produces an ordinal greater than μ; then running $\phi_p(0)$ $\mathcal{N}$ sees that $\phi_p(0)\downarrow$, and $\mathcal{N}$ can write the ordinal length of that computation, μ, to its output tape, and halt. Hence $\mu < \lambda$. **Q.E.D.Claim**
Hence every "clockable" ordinal is "writable". **Q.E.D.**

3 The Corollaries

As the relation $\phi_p^f(x)\downarrow y$ and $\phi_p^f(x)\uparrow$ are both Δ_2^1 relations, the outputs of the form $\phi_p(0)$ are all constructible sets. If η is one of ζ, λ, Σ it is natural to conjecture that $H(\eta) = L_\eta$. [4] show that $H(\zeta), H(\lambda)$ are transitive KP-models, closed under hyperjump, and much more.

Corollary 3.1 $H(\zeta) = L_\zeta$

Proof: Consider running all computations $\phi_p(0)$ inside the model L_ζ. By absoluteness considerations, this produces exactly the same course of computation as running in V. The next claim also relates the ordinals output to length of time of computation, but this time for $\mathcal{EW}$ ordinals (and sets). It says that $\overline{\nu} = \zeta$ where $\overline{\nu}$ is that of (2) of the Main Proposition.
Claim All $x \in \mathcal{EW}$ have appeared, never to be erased on their output tapes by time ζ.
Proof: This is because at the very worst, the stage ζ "snapshot" is the very first of what will be a permanently looping cycle, by the argument of Cor. 2.1. If $x \in \mathcal{EW}$ is such a code eventually written by $\phi_p(0)$, then if x is eventually written at some stage τ, then $\tau \leq \zeta$. If $\tau < \zeta$, then we are done. So suppose x is the content of the output tape of $\phi_p(0)$ only at time ζ, that is suppose $\tau = \zeta$. Then x is definable over L_ζ and hence $x \in L_{\zeta+1}$.

But now we could consider running the universal machine, and we should see that if $x \in \mathcal{EW}$ and $\phi_p(0)\uparrow x$ or $\phi_p(0)\downarrow x$, we could inspect $L_{\sigma(\nu)}$ and arrange that the constructible rank of the current contents of the (simulated) output tape of $\phi_p(0)$ (if it exists) be continually written to an output tape. If $x \in L_\alpha$, for some least α then we shall see that $\alpha \in \mathcal{EW}$. Hence $\alpha < \zeta$! A contradiction. **Q.E.D.**

Lemma 3.2 $H(\lambda) \prec_{\Sigma_1} H(\zeta) \prec_{\Sigma_1} H(\Sigma)$.

Proof: If $H(\Sigma) \models \exists x \psi(x, \vec{y})$ where $\vec{y} \in [H(\lambda)]^{<\omega}$, then we arrange a machine to simulate $\mathcal{U}$, and if x appears on the tape of $\phi_p(0)$ at some stage, we ask that the machine look for a suitable x *after* all the relevant computations finding $\vec{y}$ have halted. Once it has found such an x the machine will halt with x as output. Hence there is such an $x \in \mathcal{W}$. Hence $H(\lambda) \prec_{\Sigma_1} H(\Sigma)$, and *a fortiori* $H(\lambda) \prec_{\Sigma_1} H(\zeta)$. That $H(\zeta) \prec_{\Sigma_1} H(\Sigma)$ is similar: if now $\vec{y}$ is merely in $\mathcal{EW}$, *via* the computations whose outputs are approximating $\vec{y}$, we just keep looking for a witness x for the "current" approximations to $\vec{y}$, and writing it to an output tape. Eventually this settles down, since the approximations to $\vec{y}$ eventually become $\vec{y}$. **Q.E.D.**

Corollary 3.3 $H(\lambda) = L_\lambda$.

Proof: By the last Corollary and Lemma. **Q.E.D.**

Corollary 3.4 *$H(\Sigma)$ is also a transitive set, with $H(\Sigma) \models$ "$\forall x \exists \mathcal{A}(x \in \mathcal{A} \wedge \mathcal{A}$ is admissible)"; however Σ is not admissible.*

Proof: We first note:
Claim (i) If $\phi_p(0)\uparrow$ and $\exists \nu < \zeta$ so that ν is the first stage in a "loop" in its computation, then $L_\zeta \models$ "$\phi_p(0)\uparrow$".
(ii) $\exists q_0\, \phi_{q_0}(0)\uparrow$ but $L_\zeta \not\models$ "$\phi_{q_0}(0)\uparrow$ "; indeed $L_\Sigma \not\models$ "$\phi_{q_0}(0)\uparrow$ "
Proof: By the main proposition ζ and Σ are always identical looping stages in any such computation that does not halt. For (i) if ζ is not the first, then using Lemma 3.2 we see $\forall \nu' \nu < \nu' < \zeta, \exists \mu < \zeta$ with $\nu' < \mu$ and the snapshot at μ is identical to that at ν. Thus $\phi_p(0)$ has looped infinitely many times below ζ. For (ii): there must be such a q_0 otherwise we could solve the halting problem for such machines: given $p \in \omega$, we just run $\mathcal{U}$ until we find a wellorder of sufficient length τ so that $L_\tau \models$ "$\phi_p(0)\downarrow$ or $\phi_p(0)\uparrow$"; we then output 0 or 1 accordingly. (We have already remarked that a code for L_τ is readily computed from one for τ, and hence the first order theory of L_τ is decidable from a code for τ.) This contradicts the undecidability of the halting problem for such machines ([4] Thm 4.1). **Q.E.D.Claim**

We consider the inadmissibility of Σ first. Fix q_0 as in (ii) of the Claim. By the Main Proposition $I = \{i \mid C_i^{q_0}$ has stabilized by stage $\Sigma\}$ is definable over L_ζ. Suppose for a contradiction that Σ were admissible. We just use a simple Π_2-reflection argument to reflect the snapshot occurring at Σ to an ordinal below. For those $i \notin I$ (and there will be infinitely many such) there is $D_i \in \Delta_1(L_\Sigma)$ with D_i cub in Σ, and so that $\alpha \in D_i \longrightarrow \delta_i(\alpha) = \alpha$. But then $D =_{df} \bigcap_{i \notin I} D_i$ is easily seen to be cub beneath Σ and is $\Delta_1(L_\Sigma)$ (in the parameter I). Hence $otp(D)$ is itself Σ. But then if $\overline{\alpha} < \Sigma$ is any limit point of D, the snapshot at $\overline{\alpha}$ is again just that at ζ. Hence the ω'th limit point of D is a stage $\alpha^* < \Sigma$ with infinitely many loops in the calculation of $\phi_{q_0}(0)$ below α^*. Hence L_Σ sees that $\phi_{q_0}(0)$ is looping, contrary to the choice of q_0. Hence Σ is inadmissible.

Now we consider $H(\Sigma)$. Using the $\sigma(\nu)$ of Def.2.2 and $\mathcal{U}$, we can organise a machine so that it may compute the current value of $\sigma(\nu)$ and write on scratch tape a code for $L_\beta[x]$, if $\beta =_{df} ||\{e\}^{\sigma(\nu)}|| \in WO$, for any real x on any tape of $\mathcal{U}$ at time ν.

This shows that for any such x, $L_{\omega_{1ck}^x}[x] \subseteq H(\Sigma)$. As Σ itself is inadmissible, we must have that $L_{\omega_{1ck}^x}[x] \in H(\Sigma)$.

Q.E.D.

Corollary 3.5 *$H(\Sigma) = L_\Sigma$.*

Proof: Run the universal machine $\mathcal{U}$ inside L_Σ. Argue as in 3.1. **Q.E.D.**

The following is the notion of strong jump that [4] defines on sets of reals.

Definition 3.1 *For $f, \in {}^\omega 2, A \subseteq {}^\omega 2$, $A^\blacktriangledown =_{df} H^A = \{(p,x) \mid \phi_p^A(x)\downarrow\}$.*

There is a wealth of information in [4] on the degree structure, and on jump operators on degrees of reals, and of sets of reals.

Corollary 3.6 *$\mathcal{AW} \subsetneq \mathcal{C} =_{df} \{y \in {}^\omega 2 \mid y$ is computable from $O^\blacktriangledown\}$.*

Proof: To see this let q_0 be as in the Claim (ii) of Cor. 3.4.
(1) $\exists z_0 \in \mathcal{C} \cap WO \wedge ||z_0|| = \zeta$.
Proof: We run $\mathcal{U}$ and looking at $L_{\sigma(\nu)}$ we ask of $0^\blacktriangledown$ if any of the snapshots of $\phi_{q_0} \cap L_{\sigma(\nu)}$ are going to repeat. ($0^\blacktriangledown$ can answer these, *c.f.* the argument of 6.5 of [4]. Given a snapshot s in the computation of $\phi_{q_0}(0)$, we use this s as input, to a computation which itself simulates $\phi_{q_0}(0)$, and is designed to halt on the recurrence of the snapshot s. With $0^\blacktriangledown$ as oracle, we shall be told the answer). For the least $\overline{\zeta} < \sigma(\nu)$ such that the $\overline{\zeta}$'th snapshot repeats, then we can output z_0, a code for $\overline{\zeta}$. But $\overline{\zeta} = \zeta$! **Q.E.D.(1)**
(2) $\Sigma < \lambda^{z_0}$.
Proof: From z_0 we can write a code for $L_{\zeta+1}$ and hence a code y_0 for the snapshot of $\phi_{q_0}(0)$ at time ζ. We then run a machine which halts when it finds the next appearance of the snapshot coded by y_0. This latter machine can be construed as a computation of the form $\phi_r^{z_0}(0)$ for some z_0-program number r. Clearly this latter computation cannot loop, but halts. And the length of this computation is clearly no less than the amount of time needed to find the next repeat of the snapshot s, in other words: Σ. Hence Σ is less than a "z_0-clockable". By the relativised Main Proposition, it is then also in $\mathcal{W}^{z_0}$. **Q.E.D.(2)**

As $z_0 \in \mathcal{C}$, by (2) so is Σ; hence so is $L_\Sigma = H(\Sigma)$. Hence the inclusion is strict. **Q.E.D.**

We turn finally to answering Question 2 on the semi-decidable indescribability of λ.

Lemma 3.7 *Every $\alpha < \lambda$ is describable by a decidable property.*

Proof: Let r be chosen so that $\phi_r(0)\downarrow z$ where $z \in \mathcal{W}$ is a code for L_α. (Such exists as $L_\lambda \subseteq H(\lambda)$). Let $P(x)$ be the property "x codes a set $E_x \subseteq \omega \times \omega \wedge \langle \omega, E_x \rangle \cong \langle \omega, z \rangle$ where $z = \phi_r(0)$." Clearly then, there can be no y coding some L_β for $\beta < \alpha$ and $P(y)$. P is clearly decidable. **Q.E.D.**

We finally remark that it is possible to characterise the ordinal ζ as the smallest admissible ordinal, such that L_ζ has a transitive Σ_2-elementary end-extension. This characterises the infinite time decidable reals. Further the set $S = \{p \mid \phi_p(0)\uparrow x \in \mathcal{EW}\}$ is a complete $\Sigma_2(L_\zeta)$ set of integers. There are naturally occurring questions (and some answers) concerning the appropriate notion of minimal degree, and whether minimal upper bounds for countable sets of infinite time degrees exist. These results will appear (or have appeared) in [5] and [6].

References

[1] K.J.Barwise, *Admissible Sets and Structures*, Perspectives in Mathematical Logic, 1975, Springer Verlag, Berlin, Heidelberg.

[2] G. Boolos & R.Jeffrey, *Computability and Logic*, 1980, Cambridge University Press.

[3] K. Devlin, *Constructibility*, Perspectives in Mathematical Logic, 1984, Springer Verlag, Berlin, Heidelberg.

[4] J.D. Hamkins & A. Lewis, *Infinite Time Turing Machines*, to appear in the J. for Symbolic Logic.

[5] P.D.Welch, *Eventually Infinite Time Turing degrees: Infinite Time Decidable Reals*, to appear in the Journal for Symbolic Logic.

[6] P.D.Welch, *Minimality arguments in the Infinite Time Turing degrees*, "Sets and Proofs", Ed. S.B.Cooper & J.K.Truss, L.M.S. Lecture Notes in Mathematics Series, April 1999, C.U.P., 12pp.

Graduate School of Science & Technology,)
Kobe University,
Rokko-dai, Nada-ku
Kobe 657,
Japan.
welch@kobe-u.ac.jp

School of Mathematics,
University of Bristol,
Bristol,
BS6 6BH.
England.
p.welch@bristol.ac.uk