$G_{\delta\sigma}$-games and generalized computation

P.D. Welch

School of Mathematics, University of Bristol,
Bristol, BS8 1TW, England

23.ix.15

Abstract

We show the equivalence between the existence of winning strategies for $G_{\delta\sigma}$ (also called Σ^0_3) games in Cantor or Baire space, and the existence of functions generalized-recursive in a higher type-2 functional. (Such recursions are associated with certain transfinite computational models.)

We show, *inter alia,* that the set of indices of convergent recursions in this sense is a complete $\partial\Sigma^0_3$ set: as paraphrase, the listing of those games at this level that are won by player I, essentially has the same information as the 'halting problem' for this notion of recursion.

Moreover the strategies for the first player in such games are recursive in this sense. We thereby establish the ordinal length of monotone $\partial\Sigma^0_3$-inductive operators, and characterise the first ordinal where such strategies are to be found in the constructible hierarchy. In summary:

Theorem (a) *The following sets are recursively isomorphic.*

(i) *The complete ittm-semi-recursive-in-*eJ *set,* H^{eJ};

(ii) *the* Σ_1*-theory of* $(L_{\eta_0}, \in)$, *where* η_0 *is the closure ordinal of* $\partial\Sigma^0_3$*-monotone inductions*;

(iii) *the complete* $\partial\Sigma^0_3$ *set of integers.*

(b) *The ittm-recursive-in-*eJ *sets of integers are precisely those of* L_{η_0}.

1 Introduction

The attempt to prove the determinacy of two person perfect information games (and the consequences of the existence of such winning strategies) has a long and fruitful history, starting with work of Banach and Mazur and continuing to the present. The work in the paper [20] was initially motivated by trying to see how the Π^1_3-theory of *arithmetical quasi-inductive definitions* fits in with other subsystems of second order number theory, in particular with the determinacy of Σ^0_3-sets. There it was shown, *inter alia*, that QI's - which were known to be formally equivalent with the most basic form of generalized computation to be introduced below - are not strong enough to compute strategies for Σ^0_3-games. What had been left open was a more precise discussion of the location of those strategies. We continue that discussion here. To give this research a context we shall also mention the results previously known in this area.

The argument in [21] explicitly extracts what was undeclared in the proof, a criterion for where exactly the strategies appear in the G del constructible L_α hierarchy. Whilst we have had this result for some while, the characterisation is somewhat unusual in that it is expressed in terms of the potential for such L_α to have certain kinds of ill-founded elementary end extensions, and is not so perspicuous. We had conjectured that certain kinds of illfounded-computation trees (defined by Lubarsky) should also characterize this ordinal. This we have verified, but now see that there is a bigger picture that connects the generalized recursion theory of the late 50's and early 60's of Kleene (v. [9]) of higher types with the determinacy of games at this level. To be clearer the connection is between the existence of winning strategies and the *generalization* of Kleene which is associated with a transfinite computational model of the so-called Infinite Time Turing machines of Hamkins and Kidder [5]. Kleene in [9] developed an equational calculus, itself evolving out of his analysis of the G del-Herbrand General Recursive Functions (on integers) fom the 1930's, but now enlarged for dealing with recursion in objects of finite type. (The set of natural numbers we denote by ω and they are of type 0; $f\colon a \to \omega$ is of type $k+1$ if a is of type k.) particular type-2 functional was that of the *ordinary jump* J, where

$$J(e,\vec{m},\vec{x}) = 1 \text{ if } \{e\}(\vec{m},\vec{x})\downarrow \text{ (meaning } \textit{converges}\text{, or } \textit{is defined}\text{)}$$
$$= 0 \text{ otherwise.}$$

Here $\vec{m}$ is a string of integers, and $\vec{x}$ a vector of functions $f:\omega\to\omega$ (thus a vector of objects of type 1) and $\{e\}$ a usual index of a recursive function. The function under discussion is $\{e\}$ which is given by a natural number index coding its formation. In this formalism the index set

$$H^J(e) \leftrightarrow \{e\}^J(e)\downarrow$$

is a complete semi-recursive (in J) set of integers, and Kleene showed that this is in turn a complete Π^1_1 set of integers. Further he showed that the *J-recursive sets of integers*, *i.e.* those sets R for which

$$R(n) \leftrightarrow \{e\}^J(n)\downarrow 1 \wedge \neg R(n) \leftrightarrow \{e\}^J(n)\downarrow 0$$

for some index e, are precisely the hyperarithmetic ones.

Recall that a set $X\subseteq\omega$ (${}^\omega\omega$) is said to be in $\exists\Gamma$ for some (adequate) pointclass Γ on the integers (Baire space), if there is a set $Y\subseteq\omega\times{}^\omega\omega$ (${}^\omega\omega\times{}^\omega\omega$) so that $X=\{x|$ *Player I has a winning strategy in* $G(Y_x,\ {}^{<\omega}\omega)\}$ where $Y_x=\{y|\ \langle x,y\rangle\in Y\}$. Roughly speaking, if one has a recursive listing of the Γ sets of reals, (say from some universal Γ set): $A_0, A_1, \ldots, A_n, \ldots$, then a *complete* $\exists\Gamma$ set of integers, gives those n for which *I* has a winning strategy in $G(A_n;{}^{<\omega}\omega)$.

We have the following theorem connecting this with determinacy of open games:

Theorem 1.1. *(Moschovakis* [14]*, Svenonius* [17]*) The complete* $\exists\Sigma^0_1$ *set of integers is a complete* Π^1_1 *set of integers.*

Hence by Kleene's results just alluded to:

Corollary 1.2. *The complete* $\exists\Sigma^0_1$ *set of integers is recursively isomorphic to* H^J*, a complete J-semi-decidable set of integers.*

Moreover:

Theorem 1.3. *(Blass* [2]*) Any* Σ^0_1*-game for which the open player, that is I, has a winning strategy, has a hyperarithmetic winning strategy.*

Corollary 1.4. *Any* Σ^0_1*-game for which player I has a winning strategy, has a J-recursive strategy.*

We seek to raise these ideas to the level of Σ^0_3. Kleene also gave an equivalent account of recursion in objects of finite type using as an alternative the Turing model enhanced with oracle calls to a higher type functional, see [10],[11]; the account here is motivated in spirit by that approach. Instead of using an equational calculus we shall couch this in terms of *infinite time Turing machines* -(ittm's) computations recursive in a certain operator eJ in place of J. Indeed there is already a version of this kind of computation in the literature. In [12] Lubarsky defines the notion of a *'feedback'-ittm machine*, where a Hamkins-Kidder ittm may call upon a sub-computation handled by another such machine, and pass an index and an element of Cantor space to it as a parameter. The information passed back is as to whether the computation with the given index acting on the given parameter *halts* or not (which it may do after a transfinite number of steps, in contradistinction to the standard Turing machine). This is thus in the spirit of the jump J defined above. convergent *feedback-ittm computation* can then be conceived as a wellfounded tree of halting sub-computations. *divergent computation* ("freezing" in Lubarsky's terminology) is one which descends down an ill-founded path.

Rather than define recursions involving what would be the generalization of J above to *halting* ittm-computations, we use an *eventual jump* operator eJ. The ittm's have an arguably more fundamental behaviour than 'halting' or 'non-halting': they may eventually have some settled output on their output tape without formally entering a halting state (the Read/Write head may be meandering up and down the tape, perhaps fiddling with the Scratch or Input tape, but leaving the output alone, in some fixed loop without formally halting). This 'eventual' or 'settled' behaviour fits in with the Σ_2 definable liminf rules of its operation. We thus define:

$$\begin{aligned}\mathrm{eJ}(e,\vec{m},\vec{x}) &= 1 \text{ if } \{e\}(\vec{m},\vec{x})| \text{ (denoting } \textit{converges to a settled ouput}\text{)}\\ &= 0 \text{ otherwise}\end{aligned}$$

Here $\{e\}$ is now an index of a standard ittm-computable function, say given by some usual finite programme $P_e(\vec{m},\vec{x})$. We then consider ittm-computations recursive in eJ, for which we would now use the notation $\{e\}^{\mathrm{eJ}}$ to denote the e'th such function recursive in eJ. Here a *query* instruction or state is included as part of the machine's language. For this notion we find a level of the L hierarchy L_{α_0} to provide an analogy with the above.

Theorem 1.5. *The complete $\Im\Sigma^0_3$ set of integers is recursively isomorphic to H^{eJ}, the complete* eJ*-semi-decidable set of integers.*

Thus to paraphrase, the listing of those games that are won by I, essentially has the same information as the 'halting problem' for this notion of recursion. We feel this is interesting as it demonstrates that two, *prima facie* very different, notions are in fact intimately connected. Define τ_0 as the supremum of the convergence times of eJ-recursive computations.

Corresponding to the result on Π^1_1 we have:

Theorem 1.6. *The complete $\Im\Sigma^0_3$ set of integers is a complete $\Sigma_1^{L_{\tau_0}}$ truth set.*

(Recall that the complete Π^1_1 set is also the $\Sigma_1^{L_{\omega_1^{\mathrm{ck}}}}$ truth set.) Moreover

Theorem 1.7. *Any Σ^0_3-game for which the player I has a winning strategy, has an* eJ*-recursive winning strategy.*

Corresponding to the result on hyperarithmetic strategies we have:

Corollary 1.8. *Any Σ^0_3-game for which player I has a winning strategy, has a winning strategy in L_{τ_0}.*

We assume the reader has familiarity both with the constructible hierarchy of G del - for which see Devlin [4]. For the basic notions of descriptive set theory including the elementary theory of Gale-Stewart games, see Moschovakis [15]. Our notation is standard. Some of the results here relate to sub-systems of second order number, or analysis, and the basic theory of this is exposited in Simpson's monograph [16]. For models of admissible set theory, also called "Kripke-Platek set theory" or "KP" see Barwise [1]. By "KPI" we mean the theory KP augmented by the axiom that every set is an element of some admissible set.

In the language of generalized recursion theory, the pointclass $\partial\Sigma^0_3$ of sets of integers cannot be the 1-envelope of a normal type-2 function, by results of Harrington, Kechris, and Simpson (see [7]). ("1-envelope" is the set of relations on ω recursive in the type-2 functional.) What we are showing here is that the complete set of integers in $\partial\Sigma^0_3$ is however (recursively isomorphic to) the complete set which is ittm-semi-recursive in eJ - the eventual jump type-2 functional. It is the "ittm-1-envelope" of eJ. Section 3 contains some facts related to ittm-computations, and an exposition, and sets the scene with some basic results of our ittm-recursions-in-eJ.

We answer a further question of Lubarsky concerning Freezing-ITTM's at Corollary 4.9.

cknowledgements: We should like to warmly thank Bob Lubarsky for illuminating explanations of his paper [12], discussions on the conjecture mentioned in the second paragraph, and comments on an earlier draft of this paper.

2

We first repeat the extraction from our earlier paper [21] of a criterion for the constructible rank of Σ^0_3 games' strategies. (Note that we take our games as defined in L and using constructible, indeed an initial recursive, game trees; the existence of a winning strategy for a particular Σ^0_3 (indeed arithmetic or Borel) game is a Σ^1_2 assertion about the countable tree T and the payoff set. s $T \in L$ the truth of such an assertion has the same truth value in the universe of sets or in L. We thus expect to find such strategies in L (since Davis in [3] proved such strategies exist in the universe V of sets). But where are they?

Definition 2.1. *A pair of ordinals (μ, ν) is a* Σ_2-extendible pair*, if $L_\mu \prec_{\Sigma_2} L_\nu$ and moreover ν is the least such with this property. We say μ is* Σ_2-extendible *if there exists ν with (μ, ν) a* Σ_2-extendible pair*. By relativisation, a pair of ordinals (μ, ν) is an x-*Σ_2-extendible pair*, and μ is x-*Σ_2-extendible*, if $L_\mu[x] \prec_{\Sigma_2} L_\nu[x]$.*

Indeed all the above ideas relativise normally to real parameters $x \in 2^{\mathbb{N}}$, and we thus have $\lambda(x)$, $\zeta(x), \Sigma(x)$ *etc.*, with the latter two forming the least x-Σ_2-extendible pair.

Definition 2.2. *Let an m-*depth Σ_2-nesting *of an ordinal α be a sequence $(\zeta_n, \sigma_n)_{n<m}$ with (i) For $0 \leq n < m$: $\zeta_{n-1} \leq \zeta_n < \alpha < \sigma_n < \sigma_{n-1}$; (ii) $L_{\zeta_n} \prec_{\Sigma_2} L_{\sigma_n}$. We write $d(\alpha) \geq m$. If α is not nested we set $d(\alpha) = 0$.*

We shall want to consider non-standard admissible models (M, E) of KP together with some other properties. We let $\mathrm{WFP}(M)$ be the wellfounded part of the model. By the so-called 'Truncation Lemma' it is well known (*v.* [1]) that this well founded part must also be an admissible set. Usually for us the model will also be a countable one of "$V = L$". Let M be such and let $\alpha = \mathrm{On} \cap \mathrm{WFP}(M)$. By the above α is thus an 'admissible ordinal', *i.e.* L_α will also be a KP model.
n 'ω-depth' nesting cannot exist by the wellfoundedness of the ordinals. However an ill founded model M when viewed from the outside may have infinite descending chains of 'M-ordinals' in its ill founded part. These considerations motivate the following definition.

Definition 2.3. *An* infinite depth Σ_2-nesting of α based on M *is a sequence $(\zeta_n, s_n)_{n<\omega}$ with :*
(i) $\zeta_{n-1} \leq \zeta_n < \alpha \subset s_n \subset s_{n-1}$; (ii) $s_n \in \mathrm{On}^M$; (iii) $(L_{\zeta_n} \prec_{\Sigma_2} L_{s_n})^M$.

Thus the s_n form an infinite descending E-chain through the illfounded part of the model M. In [20] we devised a game whereby one player produced an ω-model of a theory and the other player tried to find such infinite descending chains through M's ordinals. In this paper we shall switch the roles of the players, and have Player *II* produce the model and Player I attempt to find the chain. (This is just to orientate the game as then Σ^0_3.)

In order for there to exist a non-standard model with an infinite depth nesting (of the ordinal of its wellfounded part) then the wellfounded part will already be a relatively long countable initial segment of L (it is easy to see that if $\zeta = \sup_n \zeta_n$ then already $L_\zeta \models \Sigma_1$-Separation).

Example 2.4. (i) Let δ be least so that $L_\delta \models \Sigma_2$-Separation, and let (M, E) be an admissible non-wellfounded end extension of L_δ with L_δ as its wellfounded part. Then there is an infinite depth nesting of δ based on M.

(ii) By refining considerations of the last example, let γ_0 be least such that there is $\gamma_1 > \gamma_0$ with $L_{\gamma_0} \prec_{\Sigma_2} L_{\gamma_1} \models \mathrm{KP}$. Then again there is an infinite depth nesting of γ_1 based on some illfounded end extension M of L_{γ_1}.

Both of the above can be established by standard Barwise Compactness arguments. However both these δ and γ_0 we shall see are greater than the ordinal β_0 defined from this notion of nesting as follows.

Definition 2.5. *Let β_0 be the least ordinal β so that L_β has an admissible end-extension (M, E) based on which there exists an infinite depth Σ_2-nesting of β.*

Definition 2.6. *Let γ_0 be the least ordinal so that for any game $G(A, T)$ with $A \in \Sigma^0_3$, $T \in L_{\gamma_0}$ a game tree, then there is a winning strategy for a player definable over L_{γ_0}.*

The following then pins down the location of winning strategies for games at this level played in, *e.g.* recursive trees.

Theorem 2.7. *$\gamma_0 = \beta_0$. Moreover, any Σ^0_3-game for a tree T, with a strategy for Player I, has such a strategy an element of L_{β_0}. Any Π^0_3-game for such a tree has a strategy which is definable over L_{β_0}.*

Definition 2.8. *Let η_0 be the closure ordinal of monotone $\circlearrowright\Sigma^0_3$-operators.*

This ordinal will be less than β_0.

Theorem 2.9. *(a) The following sets are recursively isomorphic.*

*(i) The complete ittm-semi-recursive-in-*eJ *set, H^{eJ};*

(ii) the Σ_1-theory of $(L_{\eta_0}, \in)$;

(iii) the complete $\circlearrowright\Sigma^0_3$ set of integers.

*(b) The ittm-recursive-in-*eJ *sets of integers are precisely those of L_{η_0}.*

Definition 2.10. *Let τ_0 be the supremum of convergence ordinals of well-founded computations, arising from infinite time Turing machine computations on integers which are ittm-recursive (in a generalized sense of Kleene* et al.*) in the Type-2 eventual jump functional* eJ.

Theorem 2.11. $\eta_0 = \tau_0$.

Remark: (i) The proof reveals more about the L-least strategies for Σ^0_3-games: those for player I, in fact can be found within a strictly bounded initial segment of β_0: they will occur in L_{η_0}.

(ii) The existence of all the above ordinals, and β-models of the above theories can be proven in the subsystem of analysis Π^1_3-CA_0, but not in Δ^1_3-CA_0 (or even some strengthenings of the latter) See [20].

2.1 The location of strategies for Σ^0_3-games

Proof: of Theorem 2.7 We look at the construction of the proof of Theorem 5 of [20] in particular that of Lemma 3. There we used an assumption that there is a triple of ordinals $\gamma_0 < \gamma_1 < \gamma_2$ with (a) $L_{\gamma_0} \prec_{\Sigma_2} L_{\gamma_1}$ and (b) $L_{\gamma_0} \prec_{\Sigma_1} L_{\gamma_2}$ and (c) γ_2 was the second admissible ordinal beyond γ_1. One assumed that I did not have a winning strategy in $G(A;T)$. The Lemma 3 there ran as follows:

Lemma 2.12. *Let $B \subseteq A \subseteq \lceil T \rceil$ with $B \in \Pi^0_2$. If $(G(A;T)$ is not a win for $I)_{L_{\gamma_0}}$, then there is a quasi-strategy $T^* \in L_{\gamma_0}$ for II with the following properties:*

(i) $\lceil T^* \rceil \cap B = \varnothing$

(ii) $(G(A;T^*)$ is not a win for $I)_{L_{\gamma_0}}$.

The format of the lemma's proof involved showing that the $\Sigma_2^{L_{\gamma_0}}$ notion of 'goodness' embodied in (i) and (ii) held for $\varnothing$. To do this involved defining goodness in general. We first define T' as *II*'s *non-losing quasi-strategy* for $G(A;T)$ (the set of positions $p \in T$ so that *I* does not have a winning strategy in $G(A;T_p)$); this is Π_1 definable over L_{γ_0} as the latter is a model KPI; in particular if we use the notation

Definition 2.13. $S^1_\gamma =_{\mathrm{df}} \{\delta < \gamma |\ L_\delta \prec_{\Sigma_1} L_\gamma\}$.

Definition 2.14. *For $n \leq \omega$, let T^n_δ denote the Σ_n-theory of L_δ.*

then " $p \in T'$ " is $\Pi_1^{L_{\zeta_0}}$, where $\zeta_0 =_{\mathrm{df}} \min S^1_{\gamma_0} \backslash \rho_L(T)$. More generally we define:

$p \in T'$ is *good* if there is a quasi-strategy T^* for *II* in T'_p so that the following hold:
(i) $\lceil T^* \rceil \cap B = \varnothing$;
(ii) $G(A;T^*)$ is not a win for I.

Here T'_p is the subtree of T' below the node p. The point of requiring that the pair (γ_0, γ_1) have the Σ_2-reflecting property of (a) above, is that the class H of good p's of L_{γ_1} is the same as that of L_{γ_0} and so is a set in L_{γ_1} as it is thus definable over L_{γ_0} by a $\Sigma_2(\{T'\})$ definition. The overall argument is a proof by contradiction, where we assume that $\varnothing$ is in fact not good, and proceeds to construct a strategy σ for Player I in the game $G(A;\ T')$, which is definable over L_{γ_1}, and is apparently winning in L_{γ_2}. (The requirement (c) that γ_2 be a couple of admissibles beyond γ_1 was only to allow for the strategy σ to be seen to be truly winning by going to the next admissible set, and verifying that there are no winning runs of play for *II*.) The contradiction arises since T' - which was defined as the subtree of T of *II*'s non-losing positions - is concluded still to be the same subtree of non-losing positions in L_{γ_2}. Being a non-losing position, p say, for *II* is a Π_1 property of p. This carries up from L_{γ_0} to L_{γ_2} as $L_{\gamma_0} \prec_{\Sigma_1} L_{\gamma_2}$, and this is the reason for the requirement (b): we want T' to survive beyond L_{γ_1} for our argument to work. (This idea is important for the arguments in Section 4, so let us refer to it as '*the survival argument*'.) There is then no winning strategy for I in $G(A;T')$ definable over L_{γ_1}, contradicting the reasoning that σ is such.

This proves the Lemma: L_{γ_1} sees there is T^* a subtree of T' witnessing that $\varnothing$ is good. The existence of such a subtree is a $\Sigma_2(\{T'\})$-sentence, and then again this reflects down to L_{γ_0}. We thus have such a T^* in L_{γ_0}.

The Theorem is proven by repeated applications of the Lemma, by using the argument for each Π^0_2 set B_n in turn where $A = \bigcup_n B_n$ and refining the trees using this procession from a tree to a subtree T^*. We thus repeat the argument with T^* replacing T. Because $T^* \in L_{\gamma_0}$ we have the same constellation of this triple of ordinals γ_i above the constructible rank of T^*, and can do this.

However we can get away with less. The definition of the subtree of non-losing positions of *II* now this time in the new T^* can be considered as taking place Π_1 over L_{δ_0} where δ_0 is the least element of $S^1_{\gamma_0}$ with $T^* \in L_{\delta_0}$. To get our contradiction we actually use that $L_{\delta_0} \prec_{\Sigma_1} L_{\gamma_2}$; we do not need that $L_{\gamma_0} \prec_{\Sigma_1} L_{\gamma_2}$. Notice that our argument that T^* exists is non-constructive: we simply say that the Σ_2-sentence of its existence reflects to L_{γ_0}: we do not have any control over its constructible rank below γ_0. Moreover any sufficiently large γ' greater than γ_1 would do for the upper ordinal, as long as it is a couple of admissibles larger than γ_1. Thus we could apply the Lemma repeatedly for different B_n if we have a guarantee that whenever a T^*_n-like subtree is defined there exists a $\zeta_n \in S^1_{\gamma_0}$ and a suitable upper ordinal $\gamma_n > \gamma_1$ with $T^*_n \in L_{\zeta_n} \prec_{\Sigma_1} L_{\gamma_n}$. Of course if there are arbitrarily large ζ_n below γ_0 with this extendability property, then this is tantamount to $L_{\gamma_0} \prec_{\Sigma_1} L_{\gamma'}$ for some suitable γ', and this shows why our original constellation of γ_i provides a sufficient condition.

ctually as the final paragraph of the Theorem 5 there shows, we are doing slightly more than this: we are, each time, applying the Lemma infinitely often to each possible subtree of T^* below some node p_2 of it which is of length 2, to define our strategy τ applied to moves of length 3. We then move on to the next Π^0_2 set. lthough we are applying the Lemma infinitely many times for each such p_2, and thus infinitely many new Σ_2-sentences, or trees, have to be instantiated, we had that L_{γ_0} is a Σ_2-admissible set, and as the class of such p_2 is just a set of L_{γ_0}, Σ_2-admissibility works for us to find a bound for the ranks of the witnessing trees, as some $\delta < \gamma_0$. We thus can claim that our final τ is an element of L_{γ_0} even after ω-many iterations of this process.

$(\beta_0 \geq \gamma_0)$ We argue for this. Let (M, E) be a non-standard model of KP with an infinite nesting (ζ_n, s_n) about β_0 as described. Note that $S^1_{\beta_0}$ must be unbounded in β_0 (so that $L_{\beta_0} \models \Sigma_1$-Separation), and each ζ_n is a limit point of $S^1_{\beta_0}$. We do not assume that β_0 is Σ_2-admissible (which in fact it is not as the proof shows). Let $T \in L_{\beta_0}$ be a game tree. By omitting finitely much of the outer nesting we assume $T \in L_{\zeta_0}$. We assume that Player I has no winning strategy for $G(A;T)$ in L_{β_0} (for otherwise we are done). Note that in M we have that L_{s_0} also has no winning strategy for this game (otherwise the existence of such would reflect into L_{β_0}. We show that II has a winning strategy definable over L_{β_0}. Let $A = \bigcup B_n$ with each $B_n \in \Pi^0_2$. For $n = 0$ we apply the argument of the Lemma using the pair (ζ_1, s_1) in the role of (γ_0, γ_1) from before, with (ζ_0, s_0) in the role of (δ_0, γ_2) described above, *i.e.* we use only that $T \in L_{\zeta_0}$ and that $L_{\zeta_0} \prec_{\Sigma_1} L_{s_0}$.

The Lemma then asserts the existence of a quasi-strategy for *II* definable using the pair (ζ_1, s_1): $T^*(\varnothing)$. By Σ_2-reflection the L-least such lies in L_{ζ_1}, and we shall assume that $T^*(\varnothing)$ refers to it.

Claim: For any pair (ζ_n, s_n) *for* $n \geq 1$ *the same tree* $T^*(\varnothing)$ *would have resulted using this pair.*

Proof: Note that we can define such a tree like $T^*(\varnothing)$ using such pairs, since for all of them we have that $(\zeta_0, s_0) \supset (\zeta_1, s_1) \supset (\zeta_m, s_m)$ for $m > 1$. s $T^*(\varnothing) \in L_{\zeta_1}$ and satisfies a Σ_2 defining condition there, and since we also have $\zeta_1 \in S^1_{\zeta_m}$, it thus satisfies the same Σ_2 condition in L_{ζ_m}.
Q.E.D. *Claim*

For any position $p_1 \in T$ with $\mathrm{lh}(p_1) = 1$, let $\tau(p_1)$ be some arbitrary but fixed move in $T'(\varnothing)$, this now *II*'s non-losing quasi-strategy for the game $G(A, T^*(\varnothing))$ as defined in L_{ζ_2}. The relation "$p \in T'(\varnothing)$" is $\Pi_1^{L_{\zeta_2}}(\{T^*(\varnothing)\})$ or equivalently $\Pi_1^{L_{\zeta_1}}(\{T^*(\varnothing)\})$, or indeed $\Pi_1^{L_\delta}(\{T^*(\varnothing)\})$ where δ is least in $S^1_{\zeta_1}$ above $\rho_L(T^*(\varnothing))$. Hence "$y = T'(\varnothing)$" $\in \Delta_2^{L_\delta}(\{T^*(\varnothing)\})$ and thus $T'(\varnothing)$ also lies in L_{ζ_1}. For definiteness we let $\tau(p_1)$ be the numerically least move.

For any play, p_2 say, of length 2 consistent with the above definition of τ so far, we apply the lemma again with $B = A_1$ replacing $B = A_0$ and with $(T^*(\varnothing))_{p_2}$ replacing T. We use the nested pair (ζ_2, s_2) to define quasi-strategies for *II,* call them $T^*(p_2)$, one for each of the countably many p_2. These are each definable in a Σ_2 way over L_{ζ_2}, in the parameter $(T^*(\varnothing))_{p_2}$. This argument uses that $(T^*(\varnothing))_{p_2} \in L_{\zeta_1} \prec_{\Sigma_1} L_{s_1}$. Let $T'(p_2) \in L_{\zeta_2}$ be *II*'s non-losing quasi-strategy for $G(A, T^*(p_2))$, this time with "$y = T'(p_2)$"$\in \Delta_2^{L_{\zeta_2}}(\{T^*(p_2)\})$. (gain these will satisfy the same definitions as over L_{ζ_m} for any $m \geq 2$.) Note that we may assume that the countably many trees $T'(p_2)$ appear boundedly below ζ_2 (using the Σ_2-admissibility of ζ_2). gain for $p_3 \in T^*(p_2)$ any position of length 3, let $\tau(p_3)$ be some arbitrary but fixed move in $T'(p_2)$. Now we consider appropriate moves p_4 of length 4, and reapply the lemma with $B = A_2$ and $(T^*(p_2))_{p_4}$. Continuing in this way we obtain a strategy τ for *II*, so that $\tau \restriction {}^{[1,2k+2)}\omega$, for $k < \omega$, is defined by a length k-recursion that is $\Sigma_2^{L_{\zeta_k}}(\{T\})$.

s the argument continues more and more of the strategy τ is defined using successive (ζ_m, s_m) to justify the existence of the relevant trees in L_{ζ_m}. *Knowing* that the trees are there for the asking, we see that τ can actually be defined by a Σ_2-recursion over L_{β_0} in the parameter T in precisely the manner given above (the Σ_2-inadmissibility of β_0 notwithstanding).

If x is any play consistent with τ, then for every n, by the defining properties of $T^*(p_{2n})$ given by the relevant application of the lemma, $x \in \lceil T^*(x \restriction 2n) \rceil \subseteq \neg A_n$. Hence $x \notin A$, and τ is a winning strategy for *II* as required. Thus $\beta_0 \geq \gamma_0$ is demonstrated.

$(\beta_0 \leq \gamma_0)$: suppose $\beta_0 > \gamma_0$. Then, since the existence of a winning strategy for a player in any particular $\Im\Sigma_3^0$ game would be part of the theory $T^1_{\beta_0} = T^1_{\alpha_0}$ where α_0 is least with $L_{\alpha_0} \prec_{\Sigma_1} L_{\beta_0}$, and since moreover that the existence of a stage γ_0 over which *all* such games have strategies, amounts also to an existential statement, we have that $\gamma_0 < \alpha_0$. But this is an immediate contradiction: find a $\psi \in T^1_{\alpha_0}$ with $\gamma_0 < \alpha_\psi < \alpha_0$. But as before *II* has as winning strategy σ to play a code for L_{α_ψ}. Hence as $\gamma_0 < \alpha_\psi$ such a strategy and so such a code can be found in L_{α_ψ}; but again as before, this contradicts Tarski. Contradiction. Hence $\beta_0 \leq \gamma_0$. Q.E.D. Theorem 2.7

Remark 2.15. We make some definitions from the $(\beta_0 \geq \gamma_0)$ part of the last proof for later use. We have our starting tree T, and the tree of non-losing positions for *II*, T'. We shall call these the trees of *depth 0*. Then for any $p \in T'$ we argued that p was good, and, since $\varnothing$ was good, we could define the tree $T^*(\varnothing)$ - the L-least tree witnessing this fact, and thence we had $T'(\varnothing)$ the tree of non-losing positions for *II* in $G(A, T^*(\varnothing))$. We give the trees $T^*(\varnothing), T'(\varnothing)$ *depth 1*. Then for any position $p_1 \in T$ with $\mathrm{lh}(p_1) = 1$, we let $\tau(p_1)$ be the numerically least move in $T'(\varnothing)$. We call any play, p_2 say, of length 2 consistent with this definition of τ so far, *relevant (of length 2).* We wished to apply the lemma again with $B = A_1$ replacing $B = A_0$ and with $(T^*(\varnothing))_{p_2}$ replacing T. We shall call a tree of the form $(T^*(\varnothing))_{p_2}$ or $((T^*(\varnothing))_{p_2})'$ (the latter the tree of non-losing moves for *II* in $G(A; (T^*(\varnothing))_{p_2})$) *relevant trees of depth 1*. We then used (ζ_2, s_2) to define the $T^*(p_2)$ (one tree for each relevant p_2) and thence the trees $T'(p_2)$ to be *II*'s non-losing quasi-strategy for $G(A, T^*(p_2))$. We give trees of the form $T^*(p_2), T'(p_2)$ *depth 2*. For $p_3 \in T^*(p_2)$ any position of length 3, $\tau(p_3)$ was the numerically least move in $T'(p_2)$. gain we call such $p_4 = p_3{}^\frown\tau(p_3)$ *relevant*, and the corresponding trees $(T^*(p_2))_{p_4}$ and $(T^*(p_2))_{p_4})'$ *relevant trees of depth 2*. $T^*(p_4), T'(p_4)$ will be of *depth 3*. nd so forth.

Definition 2.16. *Let $\mathbb{T}^k$ denote the set of trees, and relevant trees, of depth k, as just defined for $k < \omega$.*

We return now to considering the complexity of $\Im\Sigma_3^0$.

Theorem 2.17. *Let α_0 be least with $T^1_{\alpha_0} = T^1_{\beta_0}$ (thus $\alpha_0 = \min S^1_{\beta_0}$).*

(i) $T^1_{\alpha_0}$ is a complete $\Im\Sigma_3^0$ set of integers.

(ii) Hence the reals of L_{α_0} are all $\Im\Sigma_3^0$ sets of integers.

Proof: The argument is really close to that of the Corollary 2 of [20]. Indeed there we showed that the $T^1_{\alpha_\psi}$ (which occurred cofinally in L_{α_0}) were $\Im\Sigma^0_3$ sets. Some details of this are repeated. First remark that (ii) is immediate given (i) since all the other reals in L_{α_0} are all recursive in $T^1_{\alpha_0}$ and $\Im\Sigma^0_3$, being a Spector class (*v.* [15]), is closed under recursive substitution. We define a game G^*_φ for Σ_1-sentences φ.

Rules for II.
In this game *II*'s moves in x must be a set of G del numbers for the complete Σ_1-theory of an ω-model of $\mathrm{KP}+V=L+(\neg\varphi\wedge\mathrm{Det}(\Sigma^0_3))$.

Everything else remains the same *mutatis mutandis*: *I*'s Rules remain the same and his task is to find an infinite descending chain through the ordinals of *II*'s model. Note that if $\varphi\in T^1_{\alpha_0}$, *I* now has a winning strategy: for if *II* obeys her rules, and lists an x which codes an ω-model M of this theory, then M is not wellfounded, and has $\mathrm{WFP}(M)\cap\mathrm{On}<\rho(\varphi)$ where $\rho(\varphi)$ is defined as the least ρ such that $\varphi\in T^1_{\rho+1}$. However *I* playing (just as *II* did in the main Theorem 4) can find a descending chain and win. For we have $\mathrm{WFP}(M)\cap\mathrm{On}<\beta_0$ and so the argument goes through, as there are no infinite depth nestings there. On the other hand if $\varphi\notin T^1_{\alpha_0}$, *II* may just play a code for the true wellfounded $L_{\beta_0^+}$ with β_0^+ the least admissible above β_0+1, and so win. This shows that $T^1_{\alpha_0}$ is a $\Im\Sigma^0_3$ set of integers.

Now suppose $a\in\Im\Sigma^0_3$. Then we have some Σ^0_3 set $A\subseteq\omega\times{}^\omega\omega$ with $n\in a\longleftrightarrow I$ has a winning strategy to play into $A_a=\{y\in{}^\omega\omega\,|\,(a,y)\in A\}$. Then a is $\Sigma^{L_{\alpha_0}}_1$ (since all Σ^0_3-games that are a win for *I*, have a winning strategy an element of L_{β_0}, and thence by Σ_1-elementarity, the L-least such is actually an element of L_{α_0} - and we merely have to search through L_{α_0} for it) and thus is recursive in $T^1_{\alpha_0}$. Hence $T^1_{\alpha_0}$ is a complete $\Im\Sigma^0_3$ set of integers. Q.E.D. Theorem 2.17 and 2.9(a) (ii)$\leftrightarrow$(iii).

In conclusion: we saw above that α_0 was the least α with $T^1_\alpha=T^1_{\beta_0}$. Phrased in other terms, by elementary constructible hierarchy considerations, this is saying that α_0 is the minimum of $S^1_{\beta_0}$. Hence $L_{\alpha_0}\prec_{\Sigma_1}L_{\beta_0}$ but for no smaller δ is $L_\delta\prec_{\Sigma_1}L_{\beta_0}$. Since the statement "There is a winning strategy for Player I in $G(A,T)$" is equivalent in KPI to a Σ_1-assertion, if true in L_{β_0} it is true in L_{α_0}. In short for those Σ^0_3-games that are wins for I on trees $T\in L_{\alpha_0}$, there are strategies for such also within L_{α_0} itself. For those that are wins for Player *II*, when not found in L_{α_0}, these may be defined over L_{β_0}. This somewhat asymmetrical picture reflects the earlier theorems cited above. The theorems of the next section harmonise perfectly with this.

Remark: (i) Since $\Im\Sigma^0_3$ is a Spector class, one will have a $\Im\Sigma^0_3$-prewellordering of $T^1_{\alpha_0}$ as a $\Im\Sigma^0_3$ set of integers, of maximal length, here α_0.

We write down one on $T=T^1_{\alpha_0}$. bbreviate $\Gamma=\Im\Sigma^0_3$ and $\check\Gamma=\Im\Pi^0_3$. We need to provide relations $\leq_\Gamma$ and $\leq_{\check\Gamma}$ in Γ and $\check\Gamma$ respectively, so that the following hold:

$$T(y)\Longrightarrow\forall x\{[T(x)\wedge\rho(x)\leq\rho(y)]\Longleftrightarrow x\leq_\Gamma y\Longleftrightarrow x\leq_{\check\Gamma}y\}.$$

For the relation $x\leq_\Gamma y$, we define the game where *II* produces a model M^{II} of $T(y)\wedge(\neg T(x)\vee\rho(x)\not\leq\rho(y))$ and *I* tries to demonstrate that it is illfounded. ssume then $T(y)$. If $T(x)\wedge\rho(x)\leq\rho(y)$ then *either* $(\neg T(x))^{M^{II}}$ and thus M^{II} is illfounded with $\mathrm{WFP}(M^{II})\cap\mathrm{On}<\rho(x)$ and hence *I* can win as in this region there are no ω-nested sequences. *Or*: $(\rho(x)\not\leq\rho(y))^{M^{II}}$. Thus $(\rho(x)>\rho(y))^{M^{II}}$ and again this implies $\mathrm{WFP}(M^{II})\cap\mathrm{On}<\rho(x)$ with *I* winning.

Conversely suppose $x\leq_\Gamma y$. Since $T(y)$ is assumed, if $\neg T(x)$, then *II* can play a wellfounded model with $(y\wedge\neg x)^{M^{II}}$ and win. If $\rho(x)>\rho(y)$ then again the same can be done. This proves the first equivalence above. The second is similar, with now *I* producing a model M^I of $T(x)\wedge\rho(x)\leq\rho(y)$ and *II* finding descending chains. We leave the details to the reader.

(ii) One may also write out directly the theories T^1_α for $\alpha < \alpha_0$ in a $\partial\Pi^0_3$ form. This should not be surprising: a $\partial\Sigma^0_3$ norm as above should have 'good' $\Delta(\partial\Sigma^0_3)$ initial segments.

(iii) For any set $A \in \partial\Pi^0_3 \backslash \partial\Sigma_3$ there will be $n \in A$ so that the winning strategy witnessing this is definable over L_{β_0} but not an element thereof. (Otherwise an admissibility and Σ_1-reflection argument shows that there is a level L_δ with $\delta < \alpha_0$ containing strategies for both A and its complement. But that would make $A \in \Delta(\partial\Sigma^0_3)$ - a contradiction.)

Corollary 2.18. $\eta_0 = \alpha_0$.

Proof: Since $\partial\Sigma^0_3$ is a Spector class, and we see that a complete $\partial\Sigma^0_3$ set has a $\partial\Sigma^0_3$ -norm of length α_0, standard reasoning shows that there is a $\partial\Sigma^0_3$-monotone operator whose closure ordinal is α_0. Hence $\eta_0 = \alpha_0$. Q.E.D.

Results of Martin in [13] show that for a co-Spector class, $\check{\Gamma}$ say, the closure ordinal of monotone $\check{\Gamma}$-operators, $o(\check{\Gamma}\text{-mon}) =_{\rm df} \sup\{o(\Phi) \mid \Phi \in \check{\Gamma}, \Phi \text{ monotone}\}$, is *non-projectible*, that is $L_{o(\check{\Gamma}\text{-mon})} \models \Sigma_1$-Sep. Moreover $o(\Gamma) < o(\check{\Gamma}\text{-mon})$.

He shows:

Theorem 2.19. *(Theorem D [13]) Let Γ be a Spector pointclass. (i) Suppose that for every $X \subseteq \omega$, and every $\check{\Gamma}(X)$ monotone Φ, that $\Phi^\infty \in \check{\Gamma}(X)$, then $o(\check{\Gamma}\text{-mon})$ is non-projectible, that is $S^1_{o(\check{\Gamma}\text{-mon})}$ is unbounded in $o(\check{\Gamma}\text{-mon})$.*

(ii) (from the proof of his Lemma D.1) $o(\Gamma\text{-mon}) \in S^1_{o(\check{\Gamma}\text{-mon})}$.

(He shows too that for Spector classes such as $\partial\Sigma^0_3$, the supposition of (i) is fulfilled.) If we set π_0 to be the closure ordinal of $\partial\Pi^0_3$-mon. operators, then in this context we have an upper bound for π_0:

Lemma 2.20. $\alpha_0 < \pi_0 \leq \beta_0$.

Proof: By (ii) of the last theorem, $\alpha_0 \in S^1_{\pi_0}$. But for no $\beta' > \beta_0$ do we have $L_{\alpha_0} \prec_{\Sigma_1} L_{\beta'}$ (as there are games with winning strategies (for *II*) in L_{β_0+1} for which there are none in L_{β_0}).

Q.E.D.

Question: Is $\pi_0 = \beta_0$?

3 Recursion in eJ

3.1 Kleene Recursion in higher types

We take some notation and discussion from Hinman [8]. There was developed the basic theory of higher type recursion based on an equational calculus defined by Kleene and refined by him and Gandy in the 1960's. The basic intuition was to define recursions using not just recursive functions on integers but also allowing recursive schemes using 'computable' functions $f : \omega \times {}^\omega\omega \longrightarrow \omega$ (and similarly for domains which are product spaces of this type). basic result in this area is that the functions recursive in E (defined below) are precisely those recursive in J, the 'ordinary Turing jump', where we set

$$J(e,\boldsymbol{m},\boldsymbol{x}) = 0 \text{ if } \{e\}(\boldsymbol{m},\boldsymbol{x})\downarrow$$
$$= 1 \quad \text{otherwise.}$$

(We shall follow mostly Hinman in using boldface notation, early or mid-alphabet roman for integers, but end alphabet roman for elements of ${}^{\omega}\omega$, to indicate an (unspecified) number of variables of the given type in an appropriate product space ${}^{k}\omega \times {}^{l}({}^{\omega}\omega)$ - which he abbreviates as ${}^{k,l}\omega$.) Then E (often written ${}^{2}E$) is the functional:

$$E(x) \quad = 0 \quad \text{if } \exists n(x(n) = 0)$$
$$= 1 \quad \text{otherwise.}$$

For a fixed type-2 functional I of the kind above - thus a function $I\colon {}^{k}\omega \times {}^{l}({}^{\omega}\omega) \longrightarrow \omega$ such as E or J just defined, an inductive definition of a set, $\Omega(I)$, consisting of equational clauses can be built up in ω_1-steps. This defines the class of those functions $\{e\}^I$ that are recursive in I. Of course such include partial functions, as a descending chain of subcomputation calls in the tree of computations represents divergence. Just as the clauses of the induction and the set $\Omega(I)$ is an expansion of those clauses and functions of type-1 recursion, also due to Kleene and yielding an inductive set Ω, we shall wish to expand the notion of 'computation' further along another axis.

Our notation for computation will be modelled on that of the transfinite machine model, the 'infinite time Turing machine' introduced by Hamkins and Kidder [5]. The signifying feature of such ITTM's is the transfinite number of stages that they are allowed to run their standard finite Turing program, on their one-way infinite tape. The behaviour at limit stages is defined by a 'liminf' rule for the cell values of 0 or 1, and a replacing of the read/write head back at the start of the tape, and finally a special 'limit state' q_L is entered into.

ctually the formalism is quite robust: one may change details of these arrangements without altering the computational power. In [5] they considered a 3-tape arrangement (for Input, Scratch Work, and Output). The paper [6] shows this can be reduced to 1-tape (if the alphabet has more than two symbols!). One can change the limit behaviour so that instead of a liminf value being declared for each cell's value, it simply becomes blank - for ambiguity - if it has changed value cofinally in the limit stage (Theorem 1 of [18]). Similarly the special state q_L is unnecessary: one may define the "next instruction" at a limit stage to be the instruction, or transition table entry, whose number is the liminf of the previous instruction numbers - this has the machine entering the outermost subroutine that was called cofinally in the stage. Likewise the Read/Write head may be placed at the cell numbered according to the liminf values of the cells visited prior to that limit stage (unless that liminf is now infinite, in which case we do return the head to the starting cell). ll of these variants make no difference to the functions computed.

We shall review the following facts related to such machines.

3.2 Infinite Time Turing Machine computation

Such ITTM's have two modes of producing results: a program can halt outright with an infinite string of $0, 1$'s on the part of the tape designated for output (the 'output tape') but it may also have some 'eventual output': the contents of the output tape may have settled down to a fixed value, whilst the machine is still churning away perhaps moving its head around and fiddling with the scratch portion of the tape. Nevertheless on a given fixed input (some $x \in {}^{\omega}2$ may be written to a designated portion of the tape, the 'input tape') any ITTM machine will eventually start to cycle - and by the starting point of that cycling, designated $\zeta(x)$, if the output settles down, then it will have done so by $\zeta(x)$.

This last feature is in fact, quite fundamental for the study of ITTMs. We may regard a machine $P_e(x)$ in this context, as having come to a conclusion - the contents of the output tape - but has not formally reached a halting state in the usual sense.

Definition 3.1. *We shall say that a computation $P_e(x)$ is* convergent to *y (and write $P_e(x)|y$) if it enters a halting state in the usual sense, or if it has eventually settled output. We shall say that "y* is (eventually)-ittm-recursive in *x". If it does not have settled output, we shall write $P_e(x)\uparrow$.*

This enshrines our taking *(eventually) settled output*, as the criterion of a successful computation. We shall be interested in eventual output of this sort, as well as the more restricted strictly halting variety. Both types of computation, the usual halting, and the 'eventually constant' output tape outlined above, we shall regard, and term, as 'convergent' - thinking of 'halting' as only a special kind of eventually settled output. Given a set $A \subseteq \omega \cup {}^{\omega}2$, this can be used as an oracle for an ITTM in a familiar way: ?*Is the integer on (or is the whole of) the current output tape contents an element of A?* and receive a $1/0$ answer for "Yes"/"No". We identify elements of ω as coded up in ${}^{\omega}2$ in some fixed way, and so may consider such A as subsets of ${}^{\omega}2$. But further: since having A respond with one $0/1$ at a time can be repeated, we could equally as well allow A to return an element $f \in {}^{\omega}2$ as a response (we have no shortage of time). We could then also allow as functionals also $A\colon {}^{\omega}2 \longrightarrow {}^{\omega}2$. However for the moment we shall only consider functionals into ω. Some examples follow.

Definition 3.2. *(The infinite time jump* iJ*)*

$$\begin{aligned} \mathrm{iJ}(e, \boldsymbol{m}, \boldsymbol{x}) &= 1 \text{ if } \{e\}(\boldsymbol{m}, \boldsymbol{x})\downarrow \text{ (\textit{i.e.} the } e\text{'th ittm-computable function with} \\ &\qquad\qquad \text{input } \boldsymbol{m}, \boldsymbol{x} \text{ has a halting value)} \\ &= 0 \quad \text{otherwise.} \end{aligned}$$

Definition 3.3. *(The eventual jump* eJ*)*

$$\begin{aligned} \mathrm{eJ}(e, \boldsymbol{m}, \boldsymbol{x}) &= 1 \text{ if } \{e\}(\boldsymbol{m}, \boldsymbol{x})| \text{ (\textit{i.e.} the } e\text{'th ittm-computable function with} \\ &\qquad\qquad \text{input } \boldsymbol{m}, \boldsymbol{x} \text{ has an eventually settled value)} \\ &= 0 \quad \text{otherwise (for which we write } \{e\}(\boldsymbol{m}, \boldsymbol{x})\uparrow). \end{aligned}$$

These are both total functionals. We shall be interested in functions recursive in eJ. But first we summarise some facts about ordinary ittm's.

Fact 1 [5] shows:

(i) That Π^1_1-predicates are decidable: given a code $x \in 2^{\mathbb{N}}$, there's an ittm that will decide whether $x \in \mathrm{WO}$ or not.

(ii) There's a program number e so that $P_e(x)$ will halt with a code for $(L_\alpha, \in)$ if $x \in \mathrm{WO} \wedge \|x\| = \alpha$.

(iii) For $z \in 2^{\mathbb{N}}$, the set of *ittm-writable-in-z* reals, is the set $\mathcal{W}^z \subseteq 2^{\mathbb{N}}$ where

$\mathcal{W}^z = \{x \in 2^{\mathbb{N}} | \, \exists e \, P_e(z)$ halts with output $x\}$.

(iv) The set of *ittm-eventually-writable-in-z* reals, is the set

$\mathcal{EW}^z = \{x \in 2^{\mathbb{N}} | \, \exists e \, (P_e(z)$ *has x written on its output tape from some point in time onwards*$)\}$.

Fact 2 [19] shows:

(i) Let (ζ, Σ) be the lexicographically least pair of ordinals so that $L_\zeta \prec_{\Sigma_2} L_\Sigma$. Let λ be the least ordinal with $L_\lambda \prec_{\Sigma_1} L_\zeta$. Then (*The "$\lambda$-$\zeta$-$\Sigma$-Theorem"*), $L_\lambda \cap 2^{\mathbb{N}} = \mathcal{W}$, $L_\zeta \cap 2^{\mathbb{N}} = \mathcal{EW}$. s is easily seen all three ordinals are limits of Σ_2-admissibles, whilst λ is Σ_1- but not Σ_2-admissible, and Σ is not admissible at all.

(ii) (a) ny computation $P_e(n)$ that halts (in the usual sense) does so by a time $\alpha < \lambda$.

(b) ny computation $P_e(n)$ that eventually has a settled output tape, does so by a time $\alpha < \zeta$.

(c) Both λ and ζ are the suprema of such fully "halting" times, and "eventual convergence" times, over varying $e, n \in \omega$, respectively.

(iii) $T^1_\lambda \equiv_1 h$, and $T^2_\zeta \equiv_1 \tilde{h}$ where $h = \{e|\ P_e(e)$ *reaches a halting state*$\}$ and $\tilde{h} = \{e|P_e(e)$ *eventually has settled output* $\}$.

(iv) It is a consequent of (iii) that a *universal machine* (on integer input) has *snapshots* of its behaviour which, when first entering a final loop at stage ζ, will repeat with the same snapshot at time Σ; moreover (1-1) in those snapshots is the theory T^2_ζ.

(v) *Recursion,* and Snm *Theorems* may be proved in the standard manner ([5]); there are appropriate versions of the Kleene *Normal Form Theorems* ([19]).

The usual argument shows:

Theorem 3.4. (The eJ-Recursion theorem) *If $F(e, \boldsymbol{m}, \boldsymbol{x})$ is recursive in* eJ*, there is $e_0 \in \omega$ so that*

$$\varphi^{\mathrm{eJ}}_{e_0}(\boldsymbol{m}, \boldsymbol{x}) = F(e_0, \boldsymbol{m}, \boldsymbol{x}).$$

3.2.1 More on Extendability

s Fact 2 (i) above shows, the relation of "L_ζ has a Σ_2-extension to L_Σ" is fundamental to this notion.

Fact 2 (contd.)

(vi) There is moreover a *theory machine* that writes codes for L_α and their Σ_ω-theories, and hence their Σ_2-theories, T^2_α, in an *ittm*-computable fashion for any $\alpha < \Sigma$, uniformly in α. If for $\mathrm{Lim}(\lambda)$ we write $\hat{T}_\lambda = {}_{\mathrm{df}}\mathrm{Liminf}_{\alpha \to \lambda} T^2_\alpha$, then there is a uniform index $e \in \omega$ that shows that $W_e^{\hat{T}_\lambda} = T^2_\lambda$, *i.e.* T^2_λ is r.e. in $\hat{T}_\lambda$ uniformly in λ. (See Lemma 2.5 of [22]. Moreover for those λ with $L_\lambda \models \Sigma_1$-Sep, $T^2_\lambda = \hat{T}_\lambda$.)

(vii) For the lexicographically least extendible pair (ζ, Σ), whilst $\omega_{1\mathrm{ck}}^{T^2_\zeta} < \Sigma$, it is the case that $\lambda(T^2_\zeta) > \Sigma$.

We make some further definitions concerning extendability.

Definition 3.5. *(The Σ_2-extendibility tree) We let $(\mathcal{T}, \prec)$ be the natural tree on such pairs under inclusion: as follows: if $(\zeta', \Sigma'), (\bar{\zeta}, \bar{\Sigma})$ are any two countable Σ_2-extendable pairs, then set $(\zeta', \Sigma') \prec (\bar{\zeta}, \bar{\Sigma})$ iff $\zeta' \leq \bar{\zeta} < \bar{\Sigma} < \Sigma'$.*

• If we had allowed the inequality $\bar{\Sigma} \leq \Sigma'$ rather than a strict inequalitiy in the last definition we could have defined a larger relation $\prec'$, and a larger tree $(\mathcal{T}', \prec')$; however this would not have been wellfounded: if $L_\Sigma \models \Sigma_2$-Sep then it is easy to see that $(\mathcal{T}' \restriction \Sigma + 1, \prec')$ is illfounded.

Lemma 3.6. *Let δ be least such that $L_\delta \models \Sigma_2$-Sep. ; let α be maximal so that $(\mathcal{T}' \restriction \alpha, \prec')$ is wellfounded (where $\mathrm{Field}(\mathcal{T}' \restriction \alpha) = {}_{\mathrm{df}}\{(\zeta, \Sigma)\,\mathrm{extendible}|\ \Sigma < \alpha\}$). Then $\delta = \alpha$.*

Proof: ($\leq$): Suppose $\delta > \alpha$. Then $(\mathcal{T}' \restriction \delta, \prec)$ is illfounded. So there is an infinite sequence of extendible pairs (ζ_n, Σ_n) with $(\zeta_{n+1}, \Sigma_{n+1}) \subset (\zeta_n, \Sigma_n)$. By wellfoundedness of the ordinals there is an infinite subsequence $(\zeta_{n_i}, \Sigma_{n_i})$ with all Σ_{n_i} equal to a fixed Σ, whilst $\zeta_{n_i} < \zeta_{n_{i+1}}$. Let $\zeta^* = \sup_i \zeta_{n_i}$. Then we have $L_{\zeta_{n_i}} \prec_{\Sigma_2} L_{\zeta_{n_{i+1}}} \prec_{\Sigma_2} L_{\zeta^*}$. Then ζ^* is not Σ_2-projectible, and hence $L_{\zeta^*} \models \Sigma_2$-Sep. But $\zeta^* < \delta$. Contradiction.

($\geq$): $L_\delta \models \Sigma_2$-Sep. Then S^2_δ is unbounded in δ. Let $\delta_i < \delta_{i+1}$ be a cofinal sequence, for $i < \omega$. Then check that $\langle (\delta_i, \delta) | i < \omega \rangle$ is a $\prec$-descending sequence in $\mathcal{T}' \restriction \delta + 1$. So $\alpha \leq \delta$. Q.E.D.

For E a class of ordinals, let E^* denote the class of its limit points.

Definition 3.7. *Define by recursion on* $0 < \alpha \in \mathrm{On}$ *the class* E^α *the class of* α*(-*Σ_2*)-extendible ordinals:*

$E^1 = \{ {}^1\zeta |\ {}^1\zeta$ is extendible but not a limit of extendibles$\}$*;*

$E^{\alpha+1} = \{ {}^\alpha\zeta |\ {}^\alpha\zeta \in (E^\alpha)^* \cap E^0 \}$*;*

$E^\lambda = \bigcap_{\alpha<\lambda} E^\alpha \cap E^0$.

$E^{\geq\alpha} = \bigcup_{\beta\geq\alpha} E^\beta$ *etc.*

Here we decorate the variable ζ with the prefix indicating its level of extendability. We shall let ${}^\alpha\Sigma$ indicate that for some ${}^\alpha\zeta$, $({}^\alpha\zeta, {}^\alpha\Sigma)$ is an α-extendible pair. Note that for any γ the least element of $E^{\geq\alpha}$ greater than γ is always an element of E^α, *i.e.* is α-extendible.

3.3 The Lengths of computations

We analyse the tree of subcomputations to define the notion of *absolute length* of the linearised *absolute computation* corresponding to some $P^I_e(\boldsymbol{m}, \boldsymbol{x})$.

Definition 3.8. *The* local length *of a computation* $P^I_e(\boldsymbol{m}, \boldsymbol{x})$ *in a type-2 oracle I, is the least* σ_0 *(when defined) so that the snapshot at* σ_0 *is the repeat of some earlier snapshot* $\zeta_0 < \sigma_0$*, and so that the snapshot at* σ_0 *recurs unboundedly in* On.

The local length has all the relevant information then in the calculation: everything thereafter is mere repetition (σ_0 will be undefined if $P^I_e(\boldsymbol{m}, \boldsymbol{x})$ is divergent, that is, has an ill-founded computation tree). nother description of it is as the "top level" length of the computation, which disregards the lengths of the subcomputation calls below it. We now describe a computation recursive in the type-2 functional eJ. In fact we give a representation in terms of ITTM's. $P^{\mathrm{eJ}}_e(\boldsymbol{m}, \boldsymbol{x})$ will represent the e'th program in the usual format with appeal to oracle calls possible. We are thus considering computation of a partial function Φ^{eJ}_e: ${}^k\omega \times {}^l({}^\omega 2) \to {}^\omega 2$. Such a computation may conventionally halt, or may go on for ever through the ordinals. The computation of $P^{\mathrm{eJ}}_e(\boldsymbol{m}, \boldsymbol{x})$ proceeds in the usual ittm-fashion, working as TM at successor ordinals and taking $\liminf$'s of cell values *etc.* at limit ordinals. t time α an oracle query may be initiated. We shall conventionally fix that the real being queried is that infinite string on the even numbered cells of the scratch type. If this string is $(f, y_0, y_1 \ldots,)$ then the query is ?*Does* $P^{\mathrm{eJ}}_f(y)$ *have eventually settled output tape*?, and at stage $\alpha + 1$ receives a $1/0$ value corresponding to "Yes/No" respectively. We thus regard eJ as the "eventual jump" and intend the following:

$$\mathrm{eJ} = \left\{ \langle \langle f, y \rangle, i \rangle |\ i = 1 \text{ and } P^{\mathrm{eJ}}_f(y) \right| \text{ or } \left. i = 0 \text{ and and } P^{\mathrm{eJ}}_f(y) \uparrow \right\}$$

Here, $P_f^{\mathrm{eJ}}(y)\uparrow$ denotes that the computation $P_f^{\mathrm{eJ}}(y)$ loops but has no settled output, it is *not* the notation for a computation whose tree has an ill-founded branch. (Compare with above for the type-2 recursion in J: *divergence* occurs if there is an illfounded-founded branch in the tree of evaluations.) s is intended, $P_f^{\mathrm{eJ}}(y)$ has the opportunity to make similar oracle calls, and we shall thus have a *tree* representation of calls made. We wish to represent the overall order of how such calls are made, and indeed the ordinal times of the various parts of the computation as it proceeds. Overall we have a 'depth first' mode of evaluation of a tree of subcomputations. We therefore make the following conventions. During the calculation of $P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$ (the topmost node ν_0 at Level 0, in our tree $\mathfrak{T}=\mathfrak{T}(\boldsymbol{e},\boldsymbol{m},\boldsymbol{x})$) let us suppose the first oracle query concerning $P_{f_0}^{\mathrm{eJ}}(y_0)$ is made at stage δ_0. We write a node ν_1 below ν_0, and explicitly allow the computation $P_{f_0}^{\mathrm{eJ}}(y_0)$ to be performed at this Level 1. The 'local time' for this computation, of course starts at $t=0$ - although each stage is also thought of as one more step in the overall computation of the computation immediately above: namely $P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$. Suppose $P_{f_0}^{\mathrm{eJ}}(y_0)$ makes no further oracle calls and the local length of $P_{f_0}^{\mathrm{eJ}}(y_0)$ is σ_1. Control, and the correct $1/0$ bit is then passed back up to Level 0, and the master computation proceeds.

We deem that $\delta_0+\sigma_1$ steps have occurred so far towards the final *absolute length* of the calculation $H=H(e,\boldsymbol{m},\boldsymbol{x})$, of $P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$.

However if $P_{f_0}^{\mathrm{eJ}}(y_0)$ has made an oracle query, let us suppose the first such was $?P_{f_1}^{\mathrm{eJ}}(y_1)?$, then a new node ν_2 is placed below ν_1. If this piece of computation at ν_2 takes σ_2 steps without oracle calls, to cycle before control and the result is passed back up to ν_1, (*i.e.* the local length of $P_{f_1}^{\mathrm{eJ}}(y_1)$ is σ_2) then those σ_2 steps will have to be be part of the overall length of calculation for $P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$ - although those σ_2 steps only counted for 1 step in the local length of $P_{f_0}^{\mathrm{eJ}}(y_0)$'s calculation. If the $P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$ converges then we shall have its computation tree $\mathfrak{T}=\mathfrak{T}(e,\boldsymbol{m},\boldsymbol{x})$, a finite path tree (with potentially infinite branching) and some countable rank. $\mathfrak{T}$ will be labelled with nodes $\{\nu_\iota\}_{\iota<\eta(\mathfrak{T})}$ that are visited by the computation in increasing order (with backtracking up the tree of the kind indicated). Thus ν_ι is first visited only after all ν_τ have been visited for $\tau<\iota$. The β'th oracle call to Level k will generate a node placed to the right of those so far at Level k (and thus to the right of those with lesser indices $\alpha<\beta$ at that level). The tree will thus have a linear leftmost branch, before any branching occurs.

Just as the Kleene equational calculus can be seen to build up in an inductive fashion a set of indices $\Omega[I]$ for successful computations recursive in I, (see Hinman [8], pp. 259-261) so we can define the graph of eJ as the fixed point of a monotone operator Δ on $\omega\times\omega^{<\omega}\times(\omega^\omega)^{<\omega}\times 2$.

$$\Delta(X)=\big\{\langle\langle e,\boldsymbol{m},\boldsymbol{x}\rangle,i\rangle \;\big|\; P_e^X(\boldsymbol{m},\boldsymbol{x}) \textit{ is an ittm-computation making, and receiving, only oracle calls } \langle\langle e',\boldsymbol{m'},\boldsymbol{x'}\rangle,i'\rangle\in X \textit{ with } i=1/0 \textit{ if the resulting output is eventually settled or not}\big\}.$$

Let $\Delta^0=\varnothing$; $\Delta^{\alpha+1}=\Delta(\Delta^\alpha)$; $\Delta^{<\lambda}=\bigcup_{\alpha<\lambda}\Delta^\alpha$ & $\Delta^\lambda=\Delta(\Delta^{<\lambda})$ in the usual way. Then the least fixed point of Δ is the function eJ.

Definition 3.9. *With* eJ *as just defined:*

$P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$ *is* convergent *if* $\langle e,\boldsymbol{m},\boldsymbol{x}\rangle\in\mathrm{dom}(\mathrm{eJ})$. *Otherwise it is* divergent.

ssuming $P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$ convergent, we may define by recursion a function $H(f_i,y_i)$ for $1\le\iota<\eta(\mathfrak{T})$, giving that absolute length of the calculation at node ν_ι taking into account the computations at nodes below it. Suppose the oracle queries made by $P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$ at Level 0, were $P_{f_{\iota_j}}^{\mathrm{eJ}}(y_{\iota_j})$ for $j<\theta$, and they were made at increasing local times δ_j for $j<\theta$ in $P_e^{\mathrm{eJ}}(\boldsymbol{m},\boldsymbol{x})$, then let $\bar{\delta_i}$ be defined by:

$\bar{\delta}_0 = \delta_0$;
$\bar{\delta}_{j+1} = \delta_{j+1} - \delta_j$;
$\bar{\delta}_\lambda = \delta_\lambda - \sup\{\delta_k \mid k < \lambda\}$.

then the *absolute length* of the calculation is the wellordered ordinal sum:

$$\begin{aligned} H(e, \boldsymbol{m}, \boldsymbol{x}) &= \textstyle\sum_{j=0}^{\theta} (\bar{\delta}_j + H(f_{\iota_j}, y_{\iota_j})) \text{ if } \theta > 0 \\ &= \Sigma(\boldsymbol{x}) \qquad\qquad\qquad\qquad \text{otherwise;} \end{aligned}$$

of course assuming by induction that the absolute lengths of the computations $H(f_{\iota_j}, y_{\iota_j})$ have been similarly defined.

We call the master computation $P_e^{\mathrm{eJ}}(\boldsymbol{m}, \boldsymbol{x})$ together with all the subcomputations of the tree explicitly performed, the *absolute computation* (as opposed to the top level 'local computation' with simple 1-step queries).

• It is possible, and easy, to design an index $f \in \omega$, so that $P_f^{\mathrm{eJ}}(0)$ has absolute length $H(f, 0, \varnothing)$ greater than the looping length of the top level computation. Hence for performing a computation together with all its subcomputations as a tree, and seeing how the absolute computation relates to extendability in the L hierarchy, this has to be done in suitably large admissible sets.

Lemma 3.10. *Suppose $P_e^{\mathrm{eJ}}(\boldsymbol{m}, \boldsymbol{x})$ is a convergent computation with tree $\mathfrak{T} \in M$, and with $\boldsymbol{x} \in M$, where M is a transitive admissible set. Let $\theta = \mathrm{On}^M$. Suppose for every node ν_ι in $\mathfrak{T}$ that the computation at the node $P_{f_\iota}^{\mathrm{eJ}}(y_\iota)$ has* local *length $\psi_\iota < \theta$ (this includes the local length of $P_e^{\mathrm{eJ}}(\boldsymbol{m}, \boldsymbol{x})$, being at ν_0, is some $\psi_0 < \theta$). Then $H(e, \boldsymbol{m}, \boldsymbol{x}) < \theta$.*

Proof: The required ordinal sum can be performed by an induction on the *rank* of the nodes in the tree, setting $0 = \mathrm{rank}(\nu_\iota)$, for those ι with ν_ι a terminal point of a path leading downwards from ν_0. This can be effected inside the admissible set M. Q.E.D.

Better:

Lemma 3.11. *Suppose $P_e^{\mathrm{eJ}}(\boldsymbol{m}, \boldsymbol{x})$ is a convergent computation with its computation tree $\mathfrak{T} \in M$, and with $\boldsymbol{x} \in M$, where M is a transitive admissible set, closed under the function $x \rightarrowtail \tilde{x}$. Let $\theta = \mathrm{On}^M$. Then $H(e, \boldsymbol{m}, \boldsymbol{x}) < \theta$.*

Proof: This is similar to the above. By induction on $\mathrm{rk}(\mathfrak{T}) = \eta < \theta$. Note first that the closure of M ensures that for all $y \in M$, that $\Sigma(y) < \theta$. Suppose true for all such trees of convergent computations $P_f(y)$ of smaller rank than η, for $y \in M$. Suppose $P_e^{\mathrm{eJ}}(x)$ makes queries at *local* times $\langle \delta_i | \, i < \tau \rangle$ to nodes at Level 1. Note that $\tau < \theta$ as $\mathfrak{T} \in M$. Suppose the calls are to the subtrees $\langle \mathfrak{T}_i | \, i < \tau \rangle$ with (f_i, y_i) passed down at time δ_i and $\bar{y_i}$ is the real passed up at local time $\delta_i + 1$. Let the snapshot at Level 0 at time γ be $s(\gamma)$. (Thus we assume $s(\delta_i + 1)$ contains the information of $\bar{y_i}$.) Now notice that $\delta_0 < \Sigma(x)$ (because the computation prior to δ_0 is (equivalent to) an ordinary ittm computation, which of course eventually converges at time $\Sigma(x)$.) If we set

$\bar{\delta}_0 = \delta_0$; $\quad s_0 = x$
$\bar{\delta}_{j+1} = \delta_{j+1} - \delta_j$; $s_{j+1} = s(\delta_j)$
$\bar{\delta}_\lambda = \delta_\lambda - \sup\{\delta_k \mid k < \lambda\}$; $s_\lambda = s(\sup\{\delta_k \mid k < \lambda\})$

then $\bar{\delta}_{j+1} < \Sigma(s_j)$ (as the time to the next query, if it exists, is always less than the least s_j-2-extendible by the same reasoning). Similarly $\bar{\delta}_\lambda < \Sigma(s_\lambda)$. By assumption on M, all such $\Sigma(s_j)$ are less than θ. Consequently if $H(f_i, y_i) = \theta_i < \theta$, the whole length of the computation is bounded:

$$H(e, \boldsymbol{m}, \boldsymbol{x}) = \sum_{i=0}^{<\tau} \bar{\delta}_i + \theta_i \leq \sum_{i=0}^{<\tau} \Sigma(s_i) + \theta_i < \theta.$$

Q.E.D.

Definition 3.12. *(i) The* Level *of the computation $P_e^{\mathrm{eJ}}(\boldsymbol{m}, \boldsymbol{x})$ at time $\alpha < H(e, \boldsymbol{m}, \boldsymbol{x})$, denoted $\Lambda(e, (\boldsymbol{m}, \boldsymbol{x}), \alpha)$, is the level of the node ν_ι at which control is based at time α, where:*

(ii) the level *of a node ν_ι is the length of the path in the tree from ν_0 to ν_ι.*

Thus for a convergent computation, at any time the level is a finite number ('depth' would have been an equally good choice of word). divergent computation is one in which $\mathfrak{T}(e, \boldsymbol{m}, \boldsymbol{x})$ becomes illfounded (with a rightmost path of order type then ω).

Lemma 3.13. *The computation $P_e^{\mathrm{eJ}}(x)$ converges if and only if there exists some x-Σ_2-extendible pair (ζ, Σ) so that $\Lambda(e, x, \zeta) = 0$.*

Proof: Suppose $P_e^{\mathrm{eJ}}(x)\big|$. If $P_e^{\mathrm{eJ}}(x)\downarrow$ conventionally then the conclusion is trivial as then for all sufficiently large x-Σ_2-extendible pairs $(\zeta,\ \Sigma)$, the machine has halted at Level 0. If otherwise, then the computation $P_e^{\mathrm{eJ}}(x)$ will loop forever through the ordinals. But, using the definition of the liminf behaviour at limit stages, it is easy to argue that there is a cub subset $C \subseteq \omega_1$ of points α, β with the snapshots of the computation at these times identical, and with $\Lambda(e, x, \alpha) = \Lambda(e, x, \beta) = 0$. Now find a pair $(\zeta,\ \Sigma')$ both in C, with $L_\zeta[x] \prec_{\Sigma_2} L_{\Sigma'}[x]$. Now minimise Σ' to a $\Sigma > \zeta$ with $L_\zeta[x] \prec_{\Sigma_2} L_\Sigma[x]$, thus (ζ, Σ) is as required.

Conversely: if it is the case that $P_e^{\mathrm{eJ}}(x)\downarrow$ the conclusion is trivial, so suppose otherwise and that (ζ, Σ) is some x-Σ_2-extendible pair satisfying the right hand side. By Σ_2-extendibility, $\Lambda(e, x, \Sigma)$ is also 0. By the liminf rule the snapshot of $P_e^{\mathrm{eJ}}(x)$ - which we can envisage running inside $L_\Sigma[x]$ - at time ζ is $\Sigma_2^{L_\zeta[x]}(x)$. gain by Σ_2-extendibility, it is the same at time Σ. Notice that any cell of the tape, C_i say, that changes its value even once in the interval (ζ, Σ), will, by Σ_2-reflection, do so unboundedly in both ζ and Σ. Consequently we have final looping behaviour in the interval $[\zeta, \Sigma]$. Hence we have our criterion for 'eJ-convergence'. Q.E.D.

Lemma 3.14. *Suppose we have a 2-nesting $\zeta_0 < \zeta_1 < \Sigma_1 < \Sigma_0$. Suppose at time ζ_0 of the absolute computation of $P_e^{\mathrm{eJ}}(m)$ either $P_e^{\mathrm{eJ}}(m)$ or a subcomputation thereof, is not yet convergent and is at level k of its computation tree. Then at time ζ_1 it is not yet convergent and control is at a level $\geq k+1$.*

Proof: Suppose $k = 0$. By Σ_2-reflection and the liminf rule, $P_e^{\mathrm{eJ}}(m)$ is still running, and control is still at depth k at Σ_0. This mean the snapshots at ζ_0 and Σ_0 are identical and thus $P_e^{\mathrm{eJ}}(m)$ has its first loop at (ζ_0, Σ_0), and the computation is convergent, and is then effectively over. Suppose for a contradiction that control is at level 0 also at ζ_1 (and again also at Σ_1). So again $P_e^{\mathrm{eJ}}(m)$ has looping snapshots at $(\zeta_1,\ \Sigma_1)$. However this is a Σ_1-fact about $P_e^{\mathrm{eJ}}(m)$ that L_{Σ_0} sees: "*There exists a 2-extendible pair $\big(\bar{\zeta}, \bar{\Sigma}\big)$ with $P_e^{\mathrm{eJ}}(m)$ having identical snaphots at level 0 at $(\bar{\zeta}, \bar{\Sigma})$.*" But then there is such a pair $\bar{\zeta} < \bar{\Sigma} < \Sigma_0$ and $P_e^{\mathrm{eJ}}(m)$'s computation is again convergent at $\bar{\Sigma}$ contrary to assumption.

The argument for $k \geq 1$ is very similar: if $\mathrm{liminf}_{\alpha \to \zeta_0} \Lambda(e,\ m,\ \alpha) = \Lambda(e,\ m,\ \zeta_0) = k$, then $\mathrm{liminf}_{\alpha \to \Sigma_0} \Lambda\big(P_e^{\mathrm{eJ}}(m), \alpha\big) = k$ also. gain, if it entered the interval (ζ_1, Σ_1) at this same level k it would loop there, and by the same reflection argument applied repeatedly would do so not just once but unboundedly below ζ_0 at the same level k. But after each successful loop at level k, control passes up to level $k-1$. However then $\mathrm{liminf}_{\alpha \to \zeta_0} \Lambda(e, m, \alpha) = k - 1$. Contradiction! Q.E.D.

Lemma 3.15. (Boundedness Lemma for computations recursive in eJ) *Let β_0 be the least infinitely nested ordinal in some ill-founded model M with $\mathrm{WFP}(M) = L_{\beta_0}$. Let α_0 be least with $L_{\alpha_0} \prec_{\Sigma_1} L_{\beta_0}$. Then any computation $P_e^{\mathrm{eJ}}(m)$ which is not convergent by time α_0, is divergent.*

Proof: Let $\zeta_0 < \cdots < \zeta_n < \cdots \ \beta_0 \cdots \subset s_n \subset \cdots \subset s_0$ witness the infinite nesting at β_0 in M. By the definition of α_0 no $P_e^{\mathrm{eJ}}(m)$ is convergent at a time $\alpha \in [\alpha_0, \beta_0)$ as this would be a Σ_1-fact true in L_{β_0}; but then by Σ_1-reflection, it is true in L_{α_0}. But if $P_e^{\mathrm{eJ}}(m)$ is not divergent before β_0, it will be by β_0: the previous lemma shows that $\Lambda(e, m, \zeta_n) < \Lambda(e, m, \zeta_{n+1})$ holds in M. But these level facts are absolute to V, as they are grounded just on the part of the absolute computation tree being built in L_{β_0} as time goes towards β_0 (and are not dependent on oracle information from eJ^M which perforce will differ from the true eJ); so $P_e^{\mathrm{eJ}}(m)$'s computation tree will have an illfounded branch at time β_0. Q.E.D.

The above then shows that the initial segment L_{α_0} of the L-hierarchy contains all the information concerning looping or convergence of computations of the form $P_e^{\mathrm{eJ}}(m)$. computation may then continue through the wellfounded part of the computation tree for the times $\beta < \beta_0$ but if so, it will be divergent. Relativisations to real inputs $\vec{x}$ are then straightforward by defining $\beta_0(\vec{x})$ as the least such that there is an infinite nesting based at that ordinal in the $L[\vec{x}]$ hierarchy *etc.*

Lemma 3.16. *Let $x \subseteq \omega$. Then $T^2_{\Sigma(x)}(x) =_{\mathrm{df}} \Sigma_2\text{-Th}(L_{\Sigma(x)}[x])$ is* eJ*-recursive in x.*

Proof: There is an index e so that running $P_e(x)$ asks in turn if $?n \in T^2_{\Sigma(x)}(x)?$ for each n, and will receive a $0/1$ answer from the oracle eJ. Consequently P_e may compute this theory on its output tape, and then halt. Q.E.D.

Remark: $T^2_{\Sigma(x)}(x) \equiv_1 \tilde{x}$ (by Fact 2 (iii) above).

Lemma 3.17. *Let $x \subseteq \omega$. Then a code for $L_{\Sigma(x)}[x]$ is* eJ*-recursive in x.*

Proof: There is a standard ittm program that on input $\tilde{x}$ will halt after writing as output a code for $L_{\Sigma(x)}[x]$. Thus, by the last remark and lemma, a code for $L_{\Sigma(x)}[x]$ is also eJ-recursive in x. Q.E.D.

Further:

• (i) For any e, x, the first repeating snapshot $s(e, x)$ of $P_e(x)$ is eJ-computable in x, as is a code for $L_{\rho_0}[x]$, $L_{\rho_1}[x]$ and $L_{\rho_1^+}[x]$ where ρ_0, ρ_1 are the ordinal stages of appearance of the first repeating snapshot $s(e, x)$, and ρ_1^+ is the least $\bar{\rho} > \rho_1$ which is a limit of $s(e, x)$-admissibles.

• We may thus have subroutines that ask for, and compute such objects during the computation of some $P_f^{\mathrm{eJ}}(y)$ say. Since satisfaction is also ittm-computable, we may query simply whether $?L_{\rho_1}[x] \vDash \sigma?$ and receive an answer.

One may show:

Theorem 3.18. *Any two of the functionals E,* eJ*, and* iJ *are mutually ittm-recursive in each other.*

Proof: This uses, in the direction to obtain iJ or eJ recursive in E, an appropriate version of the Normal Form Theorem from [19]. Q.E.D.

We collect together some of the above Facts and results, in order to abbreviate our descriptions of algorithms This will help to have a library of basic algorithms which we shall simply quote as being 'recursive in eJ' without further justification.

Definition 3.19. *(Basic Computations-*BC*) (i) Any standard ittm-computation $P_e(n,x)$ is Basic.*

(ii) If a code for an α ordinal is given, then the computations that compute: a) for any x (a code for) $L_\alpha[x]$ b) the satisfaction relation for $L_\alpha[x]$ is Basic (in (the code for) α); (and shows those objects are eJ*-recursive, if α is).*

The following are all eJ*-recursive, and Basic:*

(iii) The function $x \rightarrowtail \tilde{x}$;

(iv) The function that computes $x \rightarrowtail \Sigma(x)$, the larger of the next extendible pair in x;

(v) The function that computes $x \rightarrowtail \Sigma(x)^+$;

(vi) Any others that we may need to add.

Stronger ordinals than simply $\Sigma(x)^+$ can be eJ-recursive:

Lemma 3.20. *There is a recursive sequence of indices $\langle e_i | 0 \leq i < \omega \rangle$ so that for any $\alpha < \omega_1$ with a code $x \in 2^{\mathbb{N}}$, $P^{\mathrm{eJ}}_{e_i}(x)$ computes a code for the next i-extendible ${}^i\zeta > \alpha$.*

Proof: For $i = 0$ this has been done using Basic Computations. Suppose e_i has been defined, and we describe the programme $P^{\mathrm{eJ}}_{e_{i+1}}$. ssume without loss of generality that $\alpha = 0$, $x = \mathrm{const}_0$. Then $P^{\mathrm{eJ}}_{e_i}(0)$ computes a code for the least i-extendible, $\zeta_0 := {}^i\zeta$ say. By a basic computation let a slice of the scratch tape R be designated to hold $T^2_{\zeta_0}$; R: $=T^2_{\zeta_0}$. code for ζ_0 is recursive in $T^2_{\zeta_0}$. Now compute $P^{\mathrm{eJ}}_{e_i}(R)$. This yields the next i-extendible $\zeta_1 = {}^i\zeta_1$. Now, using Basic Computations, write successively to R the theories $T^2_{\zeta_0}, T^2_{\zeta_0+1}, \ldots, T^2_{\zeta_0+\beta}, \ldots$ for $\beta < \zeta_1$. We note that at limit stages $\lambda \leq \zeta_1$, R will contain "liminf" theories $\hat{T}_\lambda = \mathrm{Liminf}_{\alpha\to\lambda} T^2_\alpha$ (by the usual automatic ittm liminf process) but that T^2_λ is uniformly r.e. in $\hat{T}_\lambda$. (For the latter see Fact 2. It is easy to argue that $\hat{T}_\lambda \supseteq T^2_\lambda$, and that if $\mathrm{supS}^1_\lambda = \lambda$ then we have equality, it is the bounded case of S^1_λ in λ that requires argument. The point of this exercise of writing theories to R is to ensure continuability of the computation, and that we do not start to loop too early. The 'writing out' of all levels of the theories to R, is a precautionary step: in general we do not have $\hat{T}_{i+1\zeta} = \mathrm{liminf}_{i\zeta\to i+1\zeta} \hat{T}_{i\zeta}$.) nd again a code for λ is then recursive in T^2_λ.

Set $R :=$ $\hat{T}_{\zeta_1}$; by the comments just made $T^2_{\zeta_1}$ is r.e. in R and $R \in L_{\zeta_1+1}$ (this is why we are writing out these theories, to ensure that we loop at our desired target); now compute $P^{\mathrm{eJ}}_{e_i}(R)$ and repeat this process. s there is no means for the machine to halt, there is a least looping pair (ζ, Σ) of ordinals. Let $({}^{i+1}\zeta, {}^{i+1}\Sigma)$ be the least $i+1$-extendible pair. We claim that this is the pair (ζ, Σ). Suppose $\zeta < {}^{i+1}\zeta$. By the repetition of the contents of R in the loop points, we have $\hat{T}_\zeta = \hat{T}_\Sigma$, in the above algorithm, hence $T^2_\zeta = T^2_\Sigma$, and thus $L_\zeta \prec_{\Sigma_2} L_\Sigma$. But then ζ is an extendible limit of i-extendibles, as ζ is a limit point of this looping process. This contradicts the minimality of ${}^{i+1}\zeta$. Hence ζ equals the latter, and $\Sigma = {}^{i+1}\Sigma$ follows.

Hence we may compute $\hat{T}_{i+1\zeta}$, ${}^{i+1}\zeta$ by means of an eventually stabilizing looping programme. We let $P^{\mathrm{eJ}}_{e_{i+1}}$ be the programme just described followed by the basic comp. that finds a code for ${}^{i+1}\zeta$ by a method uniformly r.e. in $\hat{T}_{i+1\zeta}$.

Finally note that the continuing description of the programme $P^{\mathrm{eJ}}_{e_{i+2}}$ from $P^{\mathrm{eJ}}_{e_{i+1}}$ merely repeats the above but altering only a few indices. We may thus determine a recursive function $i \mapsto e_{i+1}$. Q.E.D.

Entirely similar is:

Lemma 3.21. *There is a (Turing) recursive sequence of indices $\langle e'_i | \, i < \omega \rangle$ so that $P^{\mathrm{eJ}}_{e'_i}(x)$ writes a code for ${}^i\Sigma(x)$, the least Σ_2-extension of $L_{i\zeta}[x]$.*

4 The determinacy results

We shall assume a certain amount of familiarity of working with ittm's and shortcuts amounting to certain subroutines, so as not to overload the reader with details.

Theorem 4.1. *For any Σ^0_3 game $G(A;T)$, (with T say recursive) if player I has a winning strategy, then there is such a strategy recursive in* eJ*; if player II has a winning strategy, then there is such a strategy either recursive in* eJ*, or else definable over L_{β_0}.*

Proof of 4.1

Idea: We suppose $A = \bigcup_n B_n$ with each $B_n \in \Pi^0_2$, with an initial game tree T. For expository purposes we shall assume that $T = {}^{<\omega}\omega$ - relativisations will be straightforward. We shall provide an outline of a procedure which is recursive in eJ and which will either provide a strategy for *I* in $G(A;T)$ (if such exists) or else will diverge in the attempt to find a strategy for *II*. We wish to apply the main Lemma 3 of [20] for the successive B_n. The control of the procedure will be at different *Levels* of the initial finite path tree of the computation. t Level 0 will be the main process, but also the procedure for finding witnesses and strategies involved in the arguments for the Main Lemma applied with $B = B_0$. We first search for a level in the L-hierarchy whose code is eJ-recursive and for which we can define a non-losing subtree $T' \subseteq T$, for which all $p \in T'$ have witnesses $\hat{T}_p$ to p's goodness in the sense of (i) and (ii) above. In fact we shall search for pairs of levels in the L-hierarchy, in the sequel, between which we have absoluteness of our non-losing subtrees. fter having found such, this data will be encoded as a real (these routine details, the reader will be pleased to learn, we omit) and a subroutine call made to a process at the *lower Level 1* which will attempt to find the right witnesses *etc.* to apply the Lemma for $B = B_1$. We now search for a further level of the L-hierarchy which again has the right witnesses to goodness to all the possible relevant subtrees associated with positions p_2 of length 2. s we search for such an L_α, we may find that some of our original witnesses to goodness at Level 0 no longer work in our new L_α, or even more simply that our T' from Level 0 now has nodes p which have become winning for *I* in this L_α. We accordingly keep testing the data handed down to see if any of it has become 'faulty' in this respect. If so, then we throw away everything we have done at Level 1, but pass control back up to Level 0 together with the ordinal height of the current L_α we reached. We then go back to searching for an $L_{\alpha'}$ which is 'good' in all of these previous respects at Level 0 for a new T', which we then shall pass down to Level 1 for another attempt.

Eventually we shall reach a stage where we have a sufficiently large model where all the data and our witnessing subtrees work at both levels 0 and 1. ccordingly again all *this* data is passed down to the subroutine at *Level 2* for assessing potential subtrees for application in the Lemma to be applied for $B = B_2$. Proceeding in this fashion, testing as we go the validity of our data trees *en route* and passing back up to the Level of the tree that has failed if so, we find we work at increasing depth - that is at lower Levels n with increasing n. If *II* has a winning strategy then there will be an infinite path descending through all the Levels and hence the computation will diverge. One point will be to remark that if I has a winning strategy then this process will discover it: this requires us checking that we don't come up against a 'wall' in the ordinals α so that we cannot find a code for an ordering of a longer order type - because our computation has stabilized, or in other words is in a loop, and we are stuck below the length of that loop.

Hence if there is no such wall, and $G(A;T)$ has a winning strategy for *II* only definable over L_{β_0}. then we can theoretically keep computing ordinals up to β_0.

Our task now is to achieve a balance between giving enough of these details that the reader is convinced, and without causing the eye to glaze over with overwhelming (and unnecessary) *minutiae*.

In general: given a tree S in a model M, used in a game $G(\bar{A}, S)$, and without a strategy for player I in M, then we shall denote the subtree of non-losing positions for *II* in M by S'^M (or just S'). For $R \in \mathcal{P}(\mathbb{N})$, $\tau^+(R)$ will denote the sup of the first ω many R-admissibles beyond τ. By $\Sigma^k(R)$ we shall mean, where $\zeta^k(R)$ is the least k-extendible in the $L_\alpha[R]$ hierarchy, that $\Sigma^k(R)$ is the least ordinal with $L_{\zeta^k(R)}[R] \prec_{\Sigma_2} L_{\Sigma^k(R)}[R]$. If $k=1$ we drop it and write simply $\zeta(R)$ *etc.* We note that if $L_{\Sigma^+(R)}[R]$ has no proper Σ_1-substructures, then $T^1_{\Sigma^+(R)}[R] =_{\rm df} \Sigma_1$-Th$(L_{\Sigma^+(R)}[R])$ - in the language of set theory with a predicate symbol for R - is not in $L_{\Sigma^+(R)}[R]$; moreover (ordinarily) recursive in $T^1_{\Sigma^+(R)}[R]$ is a wellorder of type $\Sigma^+(R)$. We shall let the notation ${}^\alpha M$ vary over structures of the form $L_{\alpha_{\Sigma^+}}[T]$.

[Commentary is provided in square brackets following a % sign.]

s a warm-up we prove the following lemma using just Basic Computations.

Lemma 4.2. *There is a computation that on input codes for T, $\langle B_n \rangle$ will halt either with a winning strategy for I, or else with an encoded T' - the set of non-losing positions for II in $G(A;T)$- membership of which is absolute between some $L_\zeta[T]$ and $L_{\Sigma^+}[T]$.*

(0): *We commence with cutting up recursive infinite disjoint slices of the scratch tape to be reserved as 'registers' for the reals coding $\langle B_n \rangle$, $T, T', \Sigma^+, \ldots$*, (and more such will be needed at lower Levels, as data is passed down in the argument that follows, but we shall not mention these, rather leave it to the reader to do the preparatory mental scissor work).

• *Set:* $T' := T$.

(1) • *Compute:* $M := L_{\Sigma^+(T')}[T']$, and set $\Sigma^+ := \Sigma^+(T')$.

• $?T'^M = \varnothing?$ *If* $T'^M = \varnothing$ *then* $(I$ *has a winning strategy in* $G(A;T))^M$*, and this may be found in* M *and printed out on the output tape; then* STOP. Otherwise CONTINUE.

[% s M is a model of KPI such a winning strategy is winning in V.]

• ? *Is* $T'^M = T'$?

(2) • *If* NO *then* $T' \supset T'^M$ *and then some winning strategies are newly available to* I *in* M *that are for some* $p \in T' \backslash T'^M$. *Set* $T' := T'^M$; GOTO (1).

[% Note that the new T' is a proper subtree of the old.]

• *If* YES, *then we may* STOP *with a suitable* T' *encoded in its register.*

[% Of course in order to obtain M, Σ^+, *etc.* this officially requires a call to a subcomputation at the next level down, but the above is just a schematic description of the process, and so we suppress that level of detail. The point is that the T' are a decreasing sequence of sets. Hence keeping track of these T' at the top level suffices for the procedure to continue: we don't need to keep track of, *e.g.*, the ordinals heights of the structures M, and the concomitant worries about the liminf action at limit stages. Thus the above can be all effected using Basic Computations (and variants thereon).]

Claim 1 Either the program halts with a winning strategy for I in $G(A;T)$ or, at some point strictly before the next 2-extendible above $\Sigma^+(T)$ in the cycle, the answer to the query ? Is $T'^M = T'$? is affirmative.

Proof: Note first that the computation uses only BC's and each of these only require a computation of length the next extendible pair at most. Suppose (ζ_0, Σ_0) is any extendible pair that is a limit of such, above $\Sigma^+(T')$. We imagine the computation as being performed as a Σ_2-recursion in T in L_{Σ_0}. Then suppose, for a contradiction, that by the ζ_0'th turn through the cycle, we have not had an affirmative answer. In the ν'th turn through the cycle (for $\nu < \zeta_0$) let T' be denoted by T'_ν. Then the T'_ν, as remarked, are strictly decreasing. Now by an easy reflection argument, one sees that on a tail of $\nu < \zeta_0$, the T'_ν must be the same. [If "$\forall\nu\exists\nu' > \nu\exists p(p \in T'_{\nu'} \backslash T'_{\nu'+1})$" holds in L_{ζ_0} it will also hold in L_{Σ_0}. But if $p_0 \in T'_{\nu'} \backslash T'_{\nu'+1}$ the ν' for which that happens is Σ_2-definable in L_{Σ_0} from p_0; but that implies $\nu' < \zeta_0$. This contradicts the quoted formula.] So an affirmative answer must have occurred. Q.E.D. Claim 1 & Lemma]

We now assemble these building blocks to form a programme based on the argument of the proof of Theorem 2.7 surveyed above.

Proof of Theorem 4.1:

We outline the argument at the various levels of computation in the oracle calls of a master computation at level $\Lambda = 0$. We proceed by describing the actions of the programmes being called, which the reader may reformulate as official queries to the eJ-functional as oracle. t the end of the description we justify the claim that this is a bona fide eJ-recursion.

(1) $\Lambda = 0$.

• The master or control programme computes successively lengthening structures ${}^1M = L_{\Sigma^+}[T']$ until T' is seen to stabilize between one such structure ${}^1M$ and the next, ${}^1M'$.

[% This we saw done effectively by a machine in the proof of Lemma 4.2, with T' so stabilizing before the next 2-extendible. This process involved oracle queries to Level $\Lambda = 1$, but again we suppress these details.]

• With T' stabilized, the programme asks the following - when suitably formulated - oracle query of eJ. The query sub-computation we view as enacted at $\Lambda = 1$. We suppose that it is the computation $P^{\mathrm{eJ}}_{e_0}(x)$ where $x = \langle 1, \langle B_n\rangle_n, \mathbb{T}^0, {}^1M\rangle$ (suitably coded), whose action is described below starting at (2).

Q^1: ? *Defining $\mathbb{T}^1$ from the current T' in $\mathbb{T}^0$, do all the trees in $\mathbb{T}^1$ become eventually settled* ?

[% Recall that the trees of $\mathbb{T}^1$ are of the form:

a) $\hat{T}_p$ ($=_{\mathrm{df}}$ the current ${}^1M$-least witness to the goodness of $p \in T'$) and

b) $\left(\hat{T}_p\right)'$ ($=_{\mathrm{df}}$ its tree of non-losing positions for *II*); as well as (where $T^*(\varnothing)$ is set to $\hat{T}_\varnothing$)

c) $T^*(\varnothing)_{p_2}$ and $((T^*(\varnothing))_{p_2})'$ for relevant p_2.

We adopt the convention, that " $\mathbb{T}^l$ becomes eventually settled " or " $\mathbb{T}^l$ is stable up to ordinal τ" to be a shorthand affirming that all the constituent trees of the family $\mathbb{T}^l$ are stable *per* their definitions up to τ.

Note also: that since T' has survived intact from one ${}^1M$ structure to the next ${}^1M'$ say, we can deploy the 'survival argument' of Lemma 2.12; this means that both structures see that all $p \in T'$ are good, and this is a sufficient criterion for the definition of $\mathbb{T}^1$ over ${}^1M$ to instantiate all the needed trees, which then exist in ${}^1M$ (indeed $(\mathbb{T}^1)^{{}^1M} \subseteq \left(L_{1_\zeta}\right)^{{}^1M}$). Hence the query is therefore immediately meaningful.]

(2) $P^{\mathrm{eJ}}_{e_0}(x)$ answers the query by first taking from x the current data, and on seeing the initial flag 1, computes successive models ${}^1M$, and keeps a register of the successive theories T^2_α, of increasing ordinal height in the manner of the proof of Lemma 3.20. These operations are using our BC's.

If (Case 0): n ${}^1M$ is reached that contains a winning strategy σ for I in $G(A;\, T)$ then the programme H LTS and passes $x' = \langle \sigma \rangle$ back up to the master programme at $\Lambda = 0$;

If (Case 1): T' changes from one structure ${}^1M$ to the next ("T' becomes unstable") then the programme H LTS and with the current $\mathbb{T}^0 = \langle T,\, T'^{{}^1M} \rangle$, passes the current $x' = \langle 1,\, \langle B_n \rangle_n,\, \mathbb{T}^0, {}^1M \rangle$ back up to the master programme at $\Lambda = 0$; and RETURNS TO (1);

If (Case 2): T' remains stable but some $S \in \mathbb{T}^1$ does not by the end of the eventual loop in $P^{\mathrm{eJ}}_{e_0}(x)$, then the answer to Q^1 is "No" (or "0") and $x' = \langle 0 \rangle$ and control are passed back up to the master programme at $\Lambda = 0$.

In Case 0, the Master programme halts with this σ as output.

[% note that as M is closed under admissibles, σ is a w.s. for I in V.]

In Case 1, the Master programme continues to calculate successive models, re-starting from the M passed up in x'.

In Case 2, the Master programme, on receiving "No", and using BC's, computes the length of the loop just passed, call it Σ, and then continues calculating successive models, with the first such in this series containing the ordinal Σ.

[% Note that: ( ) T' must become eventually settled under the repeated calculation of longer M's by the time of the next (or indeed any) larger element ${}^2\zeta \in E^2$, or ${}^\alpha\zeta$ ($\alpha \geq 2$) for that matter. Hence the loop (1) $\longrightarrow$(2) (Case 1) $\longrightarrow$(1) will be broken out of by the time the length of the models M approaches the next ${}^2\zeta$.

(B) For Case 2: we cannot immediately deploy a shrinking argument on the trees to conclude that we have stability of all trees in $\mathbb{T}^1$ by the next extendible, since the actual underlying trees $\hat{T}_p$, $T^*(\varnothing)_{p_2}$ may be changing. However the eventual loop Σ whose length the Master programme computes, is that of a 2-extendible in E^2; this is ensured by the writing out of the theories T^2_α in the manner of the argument of the proof of Lemma 3.20. If the loop (1) $\longrightarrow$(2) (Case 2) $\longrightarrow$(1) repeatedly occurs from some point on, then for all sufficiently large ${}^2\Sigma$ below the next ${}^3\zeta$ (and so also by Σ_2-reflection, below the ${}^3\Sigma$ corresponding to ${}^3\zeta$).There is $\mathbb{T}^1$ so that (for all sufficiently large ${}^1\Sigma < {}^2\Sigma\big)$ ($\mathbb{T}^1 = (\mathbb{T}^1)^{L_{1_\Sigma}}$) and so we shall end up in Case 3 below.]

The last possibility is:

(Case 3): ll $S \in \mathbb{T}^1$ become stable between two successive ${}^1M$-structures, M_1, M_2.

The sub-computation now makes in turn a further query sub-computation which in turn we view as enacted at $\Lambda = 2$. We suppose that it is the computation $P^{\mathrm{eJ}}_{e_0}(x)$ where we collect the current values

$$\mathbb{T}^1 = \langle \langle \hat{T}_p |\, p \in T' \rangle, \langle (\hat{T}_p)' |\, p \in T' \rangle, \langle T^*(\varnothing)_{p_2}, (T^*(\varnothing)_{p_2})' |\, p_2 \text{ relevant} \rangle \rangle$$

and set:

$$x = \langle 2, \langle B_n \rangle_n, \mathbb{T}^0, \mathbb{T}^1, M_1 \rangle;$$

and where the query is:

Q^2: ? *Defining* $\mathbb{T}^2$ *from the current* $\mathbb{T}^0$, $\mathbb{T}^1$ *of* x, *do all the trees* S *in* $\mathbb{T}^2$ *become eventually settled* ?

[% Just as following Q^1, the stability of all the trees $\left(\hat{T}_p\right)'$ and $(T^*(\varnothing)_{p_2})'$ from one model to the next guarantees the existence of all the trees of $\mathbb{T}^2$ by the survival argument.]

(3) $P^{\mathrm{eJ}}_{e_0}(x)$ is programmed so that when it takes from x the current data, and sees the initial flag 2, it will continue to compute successive models ${}^2M$, (which it can by Lemma 3.21) and write out theories as before, using Basic Comps, but now act as follows.

If (Case 0): ${}^2M$ contains a winning strategy σ for I in $G(A;T)$ then this sub-computation H LTS and passes $x' = \langle\sigma\rangle$ back up to the programme at $\Lambda = 1$;

If (Case 1): T' becomes unstable, then the subcomputation H LTS and passes the current $x' = \langle 0, T, T', {}^2M\rangle$ back up to the programme at $\Lambda = 1$;

If (Case 2): T' remains stable but some $S \in \mathbb{T}^1$ does not at some stage, between two successive models ${}^2M_1$, ${}^2M_2$, then the subcomputation H LTS and the current $x' = \langle 2, \langle B_n\rangle_n, (\mathbb{T}^0, \mathbb{T}^1)^{{}^2M_2}, {}^2M_2\rangle$ with the current values of the data, and control, are passed back up to $\Lambda = 1$;

If (Case 3): T' and all $S \in \mathbb{T}^1$ remain stable but some $S \in \mathbb{T}^2$ do not, then the answer to Q^2 is "No".

In Cases 0,1 the relevant information will be passed up in turn to the master computation at $\Lambda = 0$ and will be acted on appropriately.

In Case 2, the sub-computation at $\Lambda = 1$, restarts using BC's, and computes structures ${}^1M$ as at (2).

In Case 3, the sub-computation at $\Lambda = 1$, is programmed to use BC's, to compute the length of the loop just passed, say to Σ, and then continues calculating successive models in the usual manner as at (2), with the first such in this series containing the ordinal Σ.

[% Note that: the comments on the loops at ( ), (B) will hold here. dditionally:

(C) If the loop (2) $\longrightarrow$(3) (Case 3) $\longrightarrow$(2) occurs from some point on, then for sufficiently large ${}^3\Sigma$ below the next ${}^4\zeta$, (and so also by Σ_2-reflection, below the corresponding ${}^4\Sigma$) there is $\mathbb{T}^2$ so that (for sufficiently large ${}^2\Sigma < {}^3\Sigma$ $(\mathbb{T}^2 = \mathbb{T}^2)^{L_{2_\Sigma}}$) and so we shall end up in Case 4 below.

The last possibility is:

(Case 4): $\mathbb{T}^2$ becomes stable between two successive ${}^2M$-structures, M_1, M_2.

gain, the current sub-computation makes a query sub-computation which in turn we view as enacted at $\Lambda = 3$. We suppose that it is the computation $P^{\mathrm{eJ}}_{e_0}(x)$ where we set

$$\mathbb{T}^2 = \langle \hat{T}(p_2)_p, \hat{T}(p_2)'_p |\ p \in (T^*(\varnothing)_{p_2})' \rangle, \langle T^*(p_2)_{p_4}, (T^*(p_2)_{p_4})' |\ p_4 \text{ relevant} \rangle \qquad \text{and}$$
$$x = \langle 3, \langle B_n\rangle_n, \mathbb{T}^0, \mathbb{T}^1, \mathbb{T}^2, M_2\rangle$$

and where the query is:

Q^3: ? *Defining* $\mathbb{T}^3$ *from the current* $\mathbb{T}^0, \mathbb{T}^1, \mathbb{T}^2$ *in* x*, does* $\mathbb{T}^3$ *become eventually settled* ?

We hope the reader will have seen the pattern emerging in this description of the programme P_{e_0}. However the reader is entitled to ask: have we described a genuine programme for such oracle machines? nd secondly, what is the outcome?

typical 3-nesting diagram is at Figure 1 below. T' is assumed to be stable up to Σ_3. Thus beyond the branch given, there are no winning strategies for *I* for any T'_p for any $p \in T$ appearing in the interval beyond the branch point up to Σ_3 (but such may appear in L_{Σ_3+1}). Because T' is this long-lived at positions labelled P, we can have all the relevant trees $\big(\hat{T}_p\big)'$, $T^*(\varnothing)_{p_2}$ and $((T^*(\varnothing))_{p_2})'$ (*i.e.* $\mathbb{T}^1$) occurring, and themselves are stable up to the end of the extendible loop below which they occur. t the first 2-nesting illustrated because all the $((T^*(\varnothing))_{p_2})'$ at P survive to the end of the outermost nesting, and so *beyond* the top of the inner nesting, we may conclude that at a position such as Q, all the relevant trees $T^*(p_2)_{p_4}, (T^*(p_2)_{p_4})'$ of $\mathbb{T}^2$ occur below the inner extendible ζ that starts the inner nesting loop. The analysis at the 3-nesting is similar: since T' survives beyond Σ_2, the $\mathbb{T}^1$ trees can be found at locations P; as the $\mathbb{T}^1$ trees, $\big(\hat{T}_p\big)'$, $T^*(\varnothing)_{p_2}$ and $((T^*(\varnothing))_{p_2})'$, survive beyond Σ_1, the $\mathbb{T}^2$ trees can be found at locations Q. If we had assumed that T' survived beyond Σ_1 then we could have obtained a shift, with the $\mathbb{T}^1$ trees obtainable at R, the $\mathbb{T}^2$ trees at P and then gone on to find the $\mathbb{T}^3$ trees at Q.

We could easily enough have written down Q^{k+1} which, given $\mathbb{T}^0$, ..., $\mathbb{T}^k$ from an x would have formulated definitions for $T^*(p_{2(k-1)}) =_{\mathrm{df}} \hat{T}(p_{2(k-1)})_\varnothing$, relevant p_{2k}, and then asked if trees $\hat{T}(p_{2k})_p$ (being the current ${}^{k+1}M$-least witness to the goodness of $p \in T^*(p_{2(k-1)})_{p_{2k}}\big)'$ and $\hat{T}(p_{2k})'_p$ (the latter's subtree of non-losing positions for *II*), that is the trees of $\mathbb{T}^{k+1}$, became eventually settled. The required definitions and stability questions are then entirely uniform in k. Hence the instructions for the programme P_{e_0} on input an x coding some $\langle k+1, \langle B_n\rangle_n, \mathbb{T}^i\ (i \leq k), {}^k M\rangle$ may be effectively written down in terms of k and the given tuple of data. It is enacted by considering successive ${}^{k+1}M$ structures, and by writing down theories T^2_α as before. The number of Cases to be considered at query Q^{k+1} is $k+3$: Cases (0)-(k) result in a H LT at that level $\Lambda = k+1$, with an effectively determined x' to be passed up to the level $\Lambda = k$ above; whilst Case $k+2$ requires returning to $\Lambda = k$ and computing lengths of loops *etc.* The final Case $k+3$ is the one of eventual interest and triggers the query Q^{k+2}. Each Q^{k+1} is officially a query of the form $?\mathrm{eJ}((e'_{k+1}, x)) = 0/1?$ about how the next subcomputation loops, and we calculate the relevant x from our data. The instructions that e'_{k+1} codes include of course those for calculating e'_{k+2} ready for the next query. However we may argue as below, that these calculations may be assembled into, or considered as, one whole calculation embodied in one $\varphi^{\mathrm{eJ}}_{e_0}$.

In the following we let "$\forall^* \alpha < \Sigma\varphi(\alpha)$" abbreviate "For all sufficiently large $\alpha < \Sigma\,\varphi(\alpha)$". We shall say "$T'$ is stable up to ${}^k\Sigma$" to mean "$T'^{L_{\zeta(M)}} = T'^{L_{\Sigma^+(M)}}$" for all sufficiently large structures M with $\Sigma^+(M) < {}^k\Sigma$. This can be equivalently written as "$\exists U(\forall^{*\,k-1}\Sigma < {}^k\Sigma)\Big[U' = (T')^{L_{\,k-1\Sigma}}\Big]$."

For $0 < l < k$ we shall say "$\mathbb{T}^l$ is stable up to ${}^k\Sigma$" to mean "$\exists \mathbb{T}^l \in L_{k_\Sigma}\Big(\mathbb{T}^l = \mathbb{T}^{l^{L_{\Sigma^+(M)}}}\Big)$" for all sufficiently large structures M with $\Sigma^+(M) < {}^k\Sigma$, which, as we have indicated above, of course is taken, by a convention, to be a shorthand affirming that all the constituent trees of $\mathbb{T}^l$ are stable *per* their definitions up to ${}^k\Sigma$.

In the above definition of the algorithm we are employing the following principle:

Suppose T' is stable up to some ${}^k\Sigma$, then
for all sufficiently large ${}^{k-1}\Sigma < {}^k\Sigma$ ($\mathbb{T}^1$ is stable up to ${}^{k-1}\Sigma$ &
for all sufficiently large ${}^{k-2}\Sigma < {}^{k-1}\Sigma$ ($\mathbb{T}^2$ is stable up to ${}^{k-2}\Sigma$ & ...
$\vdots$
for all sufficiently large ${}^2\Sigma < {}^3\Sigma$, $\mathbb{T}^{k-2}$ is stable up to ${}^2\Sigma$ &
for all sufficiently large ${}^1\Sigma < {}^2\Sigma$, $\mathbb{T}^{k-1}$ exists)$\cdots$)".

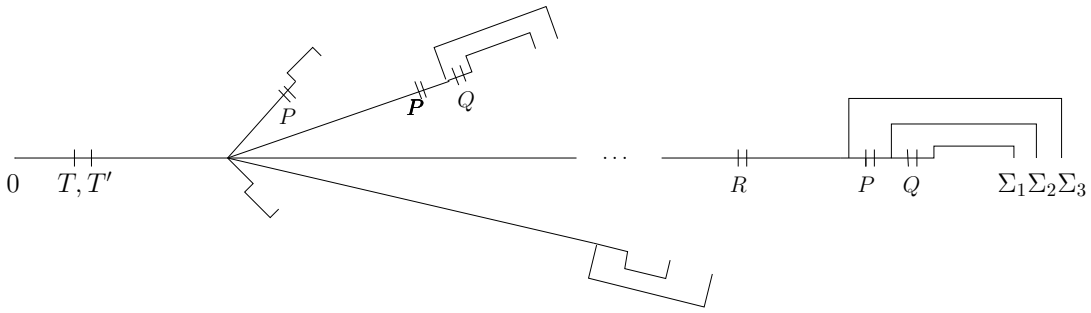

Figure 1 In this diagram T' is stable up to (but not beyond) Σ_3.

Less perspicuously but more formally we state this as:

Lemma 4.3. *Suppose T' is stable up to some ${}^k\Sigma$, then*
$(\forall^{*\,k-1}\Sigma < {}^k\Sigma)(\exists\mathbb{T}^1)(\forall^{*\,k-2}\Sigma < {}^{k-1}\Sigma)$
$\Big[\mathbb{T}^1 = (\mathbb{T}^1)^{L_{\,k-2\Sigma}} \wedge (\exists\mathbb{T}^2)(\forall^{*\,k-3}\Sigma < {}^{k-2}\Sigma)\Big[\mathbb{T}^2 = (\mathbb{T}^2)^{L_{\,k-3\Sigma}} \wedge (\exists\mathbb{T}^3)(\dots)\cdots$
$\dots(\exists\mathbb{T}^{k-2})(\forall^{*\,1}\Sigma < {}^2\Sigma)\ \big(\mathbb{T}^{k-2} = (\mathbb{T}^{k-2})^{L_{\,1\Sigma}} \wedge (\exists\mathbb{T}^{k-1})\big(\mathbb{T}^{k-1} = (\mathbb{T}^{k-1})^{L_{\,1\Sigma}}\big)\big]\big]\cdots\big]\big].$

Proof: Formally by induction on k, but the reader may convince themselves of a representative case, say with $k = 3$. Q.E.D.

Note 4.4. The Lemma is really the formal counterpart of the description that precedes it. Note that the hypothesis here is fulfilled whenever ${}^k\Sigma$ approaches some ${}^{k+1}\Sigma$: for sufficiently large ${}^k\Sigma$ below ${}^{k+1}\Sigma$, T' will be stable even beyond ${}^k\Sigma$.

lso, by the usual Σ_2-reflection arguments, the above principles are equivalent to those obtained by replacing any string "$< {}^l\Sigma$" by "$< {}^l\zeta$".

s the program runs there will eventually be subcomputation calls to arbitrary levels, as it uses various trees for as long as they survive fulfilling their role. But only after α_0 stages will we be certain that T' really does stabilize to its final value. Thereafter we shall have $\Lambda(e_0, T, \alpha) > 0$. t a later point we shall have all the correct trees to apply the Main Lemma once, and these will survive. fter such a point $\Lambda(e_0, T, \alpha)$ is greater than 1. But only at β_0 do we first have $\mathrm{Liminf}_{\alpha\to\beta_0}\Lambda(e_0, T, \alpha) = \omega$ and so divergence.

It may already be apparent that the claim that there is an index number e_0 for the above generalized ittm-recursion can be established readily from the eJ-Recursion Theorem. One may argue as follows, somewhat schematically.

Let $F(0, e)$ code the actions of the main programme at (1) above, searching through increasing M-structures for a stable T'. (The e is just a dummy parameter at this stage.) With T' stabilized, the programme asks the oracle query $?\mathrm{eJ}(i, x) = 1/0?$ about $x = \langle 1, \langle B_n\rangle_n, T, T', M\rangle$ with $i = F(1, e)$ to be defined next.

Let $F(k+1, e)$ be the function that returns the index code of the following blocks of computations:

(1) The actions to do to fulfill the query Q^{k+1}, as an explicit computation. s indicated above these can be listed effectively and the code of their formal instructions can be given as a function of k - $q(k+1)$ say. This includes the actions to compute the increasing structures and what to do if stability of any tree passed down subsequently fails. lso included are, if a stability point is reached that requires a new query to a lower subcomputation, the actions to collect together the current trees, to form part of a new coding real x.

(2) "$(x)_0 := (x)_0 + 1$" [% Increase the initial index of x by 1 - here to $k+2$.]

(3) The code of the query instruction: "$?\,\mathrm{eJ}\big(\varphi_e^{\mathrm{eJ}}((x)_0), x\big) = 0/1?$".

Let the two instructions (2) and (3) have code together $t(e) \in \mathbb{N}$.

(4) The code of the post-query actions, on receipt of an answer (in the form of what to do if information is received of a certain kind of tree from a lower subcomputation becoming unstable etc). gain these are effective in k. Let these be $p(k+1)$ say.

We thus may loosely represent the total function $F(k+1, e)$ as:

$$F(k+1, e) = q(k+1)^\frown t(e)^\frown p(k+1).$$

By the eJ-Recursion Theorem there is an index e_0, so that

$$\varphi_{e_0}^{\mathrm{eJ}}(k+1) = F(k+1, e_0) = q(k+1)^\frown t(e_0)^\frown p(k+1).$$

Then our overall computation is: $\{e_0\}^{\mathrm{eJ}}(\langle B_n |\, n < \omega\rangle, T)$.

s for the outcome we have as a final claim:

Claim: *For $A = \bigcup_n B_n \in \Sigma^0_3$ and T a recursive subtree of ${}^{<\omega}\omega$ as above, the programme $P^{\mathrm{eJ}}_{e_0}(\langle B_n | n < \omega\rangle, T)$ will either halt with a code for a strategy for I, if such exists, or else will diverge. In the latter case if it diverges after β steps, then a strategy for II is definable over L_β.*

Proof: We first observe that the master programme (at $\Lambda = 0$) cannot enter an eventual loop: suppose (ζ, Σ) was its first looping pair of ordinals. Then the level of computation at times ζ and Σ is the same: $\Lambda(\zeta) = \Lambda(\Sigma) = 0$. But the argument of Claim 1 of Lemma 4.2, shows that we must have stability of T' by any extendible ordinal ζ, and hence, by the specification of e_0, must be at a level >0 at time ζ: $\Lambda(\zeta) > 0$. The same argument shows that even with $\Lambda(\zeta) = \mathrm{liminf}_{\alpha\to\zeta}\Lambda(\alpha) = 0$, we should have T' diminishing unboundedly below the 2-extendible ζ - which cannot happen.

So the computation either halts or diverges. However divergence can only happen if there is an infinitely descending chain of query calls Q^k. nd such has been designed only to happen when we have complete stability of all our definable trees necessary for the proof of the existence of a definable winning strategy for *II* over L_β - as our procedures mimic. Lastly the main programme can only halt if it produces a winning strategy for I. Q.E.D. Theorem 4.1

Hence by the latter case of the last Claim, strategies for *II* in such games are in general not even semi-recursive in eJ.

Corollary 4.5. *There is a procedure P^{eJ}_e that only diverges at β_0.*

Proof: Let $A = \bigcup_{n<\omega} B_n \in \Sigma^0_3$ be such that $G(A;T)$ is a win for *II*, but there is no winning strategy in L_{α_0}. Then the computation $P^{\mathrm{eJ}}_{e_0}(\langle B_n | \, n < \omega\rangle, T)$ above can only diverge at β_0 since a winning strategy for *II* is definable over L_{β_0} but no earlier. Q.E.D.

• n example of such a game, of the type above, is where *II* must construct an ω-model of "KP + $\mathrm{Det}(\Sigma^0_3)$", and *I* as usual must find a descending chain of ordinals in *II*'s model. Then *II* has an obvious winning strategy, but there cannot be one where *II* produces a model with wellfounded part an ordinal smaller than β_0. We saw in the proof of the theorem above that the computation in a game of this type, continually constructs codes for the levels of the L-hierarchy unboundedly in β_0, - and hence is ultimately divergent. We thus have:

Corollary 4.6. *There is a program code f so that (i) $P^{\mathrm{eJ}}_f(x)$ computes codes for levels for the $L[x]$-hierarchy; (ii) $P^{\mathrm{eJ}}_f(0)$ is divergent , but is not divergent at any stage before β_0, whilst computing codes for levels L_α for α unbounded in β_0.* Q.E.D.

Corollary 4.7. *$\eta_0 = \tau_0$ - that is Theorem 2.11 holds.*

Proof: We have that $\alpha_0 = \eta_0$. By modifying the program of the last Corollary we can find programs $P^{\mathrm{eJ}}_f(0)$ which halt cofinally in the admissible set L_{α_0}, and hence with ranks of such computations unbounded in α_0. Hence $\tau_0 \geq \alpha_0$. By the Boundedness Lemma 3.15 $\tau_0 \leq \alpha_0$. Q.E.D.

Lemma 4.8. *Let $a \subseteq \omega$ be in L_{α_0}. Then a is* eJ*-recursive.* Q.E.D.

The following answers a question of Lubarsky:

Corollary 4.9. *The reals appearing on the tapes of freezing-ittm-computations of* [12] *are precisely those of L_{β_0}; similarly the supremum of the ranks of the wellfounded parts of divergent computation trees is β_0.*

Proof: Freezing-ittms computations are, in the terms here, divergent iJ-computations. s eJ is recursive in iJ we shall have that the iJ-recursive reals and the eJ-recursive reals coincide. These will be the reals of L_{α_0}. By the Boundedness Lemma all such computations are divergent by β_0, whilst at the same time codes for levels of L for $\alpha < \beta_0$ appear on some P_e^{eJ}'s tape. Hence the reals appearing on the divergent iJ-computations are those of L_{β_0}. Q.E.D.

Corollary 4.10. *The complete semi-decidable-in-*eJ *set of integers*

$$K = \{(e, m) \in \omega \times \omega | \, \mathrm{eJ}(e, m) = 1\}$$

is recursively isomorphic to a complete $\Im\Sigma_3^0$ *set.*

Proof: If $P_e^{\mathrm{eJ}}(m)$ is convergent it must be so before β_0: its convergence is a Σ_1-fact true in L_{β_0}. By Σ_1-reflection, it is true in L_{α_0}. Hence the Σ_1-fact of its convergence is mentioned in the Σ_1-$\mathrm{Th}(L_{\alpha_0})$. That is $K \leq_1 \Sigma_1\text{-}\mathrm{Th}(L_{\alpha_0}) \equiv_1 S$ where S is a complete $\Im\Sigma_3^0$ set. The latter holds by Theorem 2.17. For the converse, we have that $n \in S$ if there is a certain strategy in L_{α_0} for a certain game which is winning for I. Such can be found by inspecting the various L_α for $\alpha < \alpha_0$. nd Corollary 4.6 enables us to run a computation which is convergent if such can be found. Hence $S \leq_1 K$. Q.E.D.

Proof of Theorem 2.9
The last Corollary proves the (a) (i) iff (iii) direction of the Theorem, and we have already established (a)(ii) iff (iii) (in the proof of Theorem 2.17). This leaves (b). But this follows from the usual characterisation of the semi-recursive and co-semi-recursive sets as being recursive, the admissibility of L_{α_0}, and that $\alpha_0 = \eta_0$.

Q.E.D. Theorem 2.9

We may also recast the above arguments as showing:

Corollary 4.11. *Both the theory* $T_{\alpha_0}^1$ *and* K *are* $\Im\Sigma_3^0$*-inductive sets of integers.*

Remark 4.12. The same considerations show that in fact the whole of $\mathrm{dom}(\mathrm{eJ}) \cap \omega \times \omega^{<\omega}$ is $\Im\Sigma_3^0$-inductive.

The proofs of Theorems 2.7, 2.9, and 2.11 are now complete (and they cover the statements of the Theorems 1.5-1.8 in Section 1 of the Introduction).

Bibliography

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

[2] . Blass. Complexity of winning strategies. *Discrete Mathematics*, 3:295–300, 1972.

[3] M. Davis. Infinite games of perfect information. *Annals of Mathematical Studies*, 52:85–101, 1964.

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

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

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

[7] L. Harrington and . Kechris. On characterizing Spector classes. *Journal of Symbolic Logic*, 40(1):19–24, March 1975.

[8] P. Hinman. *Recursion-Theoretic Hierarchies*. Ω Series in Mathematical Logic. Springer, Berlin, 1978.

[9] S. C. Kleene. Recursive quantifiers and functionals of finite type I. *Transactions of the American Mathematical Society*, 91:1–52, 1959.

[10] S. C. Kleene. Turing-machine computable functionals of finite type I. In *Proceedings 1960 Conference on Logic, Methodology and Philosopy of Science*, pages 38–45. Stanford University Press, 1962.

[11] S. C. Kleene. Turing-machine computable functionals of finite type II. *Proceedings of the London Mathematical Society*, 12:245–258, 1962.

[12] R. Lubarsky. Well founded iterations of infinite time turing machines. In R-D Schindler, editor, *Ways of Proof Theory*. Ontos, 2010.

[13] D. . Martin. Π_2^1-monotone inductive definitions. In D. . Martin .S. Kechris and Y.N. Moschovakis, editors, *Cabal Seminar 77-79*, volume 839 of *Lecture Notes in Mathematics*, pages 215–234. Springer, Berlin, New York, 1980.

[14] Y.N. Moschovakis. The game quantifier. *Proceedings of the American Mathematical Society*, 31:245–250, 1971.

[15] Y.N. Moschovakis. *Descriptive Set theory*. Studies in Logic series. North-Holland, msterdam, 1980.

[16] S. Simpson. *Subsystems of second order arithmetic*. Perspectives in Mathematical Logic. Springer, January 1999.

[17] L. Svenonius. On the denumerable models of theories with extra predicates. In *The Theory of Models*, pages 376–389. North-Holland Publishing Co., msterdam, 1965.

[18] P.D. Welch. Post's and other problems in higher type supertasks. In B. L we, B. Piwinger, and T. R sch, editors, *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.

[19] P.D. Welch. Characteristics of discrete transfinite Turing machine models: halting times, stabilization times, and normal form theorems. *Theoretical Computer Science*, 410:426–442, January 2009.

[20] P.D. Welch. Weak systems of analysis, determinacy and arithmetical quasi-inductive definitions. *Journal of Symbolic Logic*, September 2011.

[21] P.D. Welch. $G_{\delta\sigma}$-games. Preprint Series NI-12050, Isaac Newton Institute, Cambridge, July 2012.

[22] P.D. Welch. Some observations on truth hierarchies. *Review of Symbolic Logic*, 7(1):1–30, March 2014.