*Erschienen in: The Journal of Symbolic Logic ; 80 (2015), 4. - S. 1116-1130*
*https://dx.doi.org/10.1017/jsl.2015.8*

# OPTIMAL RESULTS ON RECOGNIZABILITY FOR INFINITE TIME REGISTER MACHINES

MERLIN CARL

**Abstract.** Exploring further the properties of $ITRM$-recognizable reals started in [1], we provide a detailed analysis of recognizable reals and their distribution in Gödels constructible universe $L$. In particular, we show that new unrecognizable reals are generated at every index $\gamma \geq \omega_{\omega}^{CK}$. We give a machine-independent characterization of recognizability by proving that a real $r$ is recognizable if and only if it is $\Sigma_1$-definable over $L_{\omega_{\omega}^{CK,r}}$ and that $r \in L_{\omega_{\omega}^{CK,r}}$ for every recognizable real $r$ and show that either every or no $r$ with $r \in L_{\omega_{\omega}^{CK,r}}$ generated over an index stage $L_\gamma$ is recognizable. Finally, the techniques developed along the way allow us to prove that the halting number for $ITRM$s is recognizable and that the set of $ITRM$-computable reals is not $ITRM$-decidable.

**§1. Introduction.** Infinite Time Register Machines ($ITRM$s) were introduced by Peter Koepke and Russell Miller in [9] and provide a transfinite analogue of classical register machines in much the same way as the Infinite Time Turing Machines ($ITTM$s) introduced in [5] generalize classical Turing machines. We give a brief sketch of their behaviour here; detailed definitions can be found in [9] and [6]. These papers also contain most of the results we use in this paper.

Classical register machines, as e.g., described in [3], have as their 'hardware' finitely many registers, each of which can store a single natural number. A program for a register machine consists of finitely many numbered lines, each of which contains a single command, where the available commands are incrementing a register content by 1, setting a register content to 0, copying the content from one register to another, the oracle call (which replaces the content $i$ of some register $R$ with the $i$th bit of an oracle $x \subseteq \omega$) and a conditional jump which proceeds to a fixed program line when two registers have the same content and otherwise proceeds with the next program line. When the machine arrives at a line index which is not part of the program, the machine halts.

To generalize these machines to the transfinite, we keep the 'hardware' - finitely many registers, each of which stores a single natural number - as well as the 'software', i.e., the notion of program. What is changed is the 'working time': While the

---

2010 *Mathematics Subject Classification.* 03D65, 03D60, 03E$xx$, 03E45.
*Key words and phrases*. Infinite Time Computability, Generalized Computability, Infinite Time Register Machines, Recognizable Reals.

working time of a halting classical register machine just a natural number, $ITRM$s run along arbitrary ordinals. To make sense of this, we need to explain what state the machine is supposed to assume at a limit ordinal. At successor ordinals, we define the computation in the same way as for classical register machines. If $\lambda$ is a limit ordinal, we set the content $R_{i\lambda}$ of the $i$-th register $R_i$ at time $\lambda$ to $\liminf_{\iota<\lambda} R_{i\iota}$ if and only if this limit $< \omega$, and otherwise to 0. Likewise, the active program line $Z_\lambda$ to be carried out in the $\lambda$th step is $\liminf_{\iota<\lambda} Z_\iota$, (where the limit is always finite as the set of lines is finite).

Definition 1.1. $x \subseteq \omega$ is $ITRM$-computable in the oracle $y \subseteq \omega$ if and only if there exists an $ITRM$-program $P$ such that, for $i \in \omega$, $P$ with oracle $y$ stops for every natural number $j$ in its first register at the start of the computation and returns 1 if and only if $j \in x$ and otherwise returns 0. A real $ITRM$-computable in the empty oracle is simply called $ITRM$-computable.

The notion of the recognizability of a real was introduced by Hamkins and Lewis in [5] for Infinite Time Turing machines. The adaption to $ITRM$s is straightforward:

Definition 1.2. Let $r \subseteq \omega$. Then $r$ is recognizable if and only if there is an $ITRM$-program $P$ such that $P^x$ stops with output 1 if and only if $x = r$ and otherwise stops with output 0.

In the last section of [6], we showed that there are recognizable reals that are not computable. This is analogous to the lost melody theorem for infinite time Turing machines ($ITTM$s) demonstrated in [5]. The example given there was a $<_L$-minimal real coding an $\in$-minimal $L_\alpha$ modelling $ZF^-$ (see below). Here, we will give a much more natural example by showing that the real coding the halting problem for $ITRM$s is recognizable. After that, in [1], we obtained some more results on $ITRM$-recognizable reals and their relation to Gödel's constructible universe $L$. In particular, we showed that all recognizable reals are constructible, but that the recognizable reals do not form an initial segment of the constructible reals in the canonical well-ordering $<_L$ of $L$. In fact, there are quite large gaps: If $\sigma$ is the supremum of the ordinals indexing levels of the $L$-hierarchy at which new recognizable reals appear and $\alpha < \sigma$, then there is a $<_L$-interval of length $> \alpha$ in the constructible reals without a recognizable element, while there are recognizable reals that are $<_L$-greater than all elements of that interval. Here, our goal is to give a more precise picture of $ITRM$-recognizability and their distribution among the constructible reals. For example, it is easy to see that all computable reals are recognizable, so that, by the theorem cited above, all reals in $L_{\omega_\omega^{CK}}$ are recognizable. Is there any $\alpha > \omega_\omega^{CK}$ such that the reals in $L_\alpha$ are still all recognizable? It turns out that this is not the case: Whenever $\gamma > \omega_\omega^{CK}$ is such that $L_{\gamma+1} - L_\gamma$ contains a real number at all, it also contains a nonrecognizable real. Furthermore, we give a machine-independent, purely set-theoretical characterization of $ITRM$-recognizability and lower estimates on the length of gaps in the recognizable reals and some information on ordinals starting them. The techniques used in these results will then be applied to give a characterization of $ITRM$-decidable sets of reals and show that the set of $ITRM$-computable reals is not decidable.

Most of our notation is standard. $ZF^-$ is $ZF$ set theory without the power set axiom in the version described in [4]. For an $ITRM$-program $P$, $P^x(i)\downarrow = j$ means that the program $P$ with oracle $x$ with initial input $i$ in its first register stops with output $j$ in register 1. We will write $P^x\downarrow$ for $P^x(0)\downarrow$. When we write that $P^x$ 'stops in less than $\alpha$ many steps', we mean that $P^x(i)$ stops in less than $\alpha$ many steps for every $i \in \omega$. $On$ is the class of ordinals. Unless stated otherwise, small Greek letters refer to ordinals. For, $\iota \in On$, we denote by $\omega_\iota^{CK}$ the smallest admissible ordinal greater than $\omega_\gamma^{CK}$ when $\iota = \gamma + 1$ and $sup\{\omega_\gamma^{CK} \mid \gamma < \iota\}$ when $\iota$ is a limit ordinal. When we consider admissible ordinals relative to a real $x$, we write $\omega_\iota^{CK,x}$. For $X \subseteq L_\alpha$, $\Sigma_1^{L_\alpha}\{X\}$ denotes the $\Sigma_1$-Skolem hull of $X$ in $L_\alpha$ and $\Sigma_\omega^{L_\alpha}\{X\}$ denotes the elementary hull of $X$ in $L_\alpha$. When $H$ is a $\Sigma_1$-substructure of some $L_\alpha$, then $\pi : H \cong L_\gamma$ and $\pi : H \rightarrow_{coll} L_\gamma$ denote the Mostowski collapse of $H$ to $L_\gamma$ with isomorphism $\pi$. Throughout the paper, $p : \omega \times \omega \rightarrow \omega$ denotes the usual bijection between $\omega \times \omega$ and $\omega$. At various places, we will code certain $\in$-structures by real numbers. This works as follows: Let $(M, \in)$ be countably infinite and transitive, and let $f : \omega \rightarrow M$ be a bijection. Then the real coding $(M, \in)$ according to $f$ is $\{p(i, j) \mid [f(i) \in f(j)]\}$. We say that $x$ codes $(M, \in)$ if and only if there is a function $f$ such that $x$ codes $(M, \in)$ according to $f$. If $(M, \in) \in L$, $cc(M)$ denotes the $<_L$-smallest real coding $(M, \in)$.

## §2. ITRMs.

We give here some basic results on $ITRM$s and admissible set theory which are relevant for our further development.

THEOREM 2.1. *Let $\mathbb{P}_n$ denote the set of $ITRM$-programs using at most n registers, and let $(P_{i,n} \mid i \in \omega)$ enumerate $\mathbb{P}_n$ in some natural way. Then the bounded halting problem $H_n^x := \{i \in \omega \mid P_{i,n}^x \downarrow\}$ is computable uniformly in the oracle $x$ by an $ITRM$-program.*

*Furthermore, if $P \in \mathbb{P}_n$ and $P^x\downarrow$, then $P_x$ halts in less than $\omega_{n+1}^{CK,x}$ many steps. Consequently, if $P$ is a halting $ITRM$-program, then $P^x$ stops in less than $\omega_\omega^{CK,x}$ many steps.*

PROOF. The corresponding results from [9] easily relativize. ⊣

DEFINITION 2.2. $X \subseteq \mathfrak{P}(\omega)$ is $ITRM$-decidable if and only if there exists an $ITRM$-program $P$ such that $P^x\downarrow = 1$ if and only if $x \in X$ and $P^x\downarrow = 0$, otherwise.

COROLLARY 2.3. *Let $A$ be an $ITRM$-decidable set of reals and let $P$ be an $ITRM$-program. Then $M_1 := \{x \mid P^x\downarrow\}$, $M_2 := \{x \mid \forall i \in \omega P^x(i)\downarrow\}$, and $M_3 := \{x \mid P^x$ computes an element of $A\}$ are decidable.*

PROOF. The decidability of $M_1$ and $M_2$ is immediate from Theorem 2.1. For $M_3$, let $Q$ be a program for deciding $A$. To decide $M_2$, proceed as follows: Given some real $x$ in the oracle, first check, whether $x \in M_2$. If not, then $x \notin A$. Otherwise, $P^x$ computes some real $y$, and we can use $Q$ to decide whether $y \in A$. ⊣

LEMMA 2.4. *Let $(\phi_i \mid i \in \omega)$ be a natural enumeration of the $\in$-formulas. There is an $ITRM$-program $P$ such that, for all $x \subseteq \omega$, $i \in \omega$, $\vec{v} = (v_1, \ldots, v_n)$ a finite sequence of natural numbers of the appropriate length coded by a natural number $\bar{v}$, $P^x(i, \vec{v})\downarrow = 1$ if and only if $x$ codes some $L_\alpha$ such that $\phi_i(x_1, \ldots, x_n)$ holds in $L_\alpha$, where $x_1, \ldots, x_n$ are the elements coded by $v_1, \ldots, v_n$, respectively, and otherwise*

$P^x(i, \vec{v}) \downarrow= 0$. *The same holds with a recursive set $S$ of formulas instead of one single formula $\phi$, where $i$ is then a code for a Turing program enumerating $S$.*

PROOF. This is Corollary 13 of [1]. ⊣

DEFINITION 2.5. An ordinal $\alpha$ is called $\Sigma_1$-fixed if and only if there exists a $\Sigma_1$-statement $\phi$ such that $\alpha$ is minimal with the property that $L_\alpha \models \phi$. Let $\sigma$ denote the supremum of the $\Sigma_1$-fixed ordinals.

DEFINITION 2.6. An ordinal $\gamma$ is called an index if and only if $L_{\gamma+1} - L_\gamma$ contains a subset of $\omega$.

THEOREM 2.7. *Denote by $RECOG$ the set of recognizable reals. Then $RECOG \subseteq L_\sigma$. Furthermore, for each $\gamma < \sigma$, there exists $\alpha < \sigma$ such that $[\alpha, \alpha+\gamma]$ contains unboundedly many indices, but*
$(L_{\alpha+\gamma} - L_\alpha) \cap RECOG = \emptyset$.

PROOF. See [1]. ⊣

THEOREM 2.8. *Let $x, y \subseteq \omega$. Then $x$ is $ITRM$-computable in the oracle $y$ if and only if $x \in L_{\omega_\omega^{CK,y}}[y]$.*

PROOF. This is a straightforward relativization of the main result of [8]. ⊣

LEMMA 2.9. *Let $A \neq \emptyset$ be an $ITRM$-decidable set of reals such that $a \in L_{\omega_\omega^{CK,a}}$ for all $a \in A$. Then the $<_L$-minimal element of $A$ is recognizable.*

PROOF. Let $a$ be the $<_L$-minimal element of such an $A$. By Theorem 2.8, there is an $ITRM$-program $P$ such that $P^a$ computes a code for some $L$-level $L_\alpha$ containing $a$. Let $Q_A$ be an $ITRM$-program deciding $A$. Now $a$ can be recognized as follows: Given some $x \subseteq \omega$ in the oracle, first check whether $P^x(i) \downarrow$ for all $i \in \omega$, using a halting problem solver for $P$ which exists by Theorem 2.1. If not, then $x \neq a$. Otherwise, test whether $P^x$ computes a code $c$ for an $L$-level containing $x$. This can be done using Lemma 2.3 and Lemma 2.4. If not, then $x \neq a$. Otherwise, test whether $Q_A(x) \downarrow= 1$. If not, then $x \notin A$, so $x \neq a$. Otherwise, use $c$ to search through all reals below $x$ in $<_L$ for a real $z <_L x$ such that $Q_A(z) \downarrow= 1$. If such a real is found, then $x \neq a$. Otherwise, $x = a$. ⊣

We will need the following result of Jensen and Karp:

THEOREM 2.10. *Let $\alpha \in On$ be a limit of admissible ordinals, and let $\phi$ be a $\Sigma_1$-statement. Then $\phi$ is absolute between $L_\alpha$ and $V_\alpha$.*

PROOF. See [7]. ⊣

## §3. Unrecognizable Reals Everywhere.

The goal of this section is to show the result announced above: If $\gamma > \omega_\omega^{CK}$ is such that $L_{\gamma+1} - L_\gamma$ contains a real number, then it already contains a nonrecognizable real number. To show this, we will use Cohen-forcing over models of $KP$. A general reference for the forcing technique is [10]. For forcing over set theories weaker than $ZFC$, we refer the reader to [12], [2], and [11].

THEOREM 3.1. *Let $\mathbb{P} \in L_\xi$ be a notion of forcing, where $\xi$ is indecomposable, let $\gamma > \xi$ be admissible and let $G$ be $\mathbb{P}$-generic over $L_\gamma$. Then $L_\xi[G]$ in the sense of relative constructibility coincides with $L_\xi[G]$ in the sense of generic extensions.*

PROOF. This follows from the proof of Proposition 9.5 of [11]. ⊣

THEOREM 3.2. *Let $x \in RECOG$. Then $x \in L_{\omega_\omega^{CK,x}}$. Consequently, if $x$ is recognizable, but not ITRM-computable, we have $\omega_\omega^{CK,x} > \omega_\omega^{CK}$.*

PROOF. Let $P$ be a program that recognizes $x$.
Then $L_{\omega_\omega^{CK,x}}[x] \models \exists y P^y \downarrow = 1$. Now $\phi \equiv \exists y P^y \downarrow = 1$ is a (set theoretical) $\Sigma_1$-statement, stating that there are a real $y$ and a set $c$ such that $c$ codes the $P$-computation in the oracle $y$ and ends with 1. By Jensen-Karp (see Theorem 2.10), this is absolute between $V_\alpha$ and $L_\alpha$ whenever $\alpha$ is a limit of admissible ordinals. Now $\omega_\omega^{CK,x} = \sup\{\omega_i^{CK,x} \mid i \in \omega\}$ is a limit of admissible ordinals, so $\phi$ is absolute between $V_\alpha$ and $L_{\omega_\omega^{CK,x}}$. Also, as $\phi$ is $\Sigma_1$, it is upwards absolute. Hence $L_{\omega_\omega^{CK,x}}[x] \models \phi \implies V_{\omega_\omega^{CK,x}} \models \phi \implies L_{\omega_\omega^{CK,x}} \implies L_{\omega_\omega^{CK,x}}[x]$, so $\phi$ is absolute between $L_{\omega_\omega^{CK,x}}[x]$ and $L_{\omega_\omega^{CK,x}}$. As $\phi$ holds in $L_{\omega_\omega^{CK,x}}[x]$, it follows that $\phi$ holds in $L_{\omega_\omega^{CK,x}}$. So $L_{\omega_\omega^{CK,x}}$ contains a real $r$ such that $P^r \downarrow = 1$. By absoluteness of computations, $P^r \downarrow = 1$ also holds in $V$. So $P^r \downarrow = 1$. As $P$ recognizes $x$, it follows that $x = r$. Hence $x \in L_{\omega_\omega^{CK,x}}$.

Now let $x$ be recognizable, but not computable. As $x$ is not computable, we have $x \notin L_{\omega_\omega^{CK}}$. By the first part of the claim, $x \in L_{\omega_\omega^{CK,x}}$. Hence $\omega_\omega^{CK,x} > \omega_\omega^{CK}$. ⊣

This immediately leads to the following dichotomy:

COROLLARY 3.3. *If $x \subseteq \omega$ is such that $\omega_\omega^{CK,x} = \omega_\omega^{CK}$, then either $x$ is ITRM-computable or $x$ is not ITRM-recognizable.*

PROOF. If $x$ is *ITRM*-computable, then $x$ is clearly *ITRM*-recognizable (see [1]). If $x$ is not *ITRM*-computable, then $x \notin L_{\omega_\omega^{CK}}$, hence $x \notin L_{\omega_\omega^{CK,x}}$ if $\omega_\omega^{CK,x} = \omega_\omega^{CK}$. By Theorem 3.2 then, $x$ is not recognizable. ⊣

For the next lemma, we need the following result of Mathias:

THEOREM 3.4. *If $M$ is admissible, $\mathbb{P} \in M$ and $G$ is an $(M, \mathbb{P})$-generic filter meeting each dense open subclass of $M$ that is the union of a $\Sigma_1(M)$ and a $\Pi_1(M)$ class, then $M^{\mathbb{P}}[G]$ is admissible.*

PROOF. This is Theorem 10.11 of [11]. ⊣

LEMMA 3.5. *Let $\alpha$ be admissible, $(P, \leq) \in L_\alpha$ be a notion of forcing and $G$ be a filter on $P$ such that $P \cap D \neq \emptyset$ for every dense subset $D$ of $P$ such that $D \in L_{\alpha+1}$. Then $L_\alpha[G]$ is admissible.*

PROOF. This follows from Theorem 3.4, since unions of $\Sigma_1(L_\alpha)$ and $\Pi_1(L_\alpha)$-definable subsets of $P$ are clearly elements of $L_{\alpha+1}$. ⊣

COROLLARY 3.6. *Let $\gamma \geq \omega_\omega^{CK}$, let $(P, \leq_P)$ be the notion of forcing for adding a Cohen real (i.e., $P$ consists of the finite partial functions from $\omega$ to 2 and $x \leq_P y$ if and only if $y \subseteq x$) and let $G$ be a filter on $(P, \leq)$ which intersects every dense $D \subseteq P$ such that $D \in L_\gamma$. Then $L_{\omega_i^{CK}}[G]$ is admissible for every $i \in \omega$.*

PROOF. This is immediate from Lemma 3.5 as $L_\gamma \supseteq L_{\omega_\omega^{CK}} \supseteq L_{\omega_i^{CK}+1}$ for all $i \in \omega$. ⊣

The following will be used to show that, for each index $\gamma \geq \omega_\omega^{CK}$, $L_{\gamma+1} - L_\gamma$ contains an unrecognizable real.

THEOREM 3.7. *Let $\gamma \geq \omega_\omega^{CK}$ be an index. Then there exists $x \in L_{\gamma+1} - L_\gamma$ with $\omega_i^{CK,x} = \omega_i^{CK}$ for all $i \in \omega$. In particular, this implies that $\omega_\omega^{CK,x} = \omega_\omega^{CK}$.*

Proof. Let $P$ be the notion of forcing for adding a Cohen real (see above). Let $G$ be an $L_\gamma$-generic filter on $P$ (i.e., $G$ intersects every dense subset of $P$ which lies inside $L_\gamma$). By Corollary 3.6, $L_{\omega_i^{CK}}[G]$ is then admissible for all $i \in \omega$.

Now let $x := \bigcup G \in \mathfrak{P}(\omega)$. We show that $L_{\omega_i^{CK}}[G] = L_{\omega_i^{CK}}[x]$: As $x = \bigcup G \in L_{\omega_i^{CK}}[G]$, we have $L_{\omega_i^{CK}}[x] \subseteq L_{\omega_i^{CK}}[G]$ and $G \in L_{\omega_i^{CK}}[x]$ (since $G$ is definable from $x$), hence also $L_{\omega_i^{CK}}[G] \subseteq L_{\omega_i^{CK}}[x]$. More generally, if $\alpha$ is additively indecomposable (which certainly holds for $\alpha$ admissible) and if we have $x \in L_\alpha[y]$ and $y \in L_\alpha[x]$, then $L_\alpha[x] = L_\alpha[y]$. To see this, let $z \in L_\beta[x]$ $(\beta < \alpha)$ and $x \in L_\gamma[y]$ $(\gamma < \alpha)$. Then $z \in L_\beta[x] \in L_{\gamma+\beta+1}[y] \subseteq L_\alpha[y]$, hence $L_\alpha[x] \subseteq L_\alpha[y]$. $L_\alpha[y] \subseteq L_\alpha[x]$ now follows by symmetry.

Now it follows that $L_{\omega_1^{CK}}[x] \models KP$, i.e., $\omega_1^{CK}$ is $x$-admissible, so that $\omega_1^{CK} \geq \omega_1^{CK,x} \geq \omega_1^{CK}$. Consequently, we get $\omega_1^{CK,x} = \omega_1^{CK}$. Now assume inductively that $\omega_i^{CK} = \omega_i^{CK,x}$ for some $i \in \omega$. It then follows that $L_{\omega_{i+1}^{CK}}[x] \models KP$, hence $\omega_{i+1}^{CK}$ is $x$-admissible and thus $\omega_{i+1}^{CK} > \omega_i^{CK,x} = \omega_i^{CK}$. But then $\omega_{i+1}^{CK} \geq \omega_{i+1}^{CK,x} \geq \omega_{i+1}^{CK}$, so $\omega_{i+1}^{CK,x} = \omega_{i+1}^{CK}$. This now gives us $\omega_i^{CK,x} = \omega_i^{CK}$ for all $i \in \omega$, so that $\omega_\omega^{CK,x} = \omega_\omega^{CK}$.

Next, we demonstrate that $G$ - and hence $x = \bigcup G$ - is definable over $L_\gamma$ and hence elements of $L_{\gamma+1}$. This can be seen as follows: As $\gamma$ is an index, there is $f : \omega \to L_\gamma$ surjective such that $f \in L_{\gamma+1}$ and hence definable over $L_\gamma$. Now define $g : ({}^{<\omega}2, \omega) \to^{<\omega} 2$ thus: Let $g(\vec{s}, i)$ be the lexically minimal element of $f(i)$, of which $\vec{s}$ is a subsequence if $f(i) \subseteq^{<\omega} 2$ is a dense subset of $P$ otherwise let $g(\vec{s}, i) = \vec{s}$. Now, define recursively: $h(0) = \emptyset$, $h(i+1) = g(h(i), i+1)$. This recursion can be carried out definably over $L_\gamma$ as follows: Set (for $i \in \omega$) $h(i) = x$ if and only if
$(i = 0 \wedge x = \emptyset) \vee (i \geq 1 \wedge \exists(\vec{s_0}, \dots, \vec{s_{i-1}})[\forall j \in i((j = 0 \wedge \vec{s_0} = \emptyset) \vee (\vec{s_j} = g(\vec{s_{j-1}}, j))) \wedge x = g(\vec{s_{i-1}}, i)])$. This is definable over $L_\gamma$, as $g$ is definable over $L_\gamma$ and all finite sequences of elements of $P$ are contained in $L_\omega$, and hence certainly in $L_{\omega_\omega^{CK}}$. It follows that $x \in L_{\gamma+1}$. That $\omega_i^{CK,x} = \omega_i^{CK}$ holds for all $i \in \omega$ was seen above.

Finally, we show that $x \notin L_\gamma$: Roughly, this follows immediately from the fact that $G$ is definable from $x$ and that $G$ is generic over $L_\gamma$ as in the case of Cohen-forcing for $ZFC$ models. More precisely, let $z \in \mathfrak{P}(\omega) \cap L_\gamma$. Also, let $\beta < \gamma$ be minimal such that $z \in L_{\beta+1} - L_\beta$. Then $D_z := \{b \in^{<\omega} 2 \mid \exists i \in \omega b(i) \neq z(i)\}$ is dense in $P$ and definable over $L_\beta$, hence an element of $L_\gamma$. Consequently, every $D_z$ has nonempty intersection with every $L_\gamma$-generic filter $G$, so that $\bigcup G \neq z$. As this holds for all $z \in L_\gamma$, we get $x = \bigcup G \notin L_\gamma$. ⊣

We can now show that new unrecognizable reals appear wherever possible, i.e., are generated at every index stage:

Theorem 3.8. *Let $\gamma \geq \omega_\omega^{CK}$ be an index. Then $L_{\gamma+1} - L_\gamma$ contains an unrecognizable real.*

Proof. By Theorem 3.7, $L_{\gamma+1} - L_\gamma$ contains a real $x$ such that $\omega_\omega^{CK,x} = \omega_\omega^{Ck}$. By Corollary 3.3, $x$ is not recognizable. ⊣

## §4. The halting number is recognizable.

We obtain a very natural lost melody by showing that the halting number for *ITRM*s is in fact recognizable. Fix a natural well-ordering $(P_i \mid i \in \omega)$ of the $ITRM$-programs in order type $\omega$ by e.g., sorting the programs lexicographically. The halting number $h$ for $ITRM$s is then defined as $h := \{i \in \omega \mid P_i \downarrow\}$. This real $h$ is natural insofar its definition is purely internal to $ITRM$s (e.g., not in any way related to $L$) and it is also arguably the first noncomputable real coming to mind.

We start by showing that, given $h$, there is a universal $ITRM$:

LEMMA 4.1. *There is an $ITRM$-program $P$ such that, for every $(i, j) \in \omega^2$, we have $P^h(p(i, j)) \downarrow= k + 1$ if $P_i(j) \downarrow= k$ and $P^h(p(i, j)) \downarrow= 0$ if $P_i(j) \uparrow$. That is, $P$ can, given $i$, compute the function computed by $P_i$.*

PROOF. $P$ works as follows: Given $i$ and $j$, first use $h$ to check whether $P_i(j) \downarrow$. If $P_i(j) \uparrow$, $P$ returns 0. Otherwise, we carry out the following procedure for each $k \in \omega$: Compute (which can be done with a standard register machine, in fact) an index $l = l(i, j, k)$ such that $P_l \downarrow$ if and only if $P_i(j) \downarrow= k$. $P_l$ will use a halting problem solver for $P_i$ (which can be easily obtained from $P_i$), i.e., a sub-program $Q$ such that $Q(j) \downarrow= 1$ if and only if $P_i(j) \downarrow$ and $Q(j) \downarrow= 0$, otherwise. If it turns out that $Q(j) = 0$, then $P_l$ enters an infinite loop. Otherwise, we wait until $P_i(j)$ has stopped and check whether the outcome is $k$. If it is, we stop, otherwise we enter an infinite loop. (Note that $P$ is not required to do all this; it is only required that $P$ can compute a code for a program that does this, which is in fact easy).

Using $l$ and $h$, we can easily check whether $P_i(j) \downarrow= k$. If so, we return $k + 1$. Otherwise, we continue with $k + 1$.
As $P_i(j) \downarrow$ is already clear at this point, this has to lead to the value of $P_i(j)$ after finitely many iterations. ⊣

The next step is that, using $h$, a code for $L_{\omega_i^{CK}}$ can be computed uniformly in $i$.

COROLLARY 4.2. *There is an $ITRM$-program $Q$ such that, for every $i \in \omega$, $Q^h(i)$ computes a code for $L_{\omega_i^{CK}}$. (I.e.: $Q^h(n)$ halts for every $n \in \omega$ and $\{j \in \omega \mid Q^h(p(i, j) \downarrow= 1\}$ will be a code for $L_{\omega_i^{CK}}$.)*

PROOF. First note that codes for $L_{\omega_i^{CK}}$ are uniformly recognizable in $i$, i.e., there is a program $R$ such that, for every $i \in \omega$, $x \subseteq \omega$, $R^x(i) \downarrow= 1$ if and only if $x$ codes $L_{\omega_i^{CK}}$ and otherwise $R^x(i) \downarrow= 0$. This can be obtained using the well-foundedness checker combined with the first-order checker described in [6] for $V = L + KP+$'There are exactly $i - 1$ admissible ordinals'.

Using $h$, we can now run through $\omega$, first testing whether $P_k(j)$ will halt for each $j \in \omega$ and then, using $P$ from the last lemma, whether $P_k$ will compute a code for $L_{\omega_i^{CK}}$. (We can evaluate $P_k(j)$ for every $j$ using $P$ from the last lemma and then use $R$ to recognize whether the computed number is a code.)

As $L_{\omega_i^{CK}}$ has $ITRM$-computable codes, the minimal index $l$ such that $P_l$ computes a code for $L_{\omega_i^{CK}}$ will eventually be found in this way.
After that, we can, again using $P$ from the last lemma, evaluate $l$ to compute the desired code. ⊣

These bits can now be put together to form a code for $L_{\omega_\omega^{CK}}$. This code will be a bit different from the codes considered so far, as we allow one element of the coded structure to be represented by arbitrary many elements of $\omega$.

DEFINITION 4.3. Let $(X, \in)$ be a transitive $\in$-structure. Furthermore, let $f : \omega \to X$ be surjective. Then $\{p(i, j) \mid f(i) \in f(j)\}$ is called an odd code for $(X, \in)$.

Odd codes can be evaluated in the same way that the codes we used so far could. The possibility of elements appearing repeatedly hinders none of those methods. It is helpful, however, to note that the equality is computable:

PROPOSITION 4.4. *There is an ITRM-program $\bar{T}$ such that, for every odd code $x$ for a well-founded, transitive $\in$-structure $(X, \in)$ (with associated function $f : \omega \to X$) and all $i, j \in \omega$, $\bar{T}^x(p(i, j)) \downarrow= 1$ if and only if $f(i) = f(j)$ and $\bar{T}^x \downarrow= 0$, otherwise.*

*Furthermore, there is an ITRM-program $T$ such that, for every two odd codes $x$ and $y$ for well-founded, transitive $\in$-structures $(X, \in)$ and $(Y, \in)$ (with associated functions $f_1$ and $f_2$), $T^{x \oplus y}(p(i, j)) \downarrow= 1$ if and only if $f_1(i) = f_2(j)$ and $T^{x \oplus y}(p(i, j)) \downarrow= 0$, otherwise.*

*Finally, there is an ITRM-program $T_\in$ such that, for every two odd codes $x$ and $y$ for well-founded, transitive $\in$-structures $(X, \in)$ and $(Y, \in)$ (with associated functions $f_1$ and $f_2$), $T_\in^{x \oplus y}(p(i, j)) \downarrow= 1$ if and only if $f_1(i) \in f_2(j)$ and $T_\in^{x \oplus y}(p(i, j)) \downarrow= 0$, otherwise.*

PROOF. This is an easy application of the techniques developed in [6]. We give a brief impression how this works: To decide, given $x, y \subseteq \omega$ coding transitive $\in$-structures and $i, j \in \omega$, whether $i$ and $j$ represent the same element, we use a stack. Initially, the stack contains $p(i, j)$. We then need to decide whether $f_1(i) \subseteq f_2(j)$ and $f_2(j) \subseteq f_1(i)$. To see the former, we use $x$ to successively consider the natural numbers coding elements of $f_1(i)$. For each such natural number $n$, we search, using $y$, through all natural numbers coding elements of $f_2(j)$. For each such element $m$, we put $p(n, m)$ on our stack and decide whether $f_1(n) = f_2(m)$. When such an $m$ is found, we replace the content of the stack register with $p(i, j)$ (this ensures that the lim infs will be as desired). When all $m$ have been tried and no $m$ with $f_2(m) = f_1(n)$ has been found, then $f_1(i) \neq f_2(j)$. Otherwise, we continue with the next $n$. When an $m$ with $f_2(m) = f_1(n)$ has been found for every such $n$, we have $f_1(i) = f_2(j)$. That the algorithm must terminate follows from the well-foundedness of $(X, \in)$ and $(Y, \in)$. ⊣

LEMMA 4.5. *There is an ITRM-program $S$ such that $S^h$ computes an odd code $c$ for $L_{\omega_\omega^{CK}}$.*

PROOF. The idea is to reserve $\omega$ bits for coding $L_{\omega_i^{CK}}$; in one portion (the $i$-th portion), we use $Q^h$ to compute a code $c_i$ for $L_{\omega_i^{CK}}$ with corresponding surjection $f_i$. Then we use $T$ from the last proposition to relate the portions.

More precisely, we construct $c$ as follows: First, for all $i, j, k \in \omega$, we let $p(p(j, i), p(k, i)) \in x$ if and only if $p(j, k) \in c_i$. It remains to decide the bits of the form $p(p(j, i_1), p(k, i_2))$ with $i_1 \neq i_2$. This corresponds to the question whether $f_{i_1}(j) \in f_{i_2}(k)$, which can be answered using $T_\in$ from Proposition 4.4. ⊣

THEOREM 4.6. *Let $h := \{i \in \omega \mid P_i \downarrow\}$ be the set of indices of halting ITRM-programs. Then $h \in RECOG$.*

PROOF. We first claim that the set of odd codes for $L_{\omega_\omega^{CK}}$ is decidable. To test whether a real $x$ is an odd code of $L_{\omega_\omega^{CK}}$, check first whether $x$ codes a well-founded relation and then whether the structure coded by $x$ satisfies the statement

$V = L \wedge$'There is no largest admissible ordinal and there is no limit of admissible ordinals'$\wedge$'Every $x$ is contained in some admissible $L_\alpha$'. The techniques from the proof of Corollary 4.2 are easily adapted to this task, using Proposition 4.4.

Now let $x$ be the real in the oracle, and let $S$ be as in Lemma 4.5. As the set of odd codes for $L_{\omega_\omega^{CK}}$ is decidable, we can, by the third statement of Lemma 2.3, decide the set $\{y \subseteq \omega \mid P^y$ computes an odd code for $L_{\omega_\omega^{CK}}\}$. Consequently, we can check whether $S^x$ computes an odd code $c$ for $L_{\omega_\omega^{CK}}$. If not, return 0. Checking whether certain programs halt amounts to checking whether certain first-order statements (expressing that $P_i$ halts, i.e., there is a set coding a halting computation according to $P$) hold in $L_{\omega_\omega^{CK}}$, which can be done using $c$ by Lemma 2.4. We can then compare the results of the computation with the oracle number $x$. If the agree, then $x = h$, otherwise $x \neq h$. This identifies $h$. ⊣

As a consequence, the halting number $h$ is a minimal lost melody with respect to $ITRM$-reducibility:

Corollary 4.7. *Let $x \in RECOG - COMP$. Then there is an $ITRM$-program $P$ such that $\forall i \in \omega P^x(i) \downarrow = h(i)$.*

Proof. As $x \in RECOG - COMP$, it follows from Theorem 3.2 that $\omega_\omega^{CK,x} > \omega_\omega^{CK}$. Consequently, there is $i \in \omega$ such that $\omega_i^{CK,x} > \omega_\omega^{CK}$. As $h$ is definable over $L_{\omega_\omega^{CK}}$, we have $h \in L_{\omega_\omega^{CK}+1}$, so $h \in L_{\omega_i^{CK,x}} \subseteq L_{\omega_i^{CK,x}}[x] \subseteq L_{\omega_\omega^{CK,x}}[x]$. Hence $h$ is $ITRM$-computable from $x$, as desired. ⊣

§5. **Potential recognizability.** We saw above (via Jensen-Karp, see Theorem 3.2) that $x \in RECOG$ implies that $x \in L_{\omega_\omega^{CK,x}}$. Reals without this property are hence ruled out, we concentrate on those that have it.

Definition 5.1. $x \subseteq \omega$ is potentially recognizable if and only if $x \in L_{\omega_\omega^{CK,x}}$. We denote the set of potentially recognizable reals by $PRECOG$.

Theorem 5.2. *Let $\gamma$ be an index. Then either all potentially recognizable elements of $L_{\gamma+1} - L_\gamma$ are recognizable or none is.*

Proof. Suppose $a \in (L_{\gamma+1} - L_\gamma) \cap RECOG$ and $x \in (L_{\gamma+1} - L_\gamma) \cap PRECOG$. We want to show that $x \in RECOG$. Pick a program $Q$ that recognizes $a$. As $x \in PRECOG$, there is $i \in \omega$ such that $x \in L_{\omega_i^{CK,x}}$. In particular, we have $L_{\gamma+1} \in L_{\omega_i^{CK,x}}$. Hence $c = cc(L_{\gamma+1})$, the $<_L$-minimal real code for $L_{\gamma+1}$, is computable from $x$. Let $P$ be a program that computes $c$ from $x$.

To identify whether $y = x$ (with $y$ in the oracle), we first use a halting problem solver (see Theorem 2.1) for $P$ to check whether $P^y(i) \downarrow$ for all $i \in \omega$. If not, then $y \neq x$. If yes, we check whether $P^y$ computes a code $d$ for an $L$-level containing $y$. If not, then $y \neq x$. If yes, we check, as in the proof of the Lost Melody Theorem in [6], whether $d$ is $<_L$-minimal with that property (this is possible as $x \in L_{\gamma+1}$). If not, then $y \neq x$. If yes, we check whether the structure coded by $d$ contains a real $r$ such that $Q^r \downarrow = 1$. This can be done Lemma 2.3. If there is no such $r$, then $y \neq x$. If there is, we check whether the structure coded by $d$ contains an $L$-level that also contains $r$ (this checks the minimality of $\gamma$). If so, then $y \neq x$. Otherwise, we know that $P^y$ computes $c$. But in $c$, $x$ is coded by some fixed natural number $n$ which can be given to the program in advance. All that remains is hence to check whether $y$ is the number coded by $n$ in $c$. If not, then $y \neq x$, otherwise $y = x$. So this procedure recognizes $x$, hence $x \in RECOG$. ⊣

This allows us to give a characterization of recognizability in purely set-theoretical vocabulary, without reference to machines:

THEOREM 5.3. *Let $x \in PRECOG$. Then $x \in RECOG$ if and only if there exists a $\Sigma_1$-formula $\phi$ of set theory without parameters such that $x$ is the unique witness for $\phi$ in $L_{\omega_\omega^{CK,x}}$.*

PROOF. If $x \in RECOG$, then $x \in PRECOG$ by Theorem 3.2 above. Now, if $P$ is a program that recognizes $x$, then $P^x \downarrow= 1$ is $\Sigma_1$-expressable over $L_{\omega_\omega^{CK,x}}$. By upwards preservation of $\Sigma_1$-statements, if $L_{\omega_\omega^{CK,x}} \models P^y \downarrow= 1$ for some $y \neq x$, then $P^y \downarrow= 1$ in the real world, which contradicts the assumption that $P$ recognizes $x$.

On the other hand, if $x$ is definable as above, then let $L_\gamma$ be the first $L$-level containing $x$ such that $L_\gamma \models \phi(x)$. Then $\gamma < \omega_\omega^{CK,x}$, so $c := cc(L_\gamma)$ can be computed from $x$, say by program $Q$. Given a real $y$ and $c$ in the oracle, we can check whether $y \in L_\gamma$ and $L_\gamma \models \phi(y)$ hold.

Checking whether some oracle number $y$ is equal to $x$ then works as follows: Check (using Lemma 2.3 and the techniques from the proof of the Lost Melody theorem in [6]) whether $Q^y$ computes a minimal code for an $L$-level containing $y$, then check whether $\phi(y)$ holds in that $L$-level and then whether it fails in all earlier $L$-levels. If all of this holds, then $y = x$ (since $\Sigma_1$ is preserved upwards), otherwise $y \neq x$. ⊣

The same argument can be adapted to characterize $ITRM$-decidable sets of reals (a set $A \subseteq \mathfrak{P}(\omega)$ is $ITRM$-decidable if and only if there is an $ITRM$-program $P$ such that $P^x \downarrow= 1$ if and only if $x \in A$ and $P^x \downarrow= 0$, otherwise, for all $x \in \mathfrak{P}(\omega)$). $ITRM$-decidable sets of reals were considered and demonstrated to be $\Delta^1_2$ in [9].

THEOREM 5.4. *Let $A \subseteq \mathfrak{P}(\omega)$. Then $A$ is $ITRM$-decidable if and only if there exist $n \in \omega$ and a parameter-free $\Sigma_1$-formula $\phi(y)$ with a single free variable $y$ such that $A = \{x \in \mathfrak{P}(\omega) \mid L_{\omega_n^{CK,x}}[x] \models \phi(x)$.*

PROOF. Assume first that $A$ is $ITRM$-decidable. Let $P$ be an $ITRM$-program that decides $A$. The computation is contained in $L_{\omega_n^{CK,x}}[x]$ if $n$ is larger than the number of registers used by $P$. Hence membership of $x$ in $A$ can be expressed as the existence of a computation by $P$ with input $x$ that terminates with output 1, which is $\Sigma_1$ over $L_{\omega_n^{CK,x}}[x]$.

On the other hand, if $x \in A$ is characterized as described in the assumptions by a $\Sigma_1$-formula over $L_{\omega_n^{CK,x}}[x]$, we can $ITRM$-decide $A$ by computing a code for $L_{\omega_n^{CK,x}}[x]$ from $x$, then identifying the natural number representing $x$ in this code and finally using the code for evaluating $\phi$ in $L_{\omega_n^{CK,x}}[x]$. The only new step here is the computation of a (not necessarily $<_L$-minimal) code for $L_{\omega_n^{CK,x}}[x]$ uniformly in $x$ with a program that always halts. Using a halting problem solver $H$ as in Theorem 2.1 for $ITRM$s with $n + 3$ registers, this can be done as follows: Given $x$ in the oracle and taking $(P_j \mid j \in \omega)$ to be a natural enumeration of the $ITRM$-programs using at most $n + 3$ registers, we perform the following routine for every $i \in \omega$: First, check whether $P_i^x$ computes a real, i.e., whether $P_i^x(k)$ halts for every

$k \in \omega$. This can be done using $H$. If this is not the case, restart the routine with $i+1$. Otherwise, check, with the usual techniques, whether the real $r$ computed by $P_i^x$ is a code for $L_{\omega_n^{CK,x}}[x]$ (i.e., an admissible level of the $L[x]$-hierarchy containing exactly $n-1$ $x$-admissible ordinals). If not, restart the routine with $i+1$. Otherwise, an algorithm computing the desired code is found. ⊣

THEOREM 5.5. *For all $x \subseteq \omega$, $x$ is recognizable if and only if $x \in L_{\omega_\omega^{CK,x}}$ and $L_{\omega_\omega^{CK,x}} \models RECOG(x)$. In particular, if $L_\alpha \models ZF^-$ and $x \in \mathfrak{P}(\omega) \cap L_\alpha$, then $x \in RECOG$ holds if and only if $L_\alpha \models RECOG(x)$.*

PROOF. Suppose first that $x \in RECOG$, and let $P$ be a program that recognizes $x$. Then $x \in L_{\omega_\omega^{CK,x}}$ by Theorem 3.2. By [2], if $z \in L_\gamma$ and $\gamma^+$ is the smallest admissible ordinal greater than $\gamma$, then $\omega_1^{CK,z} \leq \gamma^+$. Inductively, we get that $\omega_i^{CK,z} \leq \gamma^{+i}$, where $\gamma^{+i}$ is the $i$th admissible ordinal above $\gamma$. Inductively, it follows that $\omega_\omega^{CK,z} \leq \omega_\omega^{CK,x}$ for all $z \in L_{\omega_\omega^{CK,x}}$ when $x$ is such that $x \in L_{\omega_\omega^{CK,x}}$. This implies that $P^z$ stops after at most $\omega_\omega^{CK,x}$ many steps for all $z \in L_{\omega_\omega^{CK,x}}$ and hence that $P^z$ can be carried out inside $L_{\omega_\omega^{CK,x}}$ for all $z \in L_{\omega_\omega^{CK,x}}$. Hence, since $P$ recognizes $x$, we have $L_{\omega_\omega^{CK,x}} \models P^z \downarrow= 0$ for all $z \neq x$ and furthermore $L_{\omega_\omega^{CK,x}} \models P^x \downarrow= 1$. Hence $L_{\omega_\omega^{CK,x}} \models RECOG(x)$.

On the other hand, assume that $x \in L_{\omega_\omega^{CK,x}}$ and that $L_{\omega_\omega^{CK,x}} \models RECOG(x)$. Hence $P^x \downarrow= 1$ and $P^z \downarrow= 0$ for all $z <_L x$. Now let $Q$ be a program such that $Q^x$ computes the $<_L$-minimal code of the first $L$-level containing $x$. Then $x$ can be recognized as follows: Given some real $r$ in the oracle, first check, using Lemma 2.3, whether $P^r \downarrow= 1$. If not, then $r \neq x$. Otherwise check - applying Lemma 2.3 to $Q$ - whether $Q^r(i) \downarrow$ for all $i \in \omega$. If not, then $r \neq x$. If yes, check whether $Q^r$ codes a minimal $L$-level containing $r$. If not, then $r \neq x$. If yes, check whether $Q^r$ is $<_L$-minimal with this property, using the usual strategy. If not, then $r \neq x$. Otherwise, use $Q^r$ (and the halting problem solver for $P$) to check whether there is any real $y <_L x$ such that $P^y \downarrow= 1$. If that is the case, then $r \neq x$. If it isn't, then $r$ is $<_L$-minimal with $P^r \downarrow= 1$ and hence $r = x$. ⊣

## §6. More on gaps.

DEFINITION 6.1 (See [1]). A strong substantial gap is an ordinal interval $[\alpha, \beta]$ with $\beta > \alpha$ such that every $\gamma \in [\alpha, \beta]$ is an index and such that $L_\beta - L_\alpha$ contains no recognizable reals, while $L_{\beta+1} - L_\beta$ and $L_\alpha - L_\gamma$ do contain recognizable reals for every $\gamma < \alpha$. A weak substantial gap is an ordinal interval $[\alpha, \beta]$ such that $\alpha$ is an index, the set of indices in that interval is unbounded in $\beta$ and such that $L_{\beta+1} - L_\alpha$ contains no recognizable reals.

Note that it is shown in [1] that strong substantial gaps exist (it suffices to see that there is some index $\gamma \in On$ with $(L_{\gamma+1} - L_\gamma) \cap RECOG = \emptyset$). We can now show that gaps in the recognizable reals must be infinite. The same reasoning in fact supports much stronger conclusions, as we shall presently see. A minor technical issue arises in the following arguments due to the fact that we consider levels of the $L$-hierarchy with only very weak closure properties, so that we cannot exclude the possibility that some real coding such a level (or its ordinal height) will actually be contained in it as an element, while our arguments need codes arising only at a later stage. We hence make the following definition:

Definition 6.2. If $\gamma < \omega_1$, we denote by $cc'(L_\gamma)$ the $<_L$-minimal real coding $L_\gamma$ which is not an element of $L_\gamma$.

Theorem 6.3. *There are no strong substantial gaps of finite length. Furthermore, strong gaps always start with limit ordinals.*

Proof. Assume for a contradiction that there is a strong substantial gap of length $i$, where $i \in \omega$. Let $\alpha \in On$ be minimal such that $[\alpha, \alpha+i]$ is a strong substantial gap of length $i$. It is easy to see that $cc(L_{\alpha+i})$ is recognizable by the usual arguments: Given $x$, check whether $x$ codes an $L$-level at which a strong substantial gap of length $i$ ends. This can be done by Lemma 2.4. The minimality of $x$ can then be checked by the routines for evaluating truth predicates described in the last section of [6]. there. By the results on the computational strength of $ITRM$s, one readily obtains that from the $<_L$-minimal code $c$ of $L_\alpha$ which is not an element of $L_\alpha$, we can compute $cc(L_{\alpha+i})$, say by program $P$. Note that, by the definition of a substantial strong gap. we will have $cc(L_{\alpha+i}) \in L_{\alpha+i+1}$. But this allows us to recognize $c$: Given the oracle $x$, first check (applying Lemma 2.3 to $P$) whether $P^x$ computes $cc(L_{\alpha+i})$ - which is possible as $cc(L_{\alpha+i})$ is recognizable. Now, in $cc(L_{\alpha+i})$, $c$ is represented by some integer $j$. It hence only remains to see whether $x$ is the number represented by $j$ in $cc(L_{\alpha+i})$, which is also easy to do.

This implies that $c$ is recognizable. But, by definition, $c \in L_{\alpha+1} - L_\alpha$. Hence $(L_{\alpha+i} - L_\alpha) \cap RECOG \neq \emptyset$, which contradicts the assumption that $\alpha$ starts a gap.

To see that, if $\alpha$ starts a strong substantial gap, $\alpha$ has to be a limit ordinal, we proceed as follows: Assume for a contradiction that $\alpha$ starts a strong substantial gap and $\alpha = \beta + 1$. Since $\alpha$ starts the gap, $L_\alpha - L_\beta$ contains a recognizable real $r$. We argue that $cc'(L_\alpha) \in L_{\alpha+1} - L_\alpha$ is recognizable, which contradicts the assumption that $\alpha$ starts a gap. A procedure for recognizing $cc'(L_\alpha)$ works as follows: Given $x$, simply check whether $x$ is the $<_L$-minimal code of a minimal $L$-level containing $r$ which is not contained in that level. This is possible since $r$ is recognizable. ⊣

Theorem 6.4. *If $\alpha$ starts a weak substantial gap $[\alpha, \beta]$, then $\beta \geq \omega_\omega^{CK,cc'(\alpha)}$.*

Proof. Assume that $\alpha$ starts a weak substantial gap $[\alpha, \beta]$ where $\beta < \omega_\omega^{CK,cc'(\alpha)}$, so that $\beta < \omega_i^{CK,cc'(\alpha)}$ for some minimal $i \in \omega$. By definition, $\alpha$ is an index, so that $cc'(\alpha) \in L_{\alpha+1}$. By definition of $cc'$, we may assume that $cc'(\alpha) \notin L_\alpha$. We now want to argue that $cc'(\alpha) \in RECOG$, which will be a contradiction to the assumption that $\alpha$ starts a gap. From $cc'(\alpha)$, one can compute $cc'(L_{\omega_i^{CK,cc'(\alpha)}})$ by Theorem 2.8. Let $P'$ be an $ITRM$-program computing $cc'(L_{\omega_i^{CK,cc'(\alpha)}})$ in the oracle $cc'(\alpha)$. Since $i \in \omega$ is a fixed natural number, we can use $i$ together with $P'$ to determine, for an arbitrary oracle $x$, whether $P'^x$ computes a $<_L$-minimal code for $L_{\omega_i^{CK,x}}$. We can hence also compute the $<_L$-minimal code for $L_{\beta+1}$ in the oracle $cc'(\alpha)$, using program $P$, say. By our assumption that $\beta$ ends the gap, we must have $RECOG \cap (L_{\beta+1} - L_\beta) \neq \emptyset$; say $r \in RECOG \cap (L_{\beta+1} - L_\beta)$, and let $Q$ be a program for recognizing $r$. Now, given $x$ in the oracle, we can determine whether $P^x$ computes the minimal code for an $L$-level containing a real $z$ such that $Q^z \downarrow = 1$. (This can be achieved by searching through the coded structure; since $r$ is recognized by $Q$, the calculation $Q^z$ will terminate for all reals $z$ from the coded structure.)

If this is not the case, then $x \neq cc'(\alpha)$. Otherwise, $P^x$ has computed $cc'(L_{\beta+1})$. In $cc'(L_{\beta+1})$, the real $cc'(\alpha)$ is represented by some fixed natural number $k \in \omega$ (which can hence be given to our program). We can now simply test whether $x$ is the real coded by $k$ in $cc(L_{\beta+1})$ by bitwise comparison. This allows us to recognize $cc'(\alpha) \in L_{\alpha+1} - L_\alpha$, which contradicts the assumption that $\alpha$ starts a gap. ⊣

We now show that ordinals starting substantial gaps cannot be too easily definable.

Definition 6.5. $\alpha \in \omega_1$ is admissibly $\Sigma_1$-describable if and only if there exists a $\Sigma_1$-formula $\phi$ of set theory without parameters such that $cc(\alpha)$ is the unique witness for $\phi(v)$ in $L_{\omega_\omega^{CK,cc(\alpha)}}$. If $\alpha$ is not admissibly $\Sigma_1$-describable, we call it admissibly $\Sigma_1$-indescribable.

Theorem 6.6. *Let $\alpha$ start a weak substantial gap. Then $\alpha$ is admissibly $\Sigma_1$-indescribable.*

Proof. Assume for a contradiction that $\alpha$ is $\Sigma_1$-describable and starts a weak substantial gap. Then $\alpha$ is an index, so that $cc'(\alpha) \in L_{\alpha+1} - L_\alpha$.

Now, if $\alpha$ was admissibly $\Sigma_1$-describable, we could compute from $cc'(\alpha)$ the $<_L$-minimal code of the first $L_\beta$ containing a witness for some $\Sigma_1$-statement $\phi$ which characterizes $\alpha$. Let $P$ be a program that achieves this. By the usual procedure, we can check for an arbitrary oracle $x$ whether $P^x$ computes a minimal code of a minimal $L$-level containing such a witness. Now we must have $cc'(\alpha) \in L_\beta$, so that $cc'(\alpha)$ is represented in $cc'(L_\beta)$ by some fixed natural number $k$. To determine whether $x = cc'(\alpha)$, it hence only remains to check whether $x$ is equal to the number represented by $k$ in the structure coded by the real computed by $P^x$, which is also possible. So $cc'(\alpha) \in L_{\alpha+1} - L_\alpha$ is recognizable, contradicting the assumption that $\alpha$ starts a gap. ⊣

In much the same way, we get:

Corollary 6.7. *Let $[\alpha, \beta]$ be a strong substantial gap, and let $\gamma \in [\alpha, \beta]$. Then $\gamma$ is admissibly $\Sigma_1$-indescribable.*

Proof. This follows by the same argument as above, since $\gamma$, being an element of a strong substantial gap, must be an index, which is the crucial property for this argument. ⊣

We have seen in Theorem 3.8 that, as soon as a new real appears in the $L$-hierarchy after $\omega_\omega^{CK}$ at all, a nonrecognizable real appears as well. In fact, reals appear that are not even potentially recognizable. This motivated us above to restrict our attention to potentially recognizable reals. Applying this here leads to the question whether at least the recognizability of potentially recognizable reals continues to hold for some stages beyond $L_{\omega_\omega^{CK}}$. This turns out to be true. In fact, the potentially recognizable reals continue being recognizable for quite a while after $L_{\omega_\omega^{CK}}$:

Theorem 6.8. *All elements of $L_{\omega_{\omega+\omega}^{CK}} \cap PRECOG$ are recognizable.*

Proof. This is clear for $x \in L_{\omega_\omega^{CK}}$. Hence let us assume from now on that $x \notin L_{\omega_\omega^{CK}}$. In the light of Theorem 5.2 (and the fact that, if $\gamma$ is an index, then an arithmetical copy of $L_\gamma$ appears in $L_{\gamma+1}$, see e.g., [2]), it suffices to show that, given $x \in L_{\omega_{\omega+\omega}^{CK}}$, the level $L_\gamma$ where $x$ appears has a recognizable $<_L$-minimal code $cc'(L_\gamma)$ not in $L_\gamma$. For this, it suffices to show that, for all indices $\gamma$ between $\omega_\omega^{CK}$ and $\omega_{\omega+\omega}^{CK}$, $cc'(L_\gamma)$ is recognizable. This can be seen as follows: Let $i \in \omega$ be minimal

such that $\gamma \in L_{\omega^{CK}_{\omega+i+1}}$. Then $cc(L_{\omega^{CK}_{\omega+i+1}})$ is $ITRM$-computable from $cc'(L_\gamma)$, as, by our assumption, $\gamma \geq \omega^{CK}_\omega$, so that $\omega^{CK,cc(L_\gamma)}_\omega \geq \omega^{CK}_{\omega+\omega}$ - and $L_{\omega^{CK}_{\omega+\omega}}$ can easily be seen to contain $cc'(L_{\omega^{CK}_{\omega+i+1}})$. From now on, let $P$ be an $ITRM$-program that computes $cc(L_{\omega^{CK}_{\omega+i+1}})$ from $cc'(L_\gamma)$.
We claim that $cc(L_{\omega^{CK}_{\omega+i+1}})$ is recognizable. This can be seen using the methods developed in the proof of the lost melody theorem in [6] by first checking whether the oracle real $r$ codes an $L$-level, then, whether this level is admissible, contains a single limit of admissible ordinals and further $i$ admissible ordinals greater than that limit and finally, whether $r$ is $<_L$-minimal with that property. (Note that the fixed natural number $i$ can be given to our program it advance.) Let $Q$ be a program that recognizes $cc(L_{\omega^{CK}_{\omega+i+1}})$.
Now, $cc'(L_\gamma)$, being an element of $L_{\omega^{CK}_{\omega+i+1}}$, is coded in $cc(L_{\omega^{CK}_{\omega+i+1}})$ by some natural number $j$ (which can also be given to our program in advance). Hence, given $cc(L_{\omega^{CK}_{\omega+i+1}})$, we can easily check whether some real $r$ is equal to $cc'(L_\gamma)$ by checking whether it is equal to the element of $cc(L_{\omega^{CK}_{\omega+i+1}})$ coded by $j$.
To recognize whether some real $r$ given in the oracle is equal to $c := cc'(L_\gamma)$, we hence proceed as follows: First check, using Lemma 2.3, whether $P^r(k)$ halts with output 1 or 0 for all $k \in \omega$. If not, then $r \neq c$. Otherwise, $P^r$ computes some real $r'$. Using $Q$, test whether or not $r' = cc(L_{\omega^{CK}_{\omega+i+1}})$. If not, then $r \neq c$. Otherwise, check whether $r$ is coded by $j$ in $r'$. If not, then $r \neq c$, otherwise $r = c$.
This shows that, whenever $\omega^{CK}_\omega \leq \gamma < \omega^{CK}_{\omega+\omega}$ is an index, then $cc'(L_\gamma)$ is a recognizable element of $L_{\gamma+1} - L_\gamma$. By our remark above, this shows that in fact all potentially recognizable reals in $L_{\omega^{CK}_{\omega+\omega}} - L_{\omega^{CK}_\omega}$ are recognizable. ⊣

§7. **Computability is not decidable.** As a final application of the techniques developed in this paper, we prove that the set of $ITRM$-computable reals is not $ITRM$-decidable.

THEOREM 7.1. *There is no $ITRM$-program $P$ such that, for $L \ni x \subseteq \omega$, $P^x \downarrow= 1$ if and only if $x$ is $ITRM$-computable and $P^x \downarrow= 0$ otherwise.*

PROOF. Assume for a contradiction that $P$ is such a program, and let $P$ use $n$ registers. Hence, since $P^x$ halts for all $x \subseteq \omega$, it follows that $P^x$ halts in less than $\omega^{CK,x}_{n+1}$ many steps. Now, if $x$ is Cohen-generic over $L_{\omega^{CK}_{n+1}+1}$, then, by our results above, we have $\omega^{CK,x}_{n+1} = \omega^{CK}_{n+1}$.

It is easy to see that $L_{\omega^{CK}_{n+2}}$ contains such a real $x$. As $x \in L_{\omega^{CK}_\omega}$, $x$ is $ITRM$-computable, so that $P^x \downarrow= 1$ in less than $\omega^{CK}_{n+1}$ many steps. Now $L_{\omega^{CK}_{n+1}}[x] \subseteq L_{\omega^{CK}_{n+2}}$, hence, as $L_{\omega^{CK}_{n+2}} \models P^x \downarrow= 1$, it follows from the upwards absoluteness of $\Sigma_1$-formulas that $L_{\omega^{CK}_{n+1}}[x] \models P^x \downarrow= 1$, where $x$ is the generic real. Hence, there is some finite subset $p$ of $x$ such that $p \Vdash P^{\bigcup \dot{G}} \downarrow= 1$, where $\dot{G}$ is the canonical name of the generic filter. Consequently, whenever $H \ni p$ is generic over $L_{\omega^{CK}_{n+1}+1}$ and $x = \bigcup H$, we have that $L_{\omega^{CK}_{n+1}}[x] \models P^x \downarrow= 1$. Now let $x$ be Cohen-generic over $L_{\omega^{CK}_\omega}$ such that $p \subseteq x$. It is easy to see that such an $x$ exists. As $x \notin L_{\omega^{CK}_\omega}$, it is not $ITRM$-computable. However, we have $L_{\omega^{CK}_{n+1}}[x] \models P^x \downarrow= 1$, so $P^x \downarrow= 1$ by absoluteness of computations. Hence $P^x \downarrow= 1$ for some noncomputable $x$, a contradiction. ⊣

§8. **Acknowledgments.** We are indebted to Philipp Schlicht for helpful discussions on forcing over $KP$ and suggesting several very helpful references. We also thank Philip Welch for suggesting the use of Cohen-forcing over $L_{\omega_\omega^{CK}}$ as a method for obtaining unrecognizable reals low down in the constructible hierarchy.

## REFERENCES

[1] M. Carl, *The distribution of ITRM-recognizable reals*. ***Annals of Pure and Applied Logic***, special issue for the CiE 2012, vol. 165 (2012), no. 9, pp. 1403–1417.

[2] C.T. Chong, *A recursion-theoretic characterization of constructible reals*. ***Bulletin of the London Mathematical Society***, vol. 9 (1977), pp. 241–24.

[3] N. Cutland, ***Computability. An introduction to recursive function theory***, Cambridge University Press, Cambridge, 1980.

[4] V. Gitman, J.D. Hamkins, and T. A. Johnstone, *What is the theory ZFC without power set?* arXiv:1110.2430 [math.LO].

[5] J. Hamkins and A. Lewis, *Infinite Time Turing Machines*, this Journal, vol. 65 (2000), vol. 2, pp. 567–604.

[6] M. Carl, T. Fischbach, P. Koepke, R. Miller, M. Nasfi, and G. Weckbecker, *The basic theory of infinite time register machines*. ***Archive for Mathematical Logic***, vol. 49 (2010), no. 2, pp. 249–273.

[7] R. Jensen and C. Karp, ***Primitive Recursive Set Functions***, Proceedings of Symposia in Pure Mathematics, vol. XIII, Part 1 (1971), pp. 143–176.

[8] P. Koepke, *Ordinal Computability*, ***Mathematical Theory and Computational Practice*** (K. Ambos-Spies et al, editors), Lecture Notes in Computer Science, vol. 5635 (2009), pp. 280–289, pp. 143-176

[9] P. Koepke and Russell Miller, *An enhanced theory of infinite time register machines*, ***Logic and Theory of Algorithms*** (A. Beckmann et al, editors), Lecture Notes in Computer Science, vol. 5028 (2008), pp. 306–315.

[10] K. Kunen, ***Set Theory - An Introduction to Independence Proofs***, Elsevier, Amsterdam, 1980.

[11] A.R.D. Mathias, *Provident sets and rudimentary set forcing*, ***Fundamenta Mathematicae,*** to appear.

[12] G. Sacks, *Countable admissible ordinals and hyperdegrees*. ***Advances in Mathematics***, vol. 19 (1976), pp. 213–262.

FACHBEREICH MATHEMATIK UND STATISTIK
UNIVERSITÄT KONSTANZ
78457 KONSTANZ
GERMANY
*E-mail*: merlin.carl@uni-konstanz.de