The Complexity of Quickly ORM-Decidable Sets

Joel David Hamkins[1], David Linetsky[2], and Russell Miller[3]

[1] The College of Staten Island of CUNY and
The CUNY Graduate Center
jdh@hamkins.org
http://jdh.hamkins.org
[2] The CUNY Graduate Center
365 Fifth Avenue, New York NY 10016
dlinetsky@gc.cuny.edu
https://wfs.gc.cuny.edu/DLinetsky/www/
[3] Queens College of CUNY and
The CUNY Graduate Center
Russell.Miller@qc.cuny.edu
http://qcpages.qc.cuny.edu/math/faculty/miller.htm

Abstract. The Ordinal Register Machine (ORM) is one of several different machine models for infinitary computability. We classify, by complexity, the sets that can be decided quickly by ORMs. In particular, we show that the arithmetical sets are exactly those sets that can be decided by ORMs in times uniformly less than ω^ω. Further, we show that the hyperarithmetical sets are exactly those sets that can be decided by an ORM in time uniformly less than ω_1^{CK}.

Key words: Ordinal, ordinal computation, infinite time computation, computability, register machine, arithmetical hierarchy, hyperarithmetical hierarchy, complexity

1 Introduction

The Ordinal Register Machine (ORM) is one of several different machine models for infinitary computability that can be found in the literature. They are a direct generalization of classical register register machines that differ from their classical counterparts in that they are permitted to contain arbitrary ordinal values in their registers and to run for ordinal time. At limit times, the program state of an ORM is determined by taking a limit inferior (liminf) of the previous states and the content of a register is defined to be the liminf of the values that appeared in it at all previous times. ORMs have been shown to be extremely powerful; the sets of ordinals that they can compute from finitely many ordinal parameters have been characterized by Koepke ([5]) as precisely the constructible sets of ordinals, that is, those found in the Gödel constructible universe, L.

We noticed a curious difference between ORMs and infinite time Turing machines, however, much lower down in the hierarchy, in terms of the lengths of

time that these machines take to decide arithmetic truth. Specifically, the natural recursive algorithm for deciding arithmetic truth with ORMs takes time unbounded in ω^ω, while with infinite time Turing machines this can be done before ω^2 (see [2]). For infinite time Turing machines, each limit allows one to decide two additional quantifiers, with Σ^0_{2n} truth being decidable in time ωn, and one pushes already into the hyperarithmetical hierarchy at time ω^2. But with ORMs, although the first limit allows one essentially to decide three quantifiers, one needs infinitely many additional limits (time ω^2) to go even just a little bit further and decide the Δ^0_4 sets. In general with ORMs, we prove that in time ω^{n+1}, one decides exactly the Δ^0_{3n+1} sets. It follows that ORMs really do need this extra time to decide arithmetic truth in comparison with infinite time Turing machines. We conclude the paper by proving that for hyperarithmetical truth, however, ORMs require time up to ω_1^{ck}, just as for infinite time Turing machines.

2 Arithmetical Sets

Given their demonstrated power, it is certainly clear that ORMs can decide any arithmetical set. However, it is not immediately clear how long such computations actually require. To begin with, we take the case of some relatively simple arithmetic sets and provide a general description of the algorithm used to decide membership in them.

Lemma 1. *Every Σ^0_3 and Π^0_3 subset of ω can be decided by an ORM in $\omega + 2$ steps.*

Proof. Let $A = \{t \mid \exists x \forall y \exists z R(t, x, y, z)\}$, where R is finite time decidable. We describe a program to determine whether $t \in A$.

Start with t in R_1 (register 1) and $0 \in R_2$ (register 2). The program consists of a large loop, each step of which consists of the following:

- For the current element $n \in R_2$, decode $n = \langle x, y, z\rangle$, check whether $R(t, x, y, z)$ and
$$(\forall y' < y)(\exists z')[\langle x, y', z'\rangle < n \;\&\; R(t, x, y', z')]\,, \tag{1}$$
and
$$(\forall z')[\langle x, y, z'\rangle < n \to \neg R(t, x, y, z')]\,. \tag{2}$$
- If all three of these hold, then we have evidence at step n that $(\forall y' \leq y)(\exists z')R(t, x, y, z')$ for a larger y the we had found before this step. In this case, copy x into register R_0.
- If not, then increment R_0 by 1.
- Finally, increment register R_2 by 1 and start over.

Notice that all of these tasks can be accomplished in finite time while using finitely many registers by making use of a program which computes R as a subroutine.

We run this loop ω many times. At stage ω, we compare the values in R_0 and R_2. If they are equal, then we zero the output register and halt. Otherwise, we write 1 in the output register and halt. Clearly, at the limit stage R_2 will contain ω. If R_2 and R_0 are equal at this point, then no x was was copied into register R_0 more than finitely often. Hence, $\forall x \exists y \forall z \neg R(t, x, y, z)$, i.e., $t \notin A$ and 0 is written in the output register. On the other hand, if R_0 does not contain ω, then it has some finite value x_0. This can only be the case if infinitely often more evidence was found that x_0 was a witness to the property $\forall y \exists z\ R(t, x, y, z)$. It immediately follows that $\exists x \forall y \exists z\, R(t, x, y, z)$, and hence we have that $t \in A$ and we output 1.

To decide membership in a Π^0_3 set, simply reverse the outputs. □

Using the algorithm described above, we can push on a little further into the arithmetical hierarchy:

Lemma 2. *Every Δ^0_4 set can be decided in time less than ω^2.*

Proof. Let $A \in \Delta^0_4$. Then A and $\omega \setminus A$ have Σ_4 definitions:

$$A = \{t \mid \exists x\ R(t, x)\} \tag{3}$$
$$\omega \setminus A = \{t \mid \exists x\ R'(t, x)\} \tag{4}$$

where $R, R' \in \Pi^0_3$. To decide if $t \in A$, we begin searching for an x so that either $R(t, x)$ or $R'(t, x)$. By Lemma 1, we can decide R, R' in $\omega + 2$ steps. Thus, we can decide whether t lies in A in $\omega \cdot n$ steps, for some $n \in \omega$. □

These initial results are easily extended to cover all arithmetical sets. This next lemma give the easy direction of the main theorem to come.

Lemma 3. *Every arithmetic set is ORM-decidable in time uniformly less than ω^ω. Indeed, if $A \in \Delta^0_{3n+1}$, then A is ORM-decidable in time less than ω^{n+1}.*

Proof. We proceed by induction on n. The case $n = 0$ is clear; it simply asserts the classical fact that Δ^0_1 sets are finite time decidable. In fact, the previous lemma gives the case $n = 1$ as well. Now, suppose the result holds below n and let $A \in \Delta^0_{3n+1}$. Then A and $\omega \setminus A$ have Σ_{3n+1} definitions:

$$A = \{t \mid \exists x\ R(t, x)\} \tag{5}$$
$$\omega \setminus A = \{t \mid \exists x\ R'(t, x)\} \tag{6}$$

where $R, R' \in \Pi^0_{3n}$. Then, $R = \{t \mid \forall x \exists y \forall z\ Q(t, x, y, z)\}$, where $Q \in \Sigma^0_{3n-3} \subseteq \Delta^0_{3n-2}$. By the inductive hypothesis, Q is ORM-decidable in time less than ω^n. So, we simply apply the algorithms described in Lemmas 1 and 2, using the program that decides Q as a subroutine. In this manner, we can decide A in time less than ω^{n+1}. □

3 Characterizing the Quickly Decidable Sets

We now prove the harder direction, that in time uniformly less than ω^ω, ORMs do not escape the arithmetical hierarchy. The Main Theorem provides a characterization of the sets decidable by ORMs in times strictly less than ω^ω.

Theorem 1. *The subsets of ω that are ORM-decidable (using arbitrary ordinal parameters!) in time uniformly less than ω^ω are exactly the arithmetical sets. In particular, a set A is ORM-decidable in time less than ω^{n+1} if and only if $A \in \Delta^0_{3n+1}$.*

This theorem should be contrasted with its analogue for Infinite Time Turing Machines (ITTMs) found in [2], which states that every arithmetic set can be decided by an ITTM in time uniformly less than ω^2. Of course, ITTMs are able to do this by making use of their infinite tape memory on which they are able to write out oracles that can be later referred to in order to greatly speed up later computations. ORMs, on the other hand, can store only finitely many ordinals and are thus unable to use this type of strategy. Instead, they must recompute the information required to preform a computation each time it is required.

Before we prove this main result, we provide some definitions and develop some of the key ideas involved. In order to analyze the descriptive complexity of a set that is decidable in time less than ω^n, for some $n \in \omega$, we need to be able to talk about ORM configurations in a first order fashion. Since the register contents of an ORM are arbitrary ordinals, in order to accomplish this, we require a method of coding these ordinals as natural numbers.

Definition 1. *For any ordinal $\alpha < \omega^\omega \cdot 2$, let $\ulcorner \alpha \urcorner = \langle m, n_0, \ldots, n_k \rangle$ where $m \in \{0, 1\}$ and $\alpha = \omega^\omega \cdot m + \omega^k \cdot n_k + \cdots + \omega \cdot n_1 + n_0 < \omega^\omega$. Furthermore, let $\prec$ be the order on these codes such that $\ulcorner \alpha \urcorner \prec \ulcorner \beta \urcorner$ if and only if $\alpha < \beta$.*

Now, since ORM programs are finite and can make use of only finitely many registers, we may also code an ORM configuration as a natural number.

Definition 2. *An* ORM configuration *consists of a program state and the contents of each register. In the case that a computation uses only ordinals less than $\omega^\omega \cdot 2$, then we use the above coding to code each of the its configurations as some natural number $\mathcal{C} \in \omega$.*

The statement of Theorem 1 allowed for arbitrary ordinal parameters while Definition 1 only allows us to code ordinals smaller than $\omega^\omega \cdot 2$. The next result shows that this is in fact sufficient.

Lemma 4. *Given any ORM program P, finite sequence of ordinals $\vec{\beta} < \omega^\omega$, and any ordinal parameter $\beta > \omega^\omega$, $P(\vec{\beta}, \beta) \downarrow = \gamma < \omega^\omega$ in time less than ω^ω if and only if $P(\vec{\beta}, \omega^\omega) \downarrow = \gamma$, and they do so in exactly the same number of steps (where $P(\vec{\alpha}) \downarrow$ means that program P converges on inputs $\vec{\alpha}$).*

Proof. The idea of the proof is that the ordinal β is much too large for the ORM algorithm to make use of in such a short time; any such β operates identically to ω^ω itself in any computation. Specifically, we prove the result by induction on time. On one ORM, $\mathcal{M}$, we run $P(\overrightarrow{\beta}, \beta)$, while on a second ORM, $\mathcal{M}'$, we run $P(\overrightarrow{\beta}, \omega^\omega)$. It is not hard to see that at every time t, the machine state of $\mathcal{M}$ is the same as that of $\mathcal{M}'$, and if any register in $\mathcal{M}$ contains a value $< \omega^\omega$, then the corresponding register of $\mathcal{M}'$ contains the same value. Moreover, any register of $\mathcal{M}$ has value $\beta + \gamma$, with $\beta > \omega^\omega$, if and only if the corresponding register in $\mathcal{M}'$ has value $\omega^\omega + \gamma$. At limit times we use the fact that for any $\alpha < \omega^\omega$, $\alpha + \omega^\omega = \omega^\omega$. □

In fact, the above proof goes through if every occurrence of ω^ω in the statement of the lemma is replaced by any ordinal of the form ω^α. Now, given the method of Definition 2 for coding machine configurations, we now define a relation that will allow us to describe how two configurations relate to each other relative to a particular program.

Definition 3. *Let $\mathcal{C}, \mathcal{C}' \in \omega$ be codes for two ORM configurations, $\alpha < \omega^\omega$ and let $P \in \omega$ be the code for an ORM-program. Define relation $R \subseteq \omega^4$ by: $R(\mathcal{C}, \mathcal{C}', P, \ulcorner \alpha \urcorner)$ if and only if the configuration coded by $\mathcal{C}'$ follows the configuration coded by $\mathcal{C}$ in exactly α steps under the operation of the program coded by P. Also, for any ordinal $\alpha < \omega^\omega$, define the relation $R_\alpha \subseteq \omega^3$ by $R_\alpha(\mathcal{C}, \mathcal{C}', P) \leftrightarrow R(\mathcal{C}, \mathcal{C}', P, \ulcorner \alpha \urcorner)$.*

Of course, in order to make use of this relation, we need to know that it can be expressed in a first order fashion. This is taken care of by the next result.

Lemma 5. *For each ordinal $\alpha < \omega^\omega$, the relation $R_\alpha(\mathcal{C}, \mathcal{C}', P)$ is arithmetical. Indeed, if $\alpha = \omega^k \cdot n_k + \cdots + \omega \cdot n_1 + n_0$, then the statement $R_\alpha(\mathcal{C}, \mathcal{C}', P)$ is Δ^0_{3k+1}.*

Proof. By induction on k where $\alpha = \omega^k \cdot n_k + \cdots + \omega \cdot n_1 + n_0$. Clearly R_1 is finite time computable, and hence Δ^0_1. Suppose the result holds below k and that $\alpha = \omega^k \cdot n_k + \cdots + \omega \cdot n_1 + n_0$. Clearly, $R_\alpha(\mathcal{C}, \mathcal{C}', P)$ holds if and only if there is a configuration $\mathcal{C}''$ such that $R_{\omega^k \cdot n_k}(\mathcal{C}, \mathcal{C}'', P)$ and $R_\beta(\mathcal{C}'', \mathcal{C}', P)$, where $\beta = \omega^{k-1} \cdot n_{k-1} + \cdots + \omega \cdot n_1 + n_0$. The latter relation is arithmetical by assumption, so we need only consider $R_{\omega^k \cdot n_k}(\mathcal{C}, \mathcal{C}'', P)$.

Now, if $n_k > 1$, then $R_{\omega^k \cdot n_k}(\mathcal{C}, \mathcal{C}'', P)$ is equivalent to asserting that there exist configuration $\mathcal{D}_0, \ldots, \mathcal{D}_{n_k}$ such that $\mathcal{C} = \mathcal{D}_0, \mathcal{C}'' = \mathcal{D}_{n_k}$, and that $R_{\omega^k}(\mathcal{D}_i, \mathcal{D}_{i+1}, P)$ for $i < n_k$, i.e., each configuration follows the previous one in ω^k steps. Thus, it suffices to consider the case $n_k = 1$, which asserts that $\mathcal{C}$ leads to $\mathcal{C}''$ in ω^k steps. Using the relation for smaller ordinals, and our coding of smaller ordinals as natural numbers, we can simply write out the liminf definition as a first order sentence and see that it is indeed arithmetical.

Indeed, by the remarks above, it suffices to show that $R_{\omega^k}(\mathcal{C}, \mathcal{C}', P) \in \Pi^0_{3k}$, from which it follow that $R_\alpha(\mathcal{C}, \mathcal{C}', P) \in \Delta^0_{3k+1}$, where $\alpha = \omega^k \cdot n_k + \cdots + \omega \cdot n_1 \cdot n_0$. To see that this is so, we give the portion of the liminf definition which bounds

its complexity, that is, we look closely at how to say that the value of the n^{th} register, $\mathcal{C}'(n)$, is equal to some ordinal ξ.

$$\begin{aligned}\mathcal{C}(n) = \xi \text{ iff } & (\forall^{\ulcorner}\beta^{\urcorner} \prec {\ulcorner}\xi^{\urcorner})(\exists^{\ulcorner}\gamma^{\urcorner} \prec {\ulcorner}\omega^k{\urcorner})(\forall\mathcal{D})(\forall^{\ulcorner}\delta^{\urcorner} \succ {\ulcorner}\gamma^{\urcorner}) \\ & \qquad \big[\big({\ulcorner}\delta^{\urcorner} \prec {\ulcorner}\omega^k{\urcorner} \;\&\; R(\mathcal{C},\mathcal{D},P,{\ulcorner}\delta^{\urcorner})\big) \rightarrow {\ulcorner}\mathcal{D}(i)^{\urcorner} \succ {\ulcorner}\beta^{\urcorner}\big] \\ & \;\&\; (\forall^{\ulcorner}\beta^{\urcorner} \prec {\ulcorner}\omega^k{\urcorner})(\exists\mathcal{D})(\exists^{\ulcorner}\gamma^{\urcorner} \succ {\ulcorner}\beta^{\urcorner}) \\ & \qquad [R(\mathcal{C},\mathcal{D},P,{\ulcorner}\gamma^{\urcorner}) \;\&\; {\ulcorner}\mathcal{D}(i)^{\urcorner} \preceq {\ulcorner}\xi^{\urcorner}]\end{aligned} \tag{7}$$

Assuming inductively that $R(\mathcal{C},\mathcal{D},P,{\ulcorner}\gamma^{\urcorner}) \in \Delta^0_{3k-2}$, for all $\gamma < \alpha$, it follows that the above sentence is indeed Π^0_{3k}, and hence that $R_\alpha(\mathcal{C},\mathcal{C}',P) \in \Delta^0_{3k+1}$. □

Putting together all of these pieces, we can now go ahead and prove the main result of this section.

Proof (of Theorem 1). Suppose A is decidable in time less than ω^{n+1} (by program P, say). Then, $x \in A$ if and only if

$$(\exists\mathcal{C})(\exists n)\ [R_\alpha(x^*,\mathcal{C},P) \;\&\; n = {\ulcorner}\alpha^{\urcorner} \;\&\; \mathcal{C} \text{ halts with output 1}] \tag{8}$$

if and only if

$$(\forall\mathcal{C})(\forall n)\ [(R_\alpha(x^*,\mathcal{C},P) \;\&\; n = {\ulcorner}\alpha^{\urcorner}) \rightarrow \mathcal{C} \text{ halts with output 1}]\ , \tag{9}$$

where x^* is the start configuration with x in the input register. Hence, since $R_\alpha(x^*,\mathcal{C},P)$ is Δ^0_{3n+1}, it follows that $A \in \Delta^0_{3n+1}$. □

4 Deciding Hyperarithmetical Sets

Of course, ORMs are capable of deciding membership in sets much more complex than the arithmetical sets. In this final section we present two results concerning hyperarithmetical sets. The first shows that the uniformity in time found in Theorem 1 is necessary, i.e., there are hyperarithmetical sets that can be decided in time less than ω^ω (but not uniformly so). The second result gives a characterization of the sets decidable in times uniformly less than ω_1^{CK}, the first non-recursive ordinal, as precisely the hyperarithmetical sets.

Theorem 2. *There exist hyperarithmetic sets that are ORM-decidable in time less than ω^ω (not uniformly). Indeed, the ω^{th} jump of zero, $\emptyset^{(\omega)}$, is such a set.*

Proof. We show that $\emptyset^{(\omega)}$ can be computed in time less than ω^ω by describing an algorithm that accomplishes this. Of course, Theorem 1 ensures that this can't be done in time uniformly less than ω^ω. The algorithm requires that we simulate a stack machine (as in [5] and [3]) with two stacks on an ORM. To decide whether $(n,k) \in \emptyset^{(\omega)}$, i.e., whether $k \in \emptyset^{(n)}$ we proceed as follows: First, we run the algorithm which computes whether $k \in \emptyset^{(n)}$ using an $\emptyset^{(n-1)}$-oracle. When the algorithm queries the oracle as to whether some natural number $k' \in \emptyset^{(n-1)}$, push $n-1$ onto stack 1, and push $\omega\cdot(n-1)+k'$ onto stack 2 and run this same program

over again. We push $\omega \cdot (n-1) + k'$ instead of just k' because when deciding whether $k' \in \emptyset^{n-1}$ we may have to make a query about some number $l > k'$. However, pushing l onto stack 2 would violate the stack protocol, which requires that any number pushed onto a non-empty stack must be smaller than the number preceding it. This recursive process will eventually decide membership in $k \in \emptyset^{(\omega)}$. Moreover, an analysis of the algorithm shows that it always halts in time less than ω^ω, but that for every $m \in \omega$ there is some $\langle n, k\rangle \in \omega$ such that the computation that decides whether $\langle n, k\rangle \in \emptyset^{(\omega)}$ takes more than ω^m steps. □

Finally we characterize the sets decidable by an ORM in time less than some recursive ordinal.

Theorem 3. *The sets that are ORM-decidable in time less than ω_1^{CK} are exactly the hyperarithmetic sets.*

We prove this result using a sequence of lemmas. To begin with, we will show that every hyperarithmetical set is decidable in time less than some recursive ordinal. In order to this, we use a result of Shoenfield (see [9] or exercise 16-93 in [7]), which states that every hyperarithmetical set is Turing reducible to some element of a particular series of sets. The sets are defined using the usual jump operator; we denote the jump of a set D by D'.

Lemma 6 (Shoenfield). *Define a sequence of sets D_α, for $0 < \alpha < \omega_1^{CK}$, as follows:*

$$\begin{aligned} D_0 &= \omega\,; \\ D_{\alpha+1} &= D_\alpha \cap (D_\alpha)'\,; \\ D_\alpha &= \bigcap_{\beta<\alpha} D_\beta\,, \quad \text{if } \alpha \text{ is a limit ordinal.} \end{aligned} \tag{10}$$

Then, $A \in \Delta_1^1$ if and only if $A \leq_T D_\alpha$ for some $\alpha < \omega_1^{CK}$.

Of course, in order for Lemma 6 to be useful, we need to know that each of the D_α's is ORM-decidable in time less than some recursive ordinal, which we take care of in the next result.

Lemma 7. *For every $\alpha < \omega_1^{CK}$, D_α is ORM-decidable in time less than some $\beta < \omega_1^{CK}$.*

Proof. We proceed by induction on α. Suppose that D_α is ORM-decidable in time less than β, where $\beta < \omega_1^{\mathrm{CK}}$. For any $x \in \omega$, to determine whether $x \in D_{\alpha+1}$, we need to determine whether $x \in D_\alpha$ (requiring time $< \beta$) and whether $x \in (D_\alpha)'$, which may require up to ω queries about D_α and possibly one more step to determine that $x \notin (D_\alpha)'$. Thus, $D_{\alpha+1}$ can be decided in time at most $\beta \cdot \omega + 1$, which is again a recursive ordinal since ω_1^{CK} is closed under ordinal arithmetic.

Now suppose that α is a limit and the result holds for all $\gamma < \alpha$. Define the function $f_\alpha : \alpha \to \omega_1^{\mathrm{CK}}$ by $f(\gamma) = \delta$ if and only if δ is least such that D_γ is ORM-decidable in time less than δ. Now, if we want to decide whether some $x \in D_\alpha$, we need to decide whether it is in D_γ for each $\gamma < \alpha$. Thus, D_α should be decidable in time at most $\epsilon := \bigcup_{\gamma<\alpha} f_\alpha(\gamma)$. Thus, if can show that $\epsilon < \omega_1^{\mathrm{CK}}$, we are done.

We note that $D_\alpha \in L_{\omega_1^{\mathrm{CK}}}$ for each $\alpha < \omega_1^{\mathrm{CK}}$, and that the notion of ORM computability can be defined in a Δ_0 way in $L_{\omega_1^{\mathrm{CK}}}$. Thus, the graph of f_α is Δ_0 definable in $L_{\omega_1^{\mathrm{CK}}}$. So, since ω_1^{CK} is an admissible ordinal, and thus satisfies Σ_1 replacement, it follows that $\epsilon = \operatorname{ran} f_\alpha \in L_{\omega_1^{\mathrm{CK}}}$, and hence that $\epsilon < \omega_1^{\mathrm{CK}}$. □

Lemmas 6 and 7 essentially complete the proof of the backwards direction of Theorem 3. To see this, suppose that A is hyperarithmetical, i.e., that $A \in \Delta_1^1$. Then, by Lemma 6, we have that $A \leq_T D_\alpha$ for some $\alpha < \omega_1^{\mathrm{CK}}$. By Lemma 7, we can decide D_α is time less than some $\beta < \omega_1^{\mathrm{CK}}$. Thus, it follows that we can decide A in time at most $\beta \cdot \omega < \omega_1^{\mathrm{CK}}$. Hence, every hyperarithmetical set can be decided by an ORM in time less than some recursive ordinal.

All that remains now is to show the converse. In order to this, we will make use of Kleeene's $\mathcal{O}$, which provides natural number notations for every recursive ordinal. We will not define $\mathcal{O}$, but the standard definitions and results concerning $\mathcal{O}$ may be found in any standard text on the subject. We will follow the notation found in [1]. Thus, for any $n \in \mathcal{O}$, we write $|n|_{\mathcal{O}}$ to denote the ordinal denoted by n and we denote the standard order relation on $\mathcal{O}$ by $<_{\mathcal{O}}$ so that $a <_{\mathcal{O}} b$ if and only if $|a|_{\mathcal{O}} < |b|_{\mathcal{O}}$. We now complete the proof of Theorem 3 by proving the following lemma.

Lemma 8. *If $A \subseteq \omega$ is ORM-decidable in time less than some recursive ordinal, then $A \in \Delta_1^1$.*

Proof. Suppose that $A \subseteq \omega$ is ORM-decidable, by the program P, in time less than α for $\alpha < \omega_1^{\mathrm{CK}}$. Fix some $a \in \mathcal{O}$ so that $|a|_{\mathcal{O}} = \alpha$. Then, the set $S := \{b \mid b <_{\mathcal{O}} a\} \subseteq \omega$ is recursively enumerable and provides a set of notations for all ordinals less than α. Using S, we can mimic Definition 2 and code any ORM configuration having register contents less than α as natural numbers. For any $\mathcal{C}, \mathcal{C}' \in \omega$ coding two such configurations, let us say that $Q^P(\mathcal{C}, \mathcal{C}', n)$ holds if and only if the configuration coded by $\mathcal{C}'$ follows that coded by $\mathcal{C}$, under the operation of the program P, in exactly $|n|_{\mathcal{O}}$ many steps.

We claim that for any $n \in S$ the relation $Q^P(\mathcal{C}, \mathcal{C}', n) \in \Delta_1^1$. This is easily shown by induction. Let $n \in S$ and suppose that the result holds for all $m <_{\mathcal{O}} n$. If $|n|_{\mathcal{O}} = \beta + 1$, then $n = 2^m$ for $m \in S$, $m <_{\mathcal{O}} n$, and $|m|_{\mathcal{O}} = \beta$. In this case, $Q^P(\mathcal{C}, \mathcal{C}', n) \in \Delta_1^1$ if and only if $(\exists \mathcal{D})[Q^P(\mathcal{C}, \mathcal{D}, m) \,\&\, Q^P(\mathcal{C}, \mathcal{D}, 2)]$ (note: $|2|_{\mathcal{O}}$=1). Thus, by the induction hypothesis, it follows that $Q^P(\mathcal{C}, \mathcal{D}, n) \in \Delta_1^1$. On the other hand, if $|n|_{\mathcal{O}}$ is a limit ordinal, then we must unravel the lim inf definition as we did in the proof of Lemma 5, except that we use codes from S and the order $<_{\mathcal{O}}$ instead of $\prec$. To do this, we require only quantifiers over ω, which do not increase the complexity. Hence, the relation $Q^P(\mathcal{C}, \mathcal{C}', n) \in \Delta_1^1$ for every $n <_{\mathcal{O}} a$.

The theorem now follows immediately, since $x \in A$ if and only if

$$(\exists \mathcal{C})(\exists n \in S)\ \left[Q^P(x^*, \mathcal{C}, n)\ \&\ \ \mathcal{C} \text{ halts with output } 1\right] \tag{11}$$

if and only if

$$(\forall \mathcal{C})(\forall n \in S)\ \left[Q^P(x^*, \mathcal{C}, n) \rightarrow\ \mathcal{C} \text{ halts with output } 1\right]\ , \tag{12}$$

where x^* is the start configuration with x in the input register, and all other registers set to zero. Hence, $A \in \Delta^1_1$ and is thus hyperarithmetical. □

References

1. Ashe, C. J. and J. Knight. *Computable Structures and the Hyperarithmetical Hierarchy*. Studies in Logic and the Foundations of Mathematics. Elsevier, 2000.
2. Hamkins, Joel David and Andy Lewis. Infinite Time Turing Machines. *J. Symbolic Logic*, 65(2):567-604, 2000.
3. Hamkins, Joel David and Russell Miller. Post's Problem for Ordinal Register Machines. To appear in this volume.
4. Jech, Thomas. *Set Theory. The Third Millenium Edition*. Springer Monographs in Mathematics. Springer-Verlag, 2003.
5. Koepke, Peter. Ordinals, Computations, and Models of Set Theory: A Tutorial at Days in Logic, Coimbra, Portugal. Tutorial Material. `http://www.mat.uc.pt/~kahle/dl06/koepke.pdf`; accessed January, 2006.
6. Koepke, Peter. Turing Computations on Ordinals. *J. Symbolic Logic*, 11(3):377-397, 2005.
7. Rogers, Hartley, Jr. *Theory of Recursive Functions and Effective Computability*. The MIT Press, 1967.
8. Sacks, G. E. *Higher Recursion Theory*. Perspectives in Mathematical Logic. Spring-Verlag, 1990.
9. Shoenfield, Joseph R. *Recursion Theory*. Spring Lecture Notes in Logic. Spring-Verlag, 1993.