Notre Dame Journal of Formal Logic
Volume 63, Number 2, 2022

Decision Times of Infinite Computations

Merlin Carl, Philipp Schlicht, and Philip Welch

Abstract The *decision time* of an infinite time algorithm is the supremum of its halting times over all real inputs. The *decision time* of a set of reals is the least decision time of an algorithm that decides the set; semidecision times of semidecidable sets are defined similarly. It is not hard to see that ω_1 is the maximal decision time of sets of reals. Our main results determine the supremum of countable decision times as σ and that of countable semidecision times as τ, where σ and τ denote the suprema of Σ_1- and Σ_2-definable ordinals, respectively, over L_{ω_1}. We further compute analogous suprema for singletons.

1 Introduction

Infinite time Turing machines (ittm's) were invented by Hamkins and Kidder as a natural machine model allowing a standard Turing machine to operate not only through unboundedly many finite stages, but transfinitely, thus passing through an ωth stage and beyond. They link computability, descriptive set theory, and low levels of the constructible hierarchy.

While the analogy pursued by Hamkins and Lewis [9] was that of Turing reducibility and its degree theory, the notion of recursion most closely analogous to it in the literature turned out to be that of Kleene recursion (see, e.g., Hinman [7] for an account of this). In this theory, Π^1_1-sets of integers were characterized by transfinite processes bounded by ω_1^{ck} in time, and could be viewed as resulting from computation calls (either viewed as arising from systems of equations, or from Turing machines) along well-founded computable trees.

This theory can be reformulated by modeling transfinite processes by ittm's. Hamkins and Lewis [9] showed that the subsets, either of the natural numbers or Baire space, which are computable by the basic ittm's, fall strictly between the Π^1_1-

Received November 12, 2020; accepted February 22, 2022
2020 Mathematics Subject Classification: Primary 03D60; Secondary 03E15
Keywords: infinite time Turing machines, decision algorithms, halting times, stable ordinals
 10.1215/00294527-2022-0012

and Δ^1_2-sets. They thus provide natural classes in this region and thereby form a natural test case for properties of classes in the Wadge hierarchy.

The computational properties of infinite time Turing machines lead to some new phenomena that do not occur in Turing machines. For instance, there are two natural notions of forming an output—besides a program halting in a final state with an element of Cantor space on its output tape, one can consider the "eventual output," if it occurs, when the output tape is seen to stabilize, even though the machine has not formally halted but is perhaps just working away on its scratch tape. In terms of the machine architecture, one could argue that this eventual output is characteristic: to analyze the halting times of ittm's, one has in any case to analyze these stabilization times, and halting is a special case of stabilization.

Another new phenomenon is the appearance of new reals on the output or work tapes beyond all stabilization times in processes which do not stabilize at all. What are these reals? Hamkins and Lewis dubbed such reals *accidental*. They are constructed in some process, written to the output tape, but are evanescent: later they may be overwritten and disappear.

We make this clearer by giving a brief sketch of how these machines work (for more details, see [9].) An *ittm-program* is just a regular Turing program. The "hardware" of an ittm consists of an input, work, and output tape, each a sequence of cells of order type ω, and a single head that may move one cell to the left or right along the tapes, all of which thus have an initial leftmost cell and are infinite in the rightward direction. The read/write head can read, say, the nth cell from each of the three tapes simultaneously, if the head is situated in position n. (The three tapes are convenient for defining stabilization, but a single tape model has the same computational strength.) Each cell contains 0 or 1. At time α, the machine proceeds to $\alpha + 1$ by following the ordinary Turing program, and acts depending on what it sees on the tape at its current position and the program dictates, just as an ordinary Turing machine of such a kind would. However, at any limit stage μ of time, by fiat, the content of every cell on the tape is set to the inferior limit or *liminf* of the earlier contents at times $\alpha < \mu$ of this cell. Thus a "1" is in a cell at stage μ if and only if $\exists \beta < \mu \ \forall \alpha \in (\beta, \mu)$ that cell has a "1." The head is positioned at the liminf of its positions at times $\alpha < \mu$, and the current state or instruction number to the liminf again of those prior to μ.

Note that in [9], an inessentially different but equivalent set of rules is obtained: limsups rather than liminfs were used for cell values, the read/write head was returned to position zero, and a special limit state was entered. This set of rules can be seen to provide the same class of computable functions. Other variations are possible. It is the liminf (or limsup) rule on the cell values that is the determining feature. Its Σ_2-nature is complete in the sense that all other possible choices of Σ_2-definable rules are reducible to it by Welch [19, Theorem 2.9].

These machines define the following classes of sets. We loosely call any element of Baire space, Cantor space, or $\mathcal{P}(\omega)$ a *real*. Let χ_A denote the characteristic function of a subset A of $\mathcal{P}(\omega)$. Such a set A is called *ittm-decidable* if and only if there is an ittm-program p such that $p(x)$ halts with output $\chi_A(x)$ for any subset x of ω. We shall often omit the prefix *ittm*. A is called *semidecidable* if and only if there is an ittm-program p such that $p(x)$ halts precisely if $x \in A$. We say that A is *cosemidecidable* if its complement is semidecidable. For singletons $A = \{x\}$, we call x *recognizable* if A is decidable. Further, x will be called *semirecognizable* if $\{x\}$ is semidecidable, and cosemirecognizable if $\{x\}$ is cosemidecidable. This terminology is analogous to automata theory and recognizable languages. From a logician's perspective, one might call a real x implicitly definable if the singleton $A = \{x\}$ is definable. In the "lost melody theorem" of [9], the divergence between implicitly and explicitly definable reals is studied. This phenomenon does not appear in Turing machines. But it is not a new phenomenon: for example, the real $0^{(\omega)}$ coding all arithmetical truths is implicitly definable by a Π^0_2-definition, while obviously it cannot be explicitly defined at any finite level of the arithmetical hierarchy.

By a result of Welch [20, Theorem 1.1], the supremum of ittm-halting times on empty input equals the supremum λ of writable ordinals, that is, those ordinals for which an ittm can compute a code on zero input. This solved a well-known problem posed by Hamkins and Lewis. We consider an extension of Hamkins and Lewis's problem by allowing all real inputs. More precisely, we consider the problem of which decision times of sets and singletons are possible. This is defined as follows.

Definition 1.1

(a) The *decision time* of a program p is the supremum of its halting times for arbitrary inputs.
(b) The *decision time* of a decidable set A is the least decision time of a program that decides A.
(c) The *semidecision time* of a semidecidable set A is the least decision time of a program that semidecides A.
(d) The *cosemidecision time* of a cosemidecidable set A is the semidecision time of its complement.

Since halting times are always countable, it is clear that these ordinals are always at most ω_1. It is also not hard to see that the bound ω_1 is attained (see Lemma 3.1 below). The questions are then: Which countable ordinals can occur as decision or semidecision times of sets of real numbers? Which ordinals occur for singletons? As there are only countable many programs, there are only countably many semidecidable sets of real numbers, so the suprema of the countable (semi-)decision times must be countable. Writing ω_1 for ω_1^V throughout, we shall show that these suprema can be determined in the course of this paper. (We call a supremum *strict* to emphasize that it is not attained.)

Definition 1.2

(a) σ denotes the first Σ_1-stable ordinal, that is, the least α with $L_\alpha \prec_{\Sigma_1} L_{\omega_1}$. Equivalently, this is the supremum of the $\Sigma_1^{L_{\omega_1}}$-definable ordinals (see Lemma 2.5).[1]
(b) τ denotes the supremum of the $\Sigma_2^{L_{\omega_1}}$-definable ordinals.

While the value of σ is absolute, we would like to remark that τ is sensitive to the underlying model of set theory. For instance, $\tau < \omega_1^L$ holds in L, but $\tau > \omega_n^L$ if ω_n^L is countable in V. Note that τ equals the ordinal γ_2^1 studied in Kechris et al. [12] by a result of Carl et al. [4].

The following are our main results.

Theorem (see Theorem 3.5) *The strict supremum of countable decision times for sets of reals equals σ.*

Theorem (see Theorem 4.5) *The strict supremum of countable semidecision times for sets of reals equals τ.*

Theorem (see Theorems 3.3, 4.6, and 4.10) *The strict suprema of decision times, semidecision times, and cosemidecision times for singletons equal σ.*

We also prove the existence of semidecidable and cosemidecidable singletons that are not recognizable. The latter answers Question 4.5.5 of Carl [2].

Since there are gaps in the clockable ordinals (see [9]), it is natural to ask whether there are gaps below σ in the countable decision times. We answer this in Theorem 3.7 by showing that gaps of arbitrarily large lengths less than σ exist.

2 Preliminaries

We fix some notation and recall some facts.

The symbols $x, y, z, \ldots$ will be reserved for reals and elements of Cantor space ${}^{\omega}2$. WO will denote the set of reals that code strict total orders of ω which are well-ordered. This class is a complete Π^1_1-set of reals. It is a basic result of [9, Corollary 2.3] that it is ittm-decidable.

As usual in admissible set theory (see Barwise [1]), we write $\omega_1^x = \omega_1^{x,\mathrm{ck}}$ for the least ordinal not recursive in x. It is thus the least ordinal α so that $L_\alpha[x]$ is an admissible set. We use the well-known fact that if M is a transitive model which is a union of admissible sets that are elements of M, then Σ^1_1-relations are absolute to M. Thus if $\varphi(y)$ is a Σ^1_1-statement about $y \in M$, then $\varphi(y) \Leftrightarrow \varphi(y)^M$.

When α is an ordinal, let $\alpha^\oplus$ denote the least admissible ordinal that is strictly above α. When α is an ordinal which is countable in L, then x_α denotes the $<_L$-least real coding α. Conversely, when a real x, regarded as a set of integers, codes an ordinal via a recursive pairing function on ω, then we denote this ordinal by α_x.

We write $p(x)\downarrow y$ if a program p with input x halts with y on its output tape, and $p(x)\downarrow^{\leq\alpha} y$ if it so halts at or before time α. Moreover, y can be omitted if one does not want to specify the output.

Definition 2.1 Each of the following is defined as the supremum of ordinals coded by reals of the following form:

(a) λ for halting outputs of some $p(0)$.
(b) ζ for stable outputs of some $p(0)$.
(c) Σ for reals which appear on the tape of some computation $p(0)$.

These three classes are then the *writable*, the *eventually writable*, and the *accidentally writable* ordinals, respectively.

The relativizations for reals x are denoted λ^x, ζ^x, and Σ^x. In particular, ω_1^x is smaller than λ^x. From this one can show the following fact, which we shall use without further mention in the sequel. If $p(x)\downarrow y$, then we say that y is *ittm-computable* from x and write $y \leq_\infty x$. We further write $x =_\infty y$ if $x \leq_\infty y$ and $y \leq_\infty x$.

Lemma 2.2 $y \leq_\infty x \Leftrightarrow y \in L_{\lambda^x}[x] \Leftrightarrow L_{\lambda^y}[y] \subseteq L_{\lambda^x}[x]$.

We shall use the following characterization of λ, ζ, and Σ.

Theorem 2.3 (The λ-ζ-Σ theorem; cf. [19], Corollary 32 of [22]) *The triple (λ, ζ, Σ) is the lexicographically least triple so that $L_\lambda \prec_{\Sigma_1} L_\zeta \prec_{\Sigma_2} L_\Sigma$.*

The action of any particular program p on input an integer will of course depend on the program itself, but there are programs p for which $p(k)$ does not halt but runs forever. The significance of the ordinals ζ and Σ in this case is that, by stage ζ, the machine will enter a *final loop* from ζ to Σ. In a final loop from α to $\alpha + \beta$, the snapshots of the machine at stage $\alpha + (\beta \cdot \gamma)$ are by definition identical for all ordinals γ. In other words, the loop is repeated endlessly. (A computation may have a loop that is iterated finitely often without it being of this final looping kind.) The point is that one can easily recognize final loops from α to $\alpha + \beta$ as the loops with the following property for each cell: if the inferior limit 1 is attained at $\alpha + \beta$, then the contents of the cell is constant throughout the loop.

Of particular interest is the *Theory Machine* (cf. Friedman and Welch [5]) that also does not stop, but on zero input writes (and overwrites) the Σ_2-theories of the levels of the L_α hierarchy to the output tape for all $\alpha \leq \Sigma$. But the Σ_2-theory of L_Σ is identical to that of L_ζ. Hence there is a final loop from ζ to Σ, but no shorter final loop and none beginning before ζ.

In this "λ-ζ-Σ-theorem" one should note that Σ is a limit of admissible ordinals, but is not itself admissible. As part of the analysis of this theorem, γ, the supremum of *halting times* of computations $p(k)$ on integer input (the *clockable* ordinals) was shown to be λ (see [20]). The ordinals which have a code computed as an output of some $p(k)$ (the *writable* ordinals) are an initial segment of the countable ordinals, but the clockables are not.

Definition 2.4 σ_ν denotes the supremum of $\Sigma_1^{L_{\omega_1}}$-definable ordinals with parameters in $\nu \cup \{\nu\}$.

Lemma 2.5 *$L_{\sigma_\nu} \prec_{\Sigma_1} L_{\omega_1}$ for any countable ordinal ν.*

Proof Assume that $\nu = 0$ for ease of notation. Let $\hat{\sigma}$ denote the least α with $L_\alpha \prec_{\Sigma_1} L_{\omega_1}$. It suffices to show that every element of $L_{\hat{\sigma}}$ is $\Sigma_1^{L_{\omega_1}}$-definable. If not, then the set N of $\Sigma_1^{L_{\omega_1}}$-definable elements of L_{ω_1} is not transitive, so the collapsing map $\pi\colon N \to \bar{N}$ moves some set x. Assume that x has minimal L-rank and x is $\Sigma_1^{L_{\omega_1}}$-definable by $\varphi(x)$. One may check that $N \prec_{\Sigma_1} L_{\omega_1}$: suppose that $(\exists z\, \psi(z, x_0, \ldots, x_n))^{L_{\omega_1}}$ with $x_i \in N$ and ψ a Σ_0-formula. Then the $<_L$-least such z is Σ_1-definable in the vector of the x_i; replacing each x_i by its Σ_1-definition yields a Σ_1-definition of such a z. Hence $(\exists z\, \psi(z, x_0, \ldots, x_n))^N$. Thus $\bar{N} \models \varphi(\pi(x))$. Since $\bar{N}$ is transitive, this implies $V \models \varphi(\pi(x))$. Since $\pi(x) \neq x$, this contradicts the assumption that $\varphi(x)$ has x as its unique solution. □

Note that Σ_1-statements in H_{ω_1} are equivalent to Σ^1_2-statements and conversely (see Jech [10, Lemma 25.25]). In particular, L_σ is Σ^1_2-correct in V. We shall use this without further reference below.

3 Decision Times

In this section, we focus on the problem of ascertaining the decision times of ittm-semidecidable sets.

3.1 The supremum of countable decision times By a standard condensation argument (see [9, Theorem 1.1]), halting times of ittm's on arbitrary inputs are always countable, so it is clear that decision times are always at most ω_1.

Recall that, by [9, Corollary 2.3], all Π^1_1-sets are ittm-decidable.

Lemma 3.1 *Every set with countable decision time is Borel. Hence any non-Borel Π^1_1-set has decision time ω_1.*

Proof Suppose that an ittm-program p semidecides a set A within a countable time α. Note that $p(x)\downarrow^{\leq\alpha}$ can be expressed by Σ^1_1- and Π^1_1-formulas in any code for α. By Lusin's separation theorem, it is Borel. □

For instance, the set WO of well-orders on the natural numbers is Π^1_1-complete and hence not Borel. Its decision time thus equals ω_1. Since Π^1_1-sets are ittm-decidable, WO is ittm-decidable with decision time ω_1 (cf. Carl [3, Proposition 32]).

It remains to study sets with countable decision times and, in particular, the following question.

Question *What is the supremum of countable decision times of sets of reals?*

We need two auxiliary results to answer this problem. The next lemma shows that if x is semirecognizable, but $x \notin L_{\alpha^\oplus}$, then $\{x\}$'s semidecision time is greater than α.

Lemma 3.2

(1) *If p semirecognizes x and $p(x)\downarrow^{\leq\alpha}$, then $x \in L_{\alpha^\oplus}$.*
(2) *The bound of $\alpha^\oplus$ is in general optimal.*

Proof (1) Let $M = L_{\alpha^\oplus}$, and take any $\mathrm{Col}(\omega,\alpha)$-generic filter $g \in V$ over M. Since $\mathrm{Col}(\omega,\alpha)$ is a set forcing in M, $M[g]$ is admissible if g is taken to be sufficiently generic. (By Mathias [13, Theorem 10.17], it suffices that the generic filter meets every dense class that is a union of a Σ_1-definable with a Π_1-definable class.) In this model everything is countable. Let $y \in \mathrm{WO} \cap M[g]$ be a real coding α.

Set $R(z)$ if "$\exists h$[*h codes a sequence of computation snapshots of $p(z)$, along the ordering y, which converges with output* 1]." Then as $x \in R$, the latter is a nonempty $\Sigma^1_1(y)$-predicate; by an effective Σ^1_1-perfect set theorem (see Sacks [15, Chapter III, Theorem 6.2]) (relativized to y), if there is no solution to R in $L_{\omega_1^y}[y]$, then there is a perfect set of such solutions in V. But $R = \{x\}$. Hence $x \in L_{\omega_1^y}[y] = M[g]$. As $\mathrm{Col}(\omega,\alpha)$ is homogeneous (cf. [10, Corollary 26.13]), we can see that $x \in L_{\alpha^\oplus}$ by asking for each $n \in \omega$ whether $\mathrm{Col}(\omega,\alpha)$ forces n to be in some real z such that $p(z)$ halts.

(2) This is essentially Rogers [14, Theorem LII]: to sketch why this is so, take any computable ordinal $\gamma < \omega_1^{\mathrm{ck}}$. One can construct a real x which is Π^0_2 as a singleton and codes a sequence of iterated (ordinary Turing jumps) of length $\omega \cdot \gamma$. Then $x \notin L_\gamma$ (as the theory of L_γ is reducible to x), but x is recognizable in time $\omega + 1$ since x is Π^0_2 (recall, for example, that on zero input a complete $\Pi^0_2(x)$-set can be written to the worktape in ω stages). This shows that $\omega^\oplus = \omega_1^{\mathrm{ck}}$ is optimal when $\alpha = \omega + 1$. □

The next result will be used to provide a lower bound for decision times of sets.

Theorem 3.3 *The supremum of decision times of singletons equals σ.*

Remark 3.4 It should be unsurprising that the supremum of decision times is at least σ. It is well known that the Π^1_1-singletons are well-ordered and appear unboundedly in L_σ by work of Suzuki [18], and these are clearly ittm-decidable. Moreover, their order type is σ (see, e.g., [14, Exercise 16.63]).

Proof To see that the supremum is at least σ, take any $\alpha < \sigma$. Pick some β with $\alpha < \beta < \sigma$ such that some Σ_1-sentence φ holds in L_β for the first time. We claim that the $<_L$-minimal code x for $L_{\beta^\oplus}$ is recognizable. To see this, let T denote the theory $\mathsf{KP} + (V = L) + \varphi$. Devise a program $p(z)$ that checks if z codes a well-founded model of $T+$ *"there is no transitive model of T"* and halts if so. However, such a code x is not an element of $L_{\beta^\oplus}$. By Lemma 3.2, the decision time of $\{x\}$ is thus at least $\beta^\oplus$.

It remains to show that any recognizable real x is recognizable with a uniform time bound strictly below σ. To see this, suppose that p recognizes x. We shall run p and a new program q synchronously, and halt as soon as one of them does. Thus q ensures that the halting time is small.

We now describe q. A run $q(y)$ simulates all ittm-programs with input y synchronously. For each halting output on one of these tapes, we check whether it codes a linear order. In this case, run a well-foundedness test and save the well-founded part, as far as it is detected. (These routines are run synchronously for all tapes, one step at a time.) A well-foundedness test works as follows. We begin by searching for a minimal element; this is done by a subroutine that searches for a strictly decreasing sequence $x_0, x_1, \ldots$. If the sequence cannot be extended at some finite stage, then we have found a minimal element and add it to the well-founded part. The rest of the algorithm is similar and proceeds by successively adding new elements to the well-founded part. Each time the well-founded part increases to some $\alpha + 1$ by adding a new element, we construct a code for $L_{\alpha+1}$. (Note that the construction of L_α takes approximately $\omega^\omega \cdot \alpha$ many steps.[2]) We then search for z such that $p(z)\downarrow^{\leq\alpha} 1$ in $L_{\alpha+1}$. We halt if such a z is found and $x \neq z$.

By Lemma 3.2, $x \in L_{\lambda^x}$. So for any y with $\lambda^y \geq \lambda^x$, some L_α satisfying $p(x)\downarrow^{\leq\alpha} 1$ appears in $q(y)$ in $<\lambda^x$ steps. Otherwise, $\lambda^y < \lambda^x$, so $p(y)$ will halt in $<\lambda^y$, and therefore $<\lambda^x$ steps. Clearly $\lambda^x < \sigma$. □

We call an ittm-program *total* if it halts for every input. We are now ready to prove the main results of this section.

Theorem 3.5 *The suprema of countable decision times of (a) total programs and of (b) decidable sets equal σ.*

Proof Given Theorem 3.3, it remains to show that σ is a strict upper bound for countable decision times of total programs. Suppose that p is total and has a countable decision time. Since $\exists\alpha < \omega_1\ \forall x p(x)\downarrow^{\leq\alpha}$ is a Σ^1_2-statement, this holds in L by Shoenfield absoluteness. Since $L_\sigma \prec_{\Sigma^1_2} L$, there is some $\alpha < \sigma$ such that $\forall x p(x)\downarrow^{\leq\alpha}$ holds in V, as required. □

3.2 Quick recognizing The lost melody theorem, that is, the existence of recognizable, but not writable, reals in [9, Theorem 4.9] shows that the recognizability strength of ittm's goes beyond their writability strength. It thus becomes natural to ask whether this result still works with bounds on the time complexity. If a real x can be written in α many steps, then it takes at most $\alpha + \omega + 1$ many steps to recognize x by simply writing x and comparing it to the input. Can it happen that a writable real can be semirecognized much quicker than it can be written? The next lemma shows that this is impossible.

Lemma 3.6 *Suppose that p recognizes x and $p(x)$ halts at time α. Then:*

(1) *$x \in L_\beta$ for some $\beta < \alpha^\oplus$.*
(2) *x is writable from any real coding β in time less than $\beta^\oplus$ steps. If β is clockable, then x is simply writable in time less than $\beta^\oplus$.*

Proof The first claim holds by Lemma 3.2. For the second claim, note that there is an algorithm that writes a code for β in at most β many steps by the quick writing theorem (see Welch [22, Lemma 48]). One can therefore write codes for L_β and any element of L_β in less than $\beta^\oplus$ many steps. □

3.3 Gaps in the decision times It is well known that there are gaps in the set of halting times of ittm's (see [9, Section 3]). We now show that the same is true for semidecision times of total programs and thus of sets.

A *gap* in the semidecision times of programs is an interval that itself contains no such times, but is bounded by one.

Theorem 3.7 *For any $\alpha < \sigma$, there is a gap below σ of length at least α in the semidecision times of programs.*

Proof Consider the Σ^1_2-statement *"there is an interval $[\beta, \gamma)$ strictly below ω_1 of length α such that for all programs p, there is (i) a real y such that $p(y)$ halts later than γ, or (ii) for all reals y such that $p(y)$ halts, it does not halt within $[\beta, \gamma)$."* This statement holds, since its negation implies that any interval $[\beta, \gamma)$ strictly below ω_1 of length α contains the decision time of a program. Since $L_\sigma \prec_{\Sigma_1} L$ by Lemma 2.5, such an interval exists below σ. □

Note that we similarly obtain gaps below τ of any length $\alpha < \tau$ by replacing σ by σ_α.

4 Semidecision Times

In this section, we shall determine the supremum of the countable semidecision times. We then study semidecision times of singletons and their complements and show that undecidable singletons of this form exists.

4.1 The supremum of countable semidecision times We shall need the following auxiliary result.

Lemma 4.1 *The supremum of $\Pi_1^{L_{\omega_1}}$-definable ordinals equals τ.*

Proof Suppose that $\bar{\alpha} < \tau$ is $\Sigma_2^{L_{\omega_1}}$-definable as a singleton by the formula $\psi(\bar{\alpha}) = \exists\beta\, \forall w \varphi(\bar{\alpha}, \beta, w)$, where φ is Δ_0. Let $\Psi(\alpha, \beta)$ abbreviate

$$\text{“}(\alpha, \beta) \text{ is } <_{lex}\text{-least such that } \forall w \varphi(\alpha, \beta, w).\text{”}$$

Then we shall have $L_{\omega_1} \models \Psi(\bar{\alpha}, \bar{\beta})$ for some $\bar{\beta}$. However, then for all sufficiently large $\delta < \omega_1$, we have likewise $L_\delta \models \Psi(\bar{\alpha}, \bar{\beta})$. To see this, take any δ such that for each $\alpha < \bar{\alpha}$ there is some $\beta < \delta$ with $\neg\varphi(\alpha, \beta, w)$. Now let $\bar{\delta}$ be least with $L_{\bar{\delta}} \models \Psi(\bar{\alpha}, \bar{\beta})$. Note that $\tau > \bar{\delta} > \max\{\bar{\alpha}, \bar{\beta}\}$. Then we have a $\Pi_1^{L_{\omega_1}}$-definition of $\bar{\delta}$ as a singleton:

$$\delta = \bar{\delta} \iff \exists \alpha, \beta < \delta \big[\forall w \varphi(\alpha, \beta, w) \wedge L_\delta \models \text{“}\Psi(\alpha, \beta) \wedge\ \forall \eta \big(\neg\Psi(\alpha, \beta)^{L_\eta}\big)\text{”}\big].$$

There is a bounded existential quantifier in front of the conjunction of two Π_1-formulas in δ. This is a Π_1-definition in δ over models of KP. The first conjunct guarantees that the witnessing α equals $\bar{\alpha}$; the second conjunct that $\beta = \bar{\beta}$. Now $\bar{\alpha} < \bar{\delta} < \tau$ as required. □

Corollary 4.2 *If $V = L$, then the Π^1_2-singleton reals appear unboundedly below τ, and $\tau = \delta^1_3$, the supremum of Δ^1_3 well-orders of ω.*

Remark 4.3 The previous corollary is the natural analogue at one level higher of the facts that the Π^1_1-singletons appear unboundedly in σ, and the latter equals the analogously defined δ^1_2. At this lower level, the relevant objects are absolute via Levy–Shoenfield absoluteness, and the assumption $V = L$ is not needed.

We shall use the effective boundedness theorem.

Lemma 4.4 (Essentially Spector [16]) *The rank of any well-founded $\Sigma^1_1(x)$-relation is strictly below $\omega_1^{\mathrm{ck},x}$. In particular, any $\Sigma^1_1(y)$-subset A of* WO *is bounded by $\omega_1^{\mathrm{ck},y}$.*

We quickly sketch the proof for the reader. The proof of the Kunen–Martin theorem in Kechris [11, Theorem 31.1] shows that the rank of R is bounded by that of a computable well-founded relation S on ω. Since $L_{\omega_1^{\mathrm{ck},x}}[x]$ is x-admissible, the calculation of the rank of S takes place in $L_{\omega_1^{\mathrm{ck},x}}$, and hence the rank is strictly less than $\omega_1^{\mathrm{ck},x}$.

While the second claim (which is essentially due to Spector) follows immediately from the first one, we give an alternative proof without use of the Kunen–Martin theorem. Fix a computable enumeration $\vec{p} = \langle p_n \mid n \in \omega\rangle$ of all Turing programs. Let N denote the set of $n \in \mathbb{N}$ such that p_n^y is total and the set decided by p_n codes an ordinal. By standard facts in effective descriptive set theory (e.g., the Spector–Gandy theorem; see [17] and [6]; see also Hjorth [8, Theorem 5.3]), N is $\Pi^1_1(y)$-complete. In particular, it is not $\Sigma^1_1(y)$. Toward a contradiction, suppose that A is unbounded below $\omega_1^{\mathrm{ck},y}$. Then $n \in N$ if and only if there exist a decided by p_n, a linear order b coded by a and some $c \in A$ such that b embeds into c. Then N is $\Sigma^1_1(y)$.

Theorem 4.5 *The supremum of countable semidecision times equals τ.*

Proof To see that τ is a strict upper bound, note that the statement *"there is a countable upper bound for the decision time"* is $\Sigma_2^{L_{\omega_1}}$. In particular, if $(\exists x\ \forall y\, \psi(x, y))^{L_{\omega_1}}$, then $(\exists x \in L_\tau\ \forall y\, \psi(x, y))^{L_{\omega_1}}$.

It remains to show that the set of semidecision times is unbounded below τ. In the following proof, we call an ordinal β an *α-index* if $\beta > \alpha$ and some $\Sigma_1^{L_{\omega_1}}$-fact with parameters in $\alpha \cup \{\alpha\}$ first becomes true in L_β. Thus σ_α is the supremum of α-indices. Any such σ_α, like σ, is an admissible limit of admissible ordinals.

Suppose that ν is $\Pi_1^{L_{\omega_1}}$-definable. (There are unboundedly many such ν below τ by Lemma 4.1.) Fix a Π_1-formula $\varphi(u)$ defining ν. We shall define a Π^1_1-subset $A = A_\nu$ of WO. A will be bounded, since for all $x \in A$, α_x will be a $\bar{\nu}$-index for some $\bar{\nu} \leq \nu$, and hence $\alpha_x < \sigma_\nu$.

For each $x \in$ WO, let ν_x denote the least ordinal $\bar{\nu} < \alpha_x$ with $L_{\alpha_x} \models \varphi(\bar{\nu})$, if this exists. Let $\psi(u)$ state that *"ν_u exists and α_u is a ν_u-index."* Let A denote the set of $x \in$ WO which satisfy $\psi(x)$. Clearly A is Π^1_1.

Claim *The decision time of A_ν equals σ_ν. Furthermore, for any ittm that semidecides A_ν, the order type of the set of halting times for real inputs is at least σ_ν.*

Proof The definition of A_ν yields an algorithm to semidecide A_ν in time σ_ν. Now suppose that for some $\gamma < \sigma_\nu$, there is an ittm-program p that semidecides A with decision time γ. Let g be $\mathrm{Col}(\omega,\gamma)$-generic over L_{σ_ν} in that it meets all dense sets of this partial order that are elements of L_{σ_ν}. Let $x_g \in L_{\sigma_\nu}[g]$ be a real coding g. Such genericity preserves the admissibility of ordinals in the interval (γ, σ_ν), in that, for such ordinals, τ, $L_\tau[g]$, and a fortiori $L_\tau[x_g]$, are admissible sets. As we have observed, σ_ν is a limit of admissibles, and thus $\gamma < \omega_1^{\mathrm{ck},x_g} < \sigma_\nu$. However, A is $\Sigma^1_1(x_g)$, since $x \in A$ holds if and only if there is a halting computation $p(x)$ of length at most γ. By Lemma 4.4, A is bounded by $\omega_1^{\mathrm{ck},x_g}$. This contradicts the definition of A, as it is unbounded in σ_ν.

For the second sentence of the claim, construct a strictly increasing sequence of halting times of length σ_ν by Σ_1-recursion over L_{σ_ν}. It is unbounded in σ_ν by the first claim, and hence its length is σ_ν. □

This proves Theorem 4.5. □

4.2 Semirecognizable reals We have essentially completed the calculation of the supremum of semidecision times of singletons. The upper bound follows from Lemma 2.5 and the lower bound from Lemma 3.2.

Theorem 4.6 *The supremum of semidecision times of singletons equals σ.*

To see that this does not follow from Lemma 3.3, note that semirecognizable, but not recognizable, reals exist by [2, Theorem 4.5.4]. In fact, we shall obtain a stronger result via the next lemma.

Lemma 4.7

(1) *If x is semirecognizable and $y =_\infty x$, then y is semirecognizable.*
(2) *If x is a* fast *real, that is, $x \in L_{\lambda^x}$, then every $y =_\infty x$ is similarly fast.*

Proof (1) Suppose that x is semirecognizable via the program p. Let $q(x) \downarrow y$ and $r(y) \downarrow x$. The following program semirecognizes y. On input $\bar{y}$, compute $r(\bar{y})$ and if this halts with output $\bar{x}$, then compute $p(\bar{x})$. If the latter halts (and so $\bar{x} = x$), then we perform $q(\bar{x}) \downarrow y$ and check that $y = \bar{y}$. If so, then we halt with output 1 and diverge otherwise.

(2) Note that $y =_\infty x$ implies $\lambda^y = \lambda^x$. So $y \in L_{\lambda^x}[x] = L_{\lambda^x}$. □

Theorem 4.8 *Let x be any real.*

(1) *No real in $L_{\Sigma^x}[x] \setminus L_{\lambda^x}[x]$ is recognizable.*
(2) *No real in $L_{\zeta^x}[x] \setminus L_{\lambda^x}[x]$ is semirecognizable.*
(3) *Suppose that y is both fast*[3] *and semirecognizable, $x \leq_\infty y$, and $\lambda^x = \lambda^y$. Then x is semirecognizable.*
(4) *All reals in $L_\Sigma \setminus L_\zeta$ are semirecognizable.*

Proof (1) Suppose that p recognizes $y \in L_{\Sigma^x}[x]$. We run a universal ittm q with oracle x and run $p(z)$ on each tape contents z produced by q. Once p is successful, we have found y and shall write it on the output tape and halt. Hence $y \in L_{\lambda^x}[x]$.

(2) Suppose that p eventually writes y from x and q semirecognizes y. We run $p(x)$ and in parallel $q(z)$, where z is the current content of the output tape of p. Whenever the latter z changes, the run of $q(z)$ is restarted. When $q(z)$ halts, output

z and halt. To see that this algorithm writes y, note that the output of $p(x)$ eventually stabilizes at y, so $q(y)$ is run and y is output when this halts. Hence $y \in L_{\lambda^x}[x]$.

(3) Since y is fast, we have $y \in L_{\lambda^y}$; as $\lambda^y = \lambda^x$, it follows that $y \in L_{\lambda^x}$. Then $y \leq_\infty x$ by Lemma 2.2, and since $x \leq_\infty y$, we get $x =_\infty y$. By part (1) of Lemma 4.7, x is semirecognizable.

(4) Take any $x \in L_\Sigma \setminus L_\zeta$.

We first show that $\lambda^x > \zeta$.[4] Assume that $\lambda^x \leq \zeta$. Since $\Sigma \leq \Sigma^x$, we have $L_{\Sigma^x}[x] \models x \in L$. By Σ_1-reflection $L_{\lambda^x}[x] \models x \in L$, and hence $x \in L_\zeta$. But this contradicts the choice of x.

We now show that $\lambda^x > \Sigma$. Since $L_{\lambda^x}[x] \prec_{\Sigma_1} L_{\Sigma^x}[x]$ and L_ζ is the maximal proper Σ_1-substructure of L_Σ, we must have $\Sigma < \Sigma^x$. The existence of a Σ_2-extendible pair reflects to L_{λ^x}, and hence $\Sigma < \lambda^x$.

Let y denote the $<_L$-least code of L_Σ. Clearly y is recognizable via first-order properties of L_Σ. We claim that y is fast. Since $y \in L_{\lambda^x}[x]$ and $x \in L_{\lambda^y}[y]$, we have $y =_\infty x$ and $\lambda^x = \lambda^y$ by Lemma 2.2. Thus $\lambda^y > \Sigma$ by the previous argument and $y \in L_{\Sigma+1} \subseteq L_{\lambda^y}$ as required. The result then follows from (3). □

We remark that some requirement on λ^x is needed in (3) of Theorem 4.8. To see this, take any Cohen-generic $x \in L_{\lambda^y}$ over L_Σ, where y denotes the $<_L$-least code for Σ. (In this case, we have $\lambda^x = \lambda < \lambda^y$, so that the condition $\lambda^x = \lambda^y$ is violated.) Since $y \in L_{\Sigma+1} \subseteq L_{\lambda^y}$, y is fast. Moreover, since $x \in L_{\lambda^y}$, we have $x \leq_\infty y$. Finally, y is recognizable (and hence, a fortiori, semirecognizable) by testing whether it is the $<_L$-minimal code of the minimal L-level that has a proper Σ_2-elementary submodel. Thus, the other assumptions of Theorem 4.8 are satisfied. We claim that x is not semirecognizable by a program p. Otherwise, $L_\Sigma[x] \models p(x)\downarrow^\alpha$ for some $\alpha < \lambda^x$. This statement is forced over L_Σ for the Cohen real, and we can take two incompatible Cohen reals over L_Σ for which p would have to halt.

4.3 Cosemirecognizable reals Here we study semidecision times for the complements of cosemirecognizable reals. We shall call them *cosemidecision times*.

We determine the supremum of cosemidecision times of singletons. To this end, we shall need an analogue to Lemma 3.2. It will be used to show that any countable cosemidecision time of a program p is strictly below σ.

Lemma 4.9 *If p cosemirecognizes x and $p(x)$ has a final loop of length $\leq \alpha$, then*

(1) $\alpha < \sigma$;
(2) $x \in L_{\alpha^\oplus}$.

Proof (1) Suppose that p cosemirecognizes x. Then x is the unique y such that $p(y)$ loops. The statement that $p(y)$ loops for some y is a true Σ_1-statement, and it therefore holds in L_σ. By uniqueness, $x \in L_\sigma$, and further the length of that final loop is some $\alpha < \sigma$.

(2) Let $M = L_{\alpha^\oplus}$, and take any $\mathrm{Col}(\omega, \alpha)$-generic filter $g \in V$ over M. Since $\mathrm{Col}(\omega, \alpha)$ is a set forcing in M, $M[g]$ is admissible. Let $y \in M[g] \cap \mathrm{WO}$ be a real coding α. As in the proof of Lemma 3.2, it suffices to show that $x \in M[g]$. Following that proof, set $R(z)$ if "$\exists h$[*h codes a sequence of computation snapshots in $p(z)$, along the ordering y, of a computation of length α with a final loop*]." The rest of the argument is identical, as $z = x$ is the only possible solution to $R(z)$. □

Theorem 4.10 *The supremum of cosemidecision times of reals equals σ.*

Proof We first show that σ is an upper bound. Suppose that p cosemirecognizes x. We define a new program r which will cosemirecognize x in less than σ steps. The program r will work on input y by simultaneously running p and the following program q, and halting as soon as $p(y)$ or $q(y)$ halts. The definition of $q(y)$ is based on the machine considered in [5], which writes the $\Sigma_2(y)$-theories of $J_\alpha[y]$ in its output, successively for α. Note that the $\Sigma_2(y)$-theory of $J_\alpha[y]$ appears in step $\omega^2 \cdot (\alpha+1)$. (This is the reason for the choice of this specific program.) $q(y)$ searches within these theories for two writable reals relative to y: a real z and a real coding an ordinal α such that $p(z)$ has a final loop of length at most α (in particular, it does not converge). Note that such a loop occurs by time Σ^z, if it occurs at all (see [20, Main Proposition] or [22, Lemma 2]). If such reals are found, then we check whether $z = y$; if that is the case, then r runs into a loop. Otherwise, r halts.

It is not hard to see that r cosemirecognizes x: If $y = x$, then $p(y)$ will diverge. Moreover, $q(y)$ will either (i) eventually produce x and a theory witnessing the fact that $p(x)$ loops and find that $x = y$ and diverge, or (ii) never produce such a theory and thus diverge while looking for it; in both cases $q(y)$ diverges. On the other hand, if $y \neq x$, then, by the definition of p, $p(y)$ and thus $r(y)$ will halt.

It now suffices to show that the semidecision time of r is at most $(\Sigma^x)^\oplus$, which is smaller than σ. Since $r(x)$ diverges, we can assume that $y \neq x$. We now consider two cases.

First, if $\lambda^y \leq \Sigma^x$, then $p(y)$ halts at time Σ^x or before. Therefore, $r(y)$ will also halt in $< (\Sigma^x)^\oplus$ many steps.

Now suppose that $\lambda^y > \Sigma^x$. Since λ^y is a limit of admissible ordinals, it follows that $\lambda^y > (\Sigma^x)^\oplus$. By Lemma 4.9, $x \in L_{(\Sigma^x)^\oplus}$. By the definition of q, the statement *"there exists z such that the length of $p(z)$'s loop is $\leq \alpha$"* appears in the computation of $q(y)$ in strictly less than $(\Sigma^x)^\oplus$ steps. By the definition of p, we will have $z = x$, and since we are assuming that $y \neq x$, we will have $y \neq z$, so that, by the definition of r, the computation $r(y)$ again halts in $< (\Sigma^x)^\oplus$ many steps.

It remains to show that σ is minimal. Toward a contradiction, suppose that $\beta < \sigma$ is a strict upper bound for the cosemidecision times. We can assume that x_β is recognizable, for instance, by taking β to be an index. Suppose that x_β is recognized by an algorithm with decision time α. Note that $\alpha < \sigma$ by Theorem 3.3. Take $\gamma \geq (\alpha + \beta)^\oplus$ such that x_γ is recognizable. Then $x_\gamma \oplus x_\beta$ is recognizable.

We claim that $x_\beta \oplus x_\gamma$ is not cosemirecognizable by an algorithm with decision time strictly less than β. So suppose that p such an algorithm. We shall describe an algorithm q that semirecognizes x_γ in at most $\alpha + \beta + 1$ steps. This contradicts Lemma 3.2. Note that x_β is coded in x_γ by a natural number n. The algorithm q extracts the real x coded by n from the input y. It then decides whether $x = x_\beta$, taking at most α steps, and diverges if $x \neq x_\beta$. If $x = x_\beta$, then we run $p(x \oplus y)$ for β steps and let q halt if and only if $p(x \oplus y)$ fails to halt before time β. Then $q(y)$ halts if and only if $y = x_\gamma$. □

The previous result would follow from Theorem 3.3 if every cosemirecognizable real were recognizable. However, the next result disproves this and thus answers [2, Question 4.5.5].

Theorem 4.11 *The cosemirecognizable, but not recognizable, reals appear cofinally in L_σ; that is, their L-ranks are cofinal in σ.*

Proof Let ξ be an index, let $x = x_\xi$, and let $y = x_{\lambda^x}$. Since $y \in L_{\zeta^x} \setminus L_{\lambda^x}$, it is not semirecognizable by Lemma 4.8. We claim that y is cosemirecognizable. We shall assume that $\lambda^x = \lambda$; the general case is similar.

For an input z, first test whether it fails to be a code for a well-founded L_α; if it is such a code, then check if it has an initial segment which itself has a Σ_2-substructure (equivalently, $\alpha \geq \Sigma$); if it fails this test, then check if it has a Σ_1-substructure (if it does, then $\alpha \neq \lambda$). If it fails this last point, then z is a code for an L_α with $\alpha \leq \lambda$. Check if z fails to be the $<_L$-least code for L_α. Furthermore, run a universal machine and check whether some program halts beyond α. □

5 Sets with Countable Decision Time

Any set A with countable semidecision time α is Σ^1_1 in any code for α. If this is witnessed by a program p, and y is a code for α, then $x \in A$ if and only if *there exists a halting computation of $p(x)$ along the ordering coded by y*. Thus A is a $\Sigma^1_1(y)$-set. Similarly, any set with a countable decision time is both Σ^1_1 and Π^1_1 in any code for α, and is thus Borel. The next result shows that, for both implications, the converse fails. This complements Lemma 3.1.

Theorem 5.1 *There is a cocountable open decidable set A that is not semidecidable in countable time.*

Proof Let $\vec{\varphi} = \langle \varphi_n \mid n \in \omega \rangle$ be a computable enumeration of all Σ_1-formulas with one free variable. Let B denote the discrete set of all $0^n{}^\frown\langle 1\rangle^\frown x$, where x is the $<_L$-least code for the least L_α where $\varphi_n(x)$ holds. Let further p denote an algorithm that semidecides B as follows. First test if the input equals 0^ω, and halt in this case. Otherwise, test if the input is of the form $0^n{}^\frown\langle 1\rangle^\frown x$, run a well-foundedness test for x, which takes at least α steps for codes for L_α, and then test whether α is least such that $\varphi_n(x)$ holds in L_α. The decision time of p is at least σ. Moreover, it is countable since B is countable.

Let A denote the complement of B. Toward a contradiction, suppose that q semidecides A in countable time. Let r be the decision algorithm for B that runs p and q simultaneously. Then r has a countable decision time α and, by Σ^1_2-reflection, we have $\alpha < \sigma$. But this is clearly false, since p's decision time is at least σ. □

6 Open Problems

The above results for sets of reals also hold for Turing machines with ordinal time and tape with virtually the same proofs, while the results for singletons do not. If we restrict ourselves to sets of natural numbers, then the suprema of decision and semidecision times equal λ, the supremum of clockable ordinals.

In the main results, we determined the suprema of various decision times, but we have not characterized the underlying sets.

Question 6.1 *Is there a precise characterization in the L-hierarchy of sets and singletons with countable decision, semidecision, and cosemidecision times?*

We would like to draw a further connection with classical results in descriptive set theory. A set of reals is called *thin* if it does not have an uncountable closed subset. It can be shown that $\{x \mid x \in L_{\lambda^x}\}$ is the largest thin semidecidable set (see [19, Definition 1.6]; this unpublished result of the third-listed author should appear in [4]). We ask if the same characterization holds for eventually semidecidable sets.

Question 6.2 *Is $\{x \mid x \in L_{\lambda^x}\}$ the largest thin eventually semidecidable set?*

In particular, is every eventually semidecidable singleton an element of L_{λ^x} (equivalently, L_{Σ^x})? An indication that these statements might be true is that one can show an analogous statement for null sets instead of countable sets: the largest ittm-semidecidable null set equals the largest ittm-eventually semidecidable null set. Related to Theorem 5.1, it is natural to ask whether a thin semidecidable set can have halting times unbounded in ω_1.

We did not study specific values of decision times in this paper. Note that this is an entirely different type of problem, since it is sensitive to the precise definition of ittm's. Regarding Section 3.3, we know from [9, Theorem 8.8] that admissible ordinals are never clockable and from [22, Theorem 50] that any ordinal that begins a gap in the clockable ordinals is always admissible. One can ask if an analogous result holds for decision times.

Question 6.3 *Is an ordinal that begins a gap of the (semi-)decision times always admissible?*

One can also ask about the decision times of single sets. By the bounding lemma (see Welch [21, Theorem 8]), no decidable set A has a countable admissible decision time. However, the semidecision time of a decidable set can be admissible. To see this, note that the set of indices as in the proof of Theorem 4.5 can be semidecided by a program with semidecision time σ. We do not know if this is possible for a set that is not decidable.

Notes

1. We mean that for an ordinal α, the set $\{\alpha\}$ is Σ_1-definable over L_{ω_1} or, equivalently, in V. The reason for writing L_{ω_1} here is to make the analogy with the definition of τ clear.

2. Roughly, this can be seen as follows: Given a code for α, split the tape into α many disjoint portions of length ω and construct the code level-wise. To pass from level ξ to level $\xi + 1$ requires computing the sets $\{a \in L_\xi : L_\xi \models \phi(a, p)\}$ for each $\in$-formula ϕ and each parameter $p \in L_\xi$. When ϕ is Σ_n, this can be done in $\omega^n + 1$ many steps. Doing this for all pairs (ϕ, p)—which can be arranged in order type ω using our code—can thus be done with time bound ω^ω.

3. For the definition of *fast*, see Lemma 4.7.

4. This argument is from [19, Theorem 2.6(3)].

References

[1] Barwise, K. J., *Admissible Sets and Structures*, Springer, Berlin, 1975. MR 0424560. 200

[2] Carl, M., *Ordinal Computability: An Introduction to Infinitary Machines*, vol. 9 of *Logic and Its Applications*, De Gruyter, Berlin, 2019. MR 4283501. DOI 10.1515/9783110496154. 200, 206, 208

[3] Carl, M. "Space and time complexity for infinite time Turing machines," *Journal of Logic and Computation*, vol. 30 (2020), pp. 1239–55. MR 4139022. DOI 10.1093/logcom/exaa025. 202

[4] Carl, M., P. Schlicht, and P. D. Welch, "Countable ranks at the first and second projective levels," in preparation, 2022. 199, 209

[5] Friedman, S.-D., and P. D. Welch, "Two observations regarding infinite time Turing machines," pp. 44–48 in *Bonn International Workshop on Ordinal Computability*, 2007, http://www.math.uni-bonn.de/ag/logik/events/biwoc/report.pdf 201, 208

[6] Gandy, R. O., "Proof of Mostowski's conjecture," *Bulletin de l'Académie Polonaise des Sciences (Série des Sciences Mathématiques, Astronomiques et Physiques)*, vol. 8 (1960), pp. 571–75. Zbl 0156.01101. MR 0126383. 205

[7] Hinman, P., *Recursion-Theoretic Hierarchies*, Springer, Berlin, 1978. MR 0499205. 197

[8] Hjorth, G., "Vienna notes on effective descriptive set theory and admissible sets," preprint, 2010, http://www.math.uni-bonn.de/people/logic/events/young-set-theory-2010/Hjorth.pdf. 205

[9] Hamkins, J. D., and A. Lewis, "Infinite time Turing machines," *Journal of Symbolic Logic*, vol. 65 (2000), pp. 567–604. MR 1771072. DOI 10.2307/2586556. 197, 198, 199, 200, 201, 203, 204, 210

[10] Jech, T. J., *Set Theory*, 3rd ed., Springer, Berlin, 2003. MR 1940513. 201, 202

[11] Kechris, A. S., *Classical Descriptive Set Theory*, vol. 156 of *Graduate Texts in Mathematics*, Springer, New York, 2012. MR 1321597. DOI 10.1007/978-1-4612-4190-4. 205

[12] Kechris, A. S., D. Marker, and R. L. Sami, "Π^1_1 Borel sets," *Journal of Symbolic Logic*, vol. 54 (1989), pp. 915–20. MR 1011178. DOI 10.2307/2274751. 199

[13] Mathias, A. R. D., "Provident sets and rudimentary set forcing," *Fundamenta Mathematicae*, vol. 230 (2015), pp. 99–148. Zbl 1373.03104. MR 3337222. DOI 10.4064/fm230-2-1. 202

[14] Rogers, H., *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, New York, 1967. 202

[15] Sacks, G. E., *Higher Recursion Theory*, Springer, Berlin, 1990. MR 1080970. DOI 10.1007/BFb0086109. 202

[16] Spector, C., "Recursive well-orderings," *Journal of Symbolic Logic*, vol. 20 (1955), pp. 151–63. MR 0074347. DOI 10.2307/2266902. 205

[17] Spector, C., "Hyperarithmetic quantifiers," *Fundamenta Mathematicae*, vol. 48 (1960), pp. 313–20. MR 0120147. DOI 10.4064/fm-48-3-313-320. 205

[18] Suzuki, Y., "A complete classification of the the Δ^1_2 functions," *Bulletin of the American Mathematical Society*, vol. 70 (1964), pp. 246–53. Zbl 0166.26302. MR 0158819. DOI 10.1090/S0002-9904-1964-11104-6. 202

[19] Welch, P. D., "Eventually infinite time Turing degrees: Infinite time decidable reals," *Journal of Symbolic Logic*, vol. 65 (2000), pp. 1193–1203. Zbl 0959.03025. MR 1791371. DOI 10.2307/2586695. 198, 200, 209, 210

[20] Welch, P. D., "The length of infinite time Turing machine computations," *Bulletin of the London Mathematical Society*, vol. 32 (2000), pp. 129–36. Zbl 1016.03040. MR 1734198. DOI 10.1112/S0024609399006657. 199, 201, 208

[21] Welch, P. D., "Bounding lemmata for non-deterministic halting times of transfinite Turing machines," *Theoretical Computer Science*, vol. 394 (2008), pp. 223–28. Zbl 1136.03027. MR 2404628. DOI 10.1016/j.tcs.2007.12.014. 210

[22] Welch, P. D., "Characteristics of discrete transfinite time Turing machine models: Halting times, stabilization times, and normal form theorems," *Theoretical Computer Science*, vol. 410 (2009), pp. 426–42. Zbl 1160.68013. MR 2493990. DOI 10.1016/j.tcs.2008.09.050. 200, 204, 208, 210

Acknowledgments

We thank our anonymous referee for several remarks that helped to clarify and improve the exposition of our work. This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement 794020 of the second-listed author ("IMIC: Inner Models and Infinite Computations"). He was also partially supported by FWF grant I4039.

Carl
Europa-Universität Flensburg, Institut für mathematische, naturwissenschaftliche und technische Bildung, Abteilung für Mathmematik und ihre Didaktik, Flensburg, Germany; merlin.carl@uni-flensburg.de

Schlicht
School of Mathematics, University of Bristol, Bristol, United Kingdom, and Institute for Mathematics, University of Vienna, Vienna, Austria; philipp.schlicht@bristol.ac.uk

Welch
School of Mathematics, University of Bristol, Bristol, United Kingdom; p.welch@bristol.ac.uk