FEEDBACK COMPUTABILITY ON CANTOR SPACE

NATHANAEL L. ACKERMAN, CAMERON E. FREER, AND ROBERT S. LUBARSKY

Abstract. We introduce the notion of feedback computable functions from 2^ω to 2^ω, extending feedback Turing computation in analogy with the standard notion of computability for functions from 2^ω to 2^ω. We then show that the feedback computable functions are precisely the effectively Borel functions. With this as motivation we define the notion of a feedback computable function on a structure, independent of any coding of the structure as a real. We show that this notion is absolute, and as an example characterize those functions that are computable from a Gandy ordinal with some finite subset distinguished.

keywords: feedback computability, Borel functions, computable model theory

AMS MSC 2010: Primary 03D65; Secondary 03C57, 03D30, 03E15

1. Feedback machines and Borel maps

One of the most important observations of (effective) descriptive set theory is that every continuous map between Polish spaces is computable with respect to some oracle, and every map which is computable with respect to some oracle is continuous. (See [Mos09, Ex. 3D.21].)

This fact allows one to transport results from computability theory to the theory of continuous functions, and vice versa. But, it also is important because it provides a machine model for continuity. Specifically, it provides a way of thinking about a continuous map on a Polish space as a construction of the output

from the input, instead of simply via the abstract definition (requiring the inverse image of open sets to be open).

We will show that this correspondence extends to feedback Turing computation [AFL15] and Borel functions. That is, we will show that feedback Turing computability provides a machine model for Borel functions on 2^ω, and conversely that every feedback computable function is itself Borel. These results should not be surprising, as it is already known (as reviewed below) that feedback computable reals are exactly the hyperarithmetic (i.e., Δ^1_1) reals, and it is an old result of descriptive set theory that the Borel sets are exactly the Δ^1_1-definable sets. So what we are doing here could be viewed as merely a type shift — from Δ^1_1 reals, which are Δ^1_1 properties of natural numbers, to Δ^1_1 properties on reals (for sets, or on pairs of reals if thinking about the graph of a function).

1.1. **Notation.** It will be useful to let $\tau\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ be a computable bijection. We will also need a computable bijection $\iota\colon \mathbb{N}\to 2^{<\omega}$. For $\sigma\in 2^{<\omega}$, define $\widehat{\sigma}:=\{x\in 2^\omega : \sigma\prec x\}$, i.e., the collection of elements of 2^ω extending σ.

For a set A, define $\mathcal{P}_{<\omega}(A)$ to be the finite powerset of A, i.e., the set of all finite subsets of A.

By a **countable ordinal** we will mean a well-founded linear order $(A,\lhd)$ such that $A\subseteq\omega$. In particular, this representation of countable ordinals will make it possible for oracle machines, as well as feedback machines, to access them as oracles.

Given an admissible ordinal α, let α^+ be the **next admissible after** α, i.e., the least admissible ordinal greater than α. An ordinal α is defined to be a **Gandy ordinal** [AS76] if it is admissible and for all $\gamma<\alpha^+$ there is an α-computable well-ordering of order type γ.

1.2. **Feedback computability.** We now review the notion of feedback computability studied in [AFL15]. The intuitive idea is that we want to make sense of the notion of a machine which can ask halting queries of *machines of the same type*.

The notation $\{e\}^X_F(n)$ denotes the eth Turing machine with oracle X and halting function F (which can also be interpreted as an oracle) on input n. When $\{e\}^X_F(n)$ queries X it is said to be making an **oracle query**, and when it queries F it is said to be making a **halting query**.

Definition 1. *For any $X\colon\omega\to\{0,1\}$ define the set $H_X\subseteq\omega\times\omega$ to be the smallest collection for which there is a function $h_X\colon H_X\to\{\uparrow,\downarrow\}$ satisfying the following:*

- ($\downarrow$) *If $\{e\}^X_{h_X}(n)$ makes no halting queries outside of H_X and converges after a finite number of steps then $(e,n)\in H_X$ and $h_X(e,n)=\downarrow$, and conversely.*
- ($\uparrow$) *If $\{e\}^X_{h_X}(n)$ makes no halting queries outside of H_X and does not converge (i.e., runs forever) then $(e,n)\in H_X$ and $h_X(e,n)=\uparrow$, and conversely.*

Furthermore, this h_X is unique.

Definition 2. *A* **feedback Turing machine** *(or* **feedback machine** *for short) is a machine of the form $\{e\}^X_{h_X}$ for some $e \in \omega$. The notation $\langle e\rangle^X(n)$ is shorthand for $\{e\}^X_{h_X}(n)$.*

The set H_X is the collection of **non-freezing** *computations and the notation $\langle e\rangle^X(n) \Downarrow$ means $(e,n) \in H_X$. If $(e,n) \notin H_X$ then $\langle e\rangle^X(n)$ is* **freezing***, written $\langle e\rangle^X(n) \Uparrow$.*

One of the most important results about feedback computability is that feedback reducibility is equivalent to Δ^1_1- or hyperarithmetic reducibility. A set X is hyperarithmetically reducible to Y if X is in the least admissible set containing Y.

Theorem 3 ([AFL15, Theorem 16])**.** *For any $X, Y : \omega \to \{0,1\}$ the following are equivalent.*

- *X is $\Delta^1_1(Y)$.*
- *There is a feedback machine e such that $\langle e\rangle^Y(n) = X(n)$ for all $n \in \mathbb{N}$.*

Having recalled the notion of feedback computability, we now introduce the notion of a feedback computable function from 2^ω to 2^ω.

Definition 4. *For $X \subseteq \mathbb{N}$, a map $f : 2^\omega \to 2^\omega$ is* **feedback computable with respect to** *X if there is a feedback machine with code e such that for all $Y \in 2^\omega$, the function $\langle e\rangle^{X,Y}$ is total and for all $n \in \omega$, $\langle e\rangle^{X,Y}(n) = f(Y)(n)$. In this case e is said to* **code** *(with respect to X) a feedback computable function from 2^ω to 2^ω.*

In other words, a function is feedback computable if there is a feedback machine which, when given a description of a point in the domain, outputs a description of the image of the point in the range.

1.3. **Borel codes.** A *Borel code* of a Borel subset of 2^ω captures the way in which the Borel set was built up from basic open sets using the operations of countable union and complementation. There are many different types of Borel codes, all of which are, for practical purposes, equivalent. However, for our purposes it will be convenient to give a concrete coding system where each Borel code is a an element of $\mathbb{N}^\mathbb{N}$.

We begin with a couple of basic operations on functions from $\mathbb{N}$ to $\mathbb{N}$. Suppose $f : \mathbb{N} \to \mathbb{N}$. Let $f_* : \mathbb{N} \to \mathbb{N}$ be such that $f_*(n) := f(n+1)$ for all $n \in \mathbb{N}$. Also, for $m \in \mathbb{N}$, let $f_m : \mathbb{N} \to \mathbb{N}$ be such that $f_m(n) := f(\tau(m,n)+1)$ for all $n \in \mathbb{N}$.

Definition 5. *Let $\mathrm{BC} \subseteq \mathbb{N}^\mathbb{N}$ be the collection of* **Borel codes** *(for 2^ω), defined by induction as follows.*

- $\mathrm{BC}_0 := \{f : f(0) \geq 2\}$.
- *If $\beta \in \mathrm{ORD}$ is greater than 0 then BC_β is the smallest set containing*
  - *$B_{\beta^*} \cup \{f : f(0) = 0 \wedge f_* \in \mathrm{BC}_{\beta^*}\}$ for $\beta^* < \beta$, and*

- *all functions* $f\colon \mathbb{N} \to \mathbb{N}$ *such that* $f(0) = 1$ *and for all* $m \in \mathbb{N}$ *there is a* $\beta_n < \beta$ *such that* $f_m \in B_{\beta_n}$.

Finally, set $\mathrm{BC} := \bigcup_{\alpha<\omega_1} \mathrm{BC}_\alpha$. *If* $f \in \mathrm{BC}$ *then define the* **rank** *of* f *to be the least ordinal* α *such that* $f \in \mathrm{BC}_\alpha$.

A Borel code (for 2^ω) has an associated Borel set, called its *realization*.

Definition 6. *Suppose* $\zeta \in \mathbb{N}^\mathbb{N}$ *is a Borel code (for* 2^ω*). Define the* **realization** *of a Borel* ζ*, written* $\boldsymbol{R}(\zeta)$*, to be the Borel subset of* 2^ω *defined by induction on the rank of the* ζ *as follows.*

- *If* $\zeta(0) \geq 2$*, then* $\boldsymbol{R}(\zeta) := \widehat{\sigma}$ *where* $\sigma = \iota(\zeta(0) - 2)$*. Note that in this case,* $\mathrm{rank}(\zeta) = 0$.
- *If* $\zeta(0) = 0$*, then let* $\boldsymbol{R}(\zeta) := 2^\omega \setminus \boldsymbol{R}(\zeta_*)$*. Note that in this case, if* $\mathrm{rank}(\zeta) = \beta + 1$ *for some* $\beta \in \mathrm{ORD}$*, then* $\mathrm{rank}(\zeta_*) = \beta$.
- *If* $\zeta(0) = 1$ *then* $\boldsymbol{R}(\zeta) := \bigcup_{m\in\mathbb{N}} \boldsymbol{R}(\zeta_m)$*. Note that in this case,* $\mathrm{rank}(\zeta_m) < \mathrm{rank}(\zeta)$ *for all* $m \in \mathbb{N}$.

By the *union* or *intersection* of a collection of Borel codes, we will mean the code for union or intersection of their realizations.

One of the first steps in showing that the Borel functions and the feedback computable functions coincide is to show that feedback computability interacts well with Borel codes.

Lemma 7. *There is a feedback machine* **bor** *such that for any* $\alpha \in \omega_1$*, any* $X \in 2^\omega$ *and any* $n \in \mathbb{N}$ *such that* $\langle n\rangle^X$ *is total, we have*

- $\langle \mathbf{bor}\rangle^{X,\alpha}(n) = 1$ *if* $\langle n\rangle^X \in \mathrm{BC}_\alpha$*, and*
- $\langle \mathbf{bor}\rangle^{X,\alpha}(n) = 0$ *if* $\langle n\rangle^X \notin \mathrm{BC}_\alpha$.

Proof. Let $\langle \mathbf{bor}\rangle^{X,\alpha}(n)$ do the following.

Step 1:
If $\langle n\rangle^X(0) \geq 2$ then return 1.

Step 2:
If $\langle n\rangle^X(0) = 0$ then search for a $\beta < \alpha$ such that $\langle \mathbf{bor}\rangle^{X,\beta}(n_*) = 1$ where $\langle n_*\rangle^X(m) := \langle n\rangle^X(m+1)$ for all $m \in \mathbb{N}$. If there exists such a β then return 1, and otherwise return 0.

Step 3:
If $\langle n\rangle^X(0) = 1$ then for each $m \in \mathbb{N}$ search for a $\beta_m < \alpha$ such that $\langle \mathbf{bor}\rangle^{X,\beta}(n_m) = 1$ where for all $m \in \mathbb{N}$, $\langle n_m\rangle^X(k) := \langle n\rangle^X(\tau(m,k)+1)$. If there exists such a β_m for each $m \in \mathbb{N}$ then return 1, and otherwise return 0.

It is an easy induction to show that **bor** is the desired code. □

In other words, there is a feedback machine which, uniformly in α and an oracle X, can check whether or not a total function feedback computable in X is a Borel code of rank at most α.

Lemma 8. *There is a feedback machine* **in** *such that for any Borel code C, any $X \in 2^\omega$ and any $n \in \mathbb{N}$ such that $\langle n \rangle^X$ is total, we have*

- $\langle \mathbf{in} \rangle^{C,X}(n) = 1$ *if* $\langle n \rangle^X \in \boldsymbol{R}(C)$, *and*
- $\langle \mathbf{in} \rangle^{C,X}(n) = 0$ *if* $\langle n \rangle^X \notin \boldsymbol{R}(C)$.

Proof. Call C the first oracle and X the second oracle. We will need to unravel the code C. However, there is no mechanism for changing an oracle. This ends up not being a problem, because the changes we would like to make are simple — which we formalize using the Recursion Theorem.

Toward this end, suppose we have computer code e, which we will think of as instructions for a feedback machine $\langle e \rangle$. We will define new code e^*, as follows. The behavior of the computation $\langle e^* \rangle^{C,X}$ depends on the value of $C(0)$.

Case 1: $C(0) \geq 2$.
Let $\sigma := \iota(C(0) - 2)$. If $\langle n \rangle^X(m) = \sigma(m)$ for all $m \in \mathrm{dom}(\sigma)$, then return 1. Otherwise return 0.

Case 2: $C(0) = 0$.
Let $\langle e \rangle^{C_*,X}$ be the machine that runs just like $\langle e \rangle$ with oracle C, X, except that whenever e makes a query of k to the first oracle, a query of $k+1$ is made instead. Return $1 - \langle e \rangle^{C_*,X}$.

Case 3: $C(0) = 1$.
Let $\langle e \rangle^{C_m,X}$ be the machine that runs just like $\langle e \rangle$ with oracle C, X, except that whenever e makes a query of k to the first oracle, a query of $\tau(m,k)+1$ is made instead. Let $\langle e^* \rangle^{C,X}$ search for an m such that $\langle e \rangle^{C_m,X} = 1$, and if it finds such an m it returns 1, else 0.

The function from e to e^* is computable. Let **in** be a fixed point. It is an easy induction on the rank the Borel code C to show that **in** has the desired properties. □

In other words, there is a feedback machine which can determine whether or not a feedback computable function (relative to an oracle) is in the realization of a Borel code, uniformly in the Borel code and the oracle. Now we show, unsurprisingly, that if given a feedback computable collection of Borel codes, they can be combined to form a new Borel code.

Lemma 9. *There are feedback machines* **neg**, **cup** *such that*

- *if* $\langle n\rangle^X \in \mathrm{BC}$, *then* $\langle \mathbf{neg}\rangle^X(n,\cdot\,) \in \mathrm{BC}$ *with* $\boldsymbol{R}\big(\langle \mathbf{neg}\rangle^X(n,\cdot\,)\big) = 2^\omega \setminus \boldsymbol{R}\big(\langle n\rangle^X\big)$, *and*
- *if* $\langle n\rangle^X(m,\cdot\,) \in \mathrm{BC}$ *for all* $m \in \mathbb{N}$, *then* $\langle \mathbf{cup}\rangle^X(n,\cdot\,) \in \mathrm{BC}$ *with* $\boldsymbol{R}\big(\langle \mathbf{cup}\rangle^X(n,\cdot\,)\big) = \bigcup_{m\in\omega} \boldsymbol{R}\big(\langle n\rangle^X(m,\cdot\,)\big)$.

Proof. Let $\langle \mathbf{neg}\rangle^X(n,0) = 0$ and $\langle \mathbf{neg}\rangle^X(n,m+1) = \langle n\rangle^X(m)$ for all $m \in \mathbb{N}$. Let $\langle \mathbf{cup}\rangle^X(n,0) = 1$ and $\langle \mathbf{cup}\rangle^X(n,m+1) = \langle n\rangle^X(\tau^{-1}(m))$. □

We now define the notion of $\Delta^1_1(X)$ function. We do this in terms of Borel codes, since the Borel subsets of 2^ω are exactly all the Δ^1_1-sets. For more on this notion of Δ^1_1-function, see [Mos09, Section 3D and Theorem 3E.5].

Definition 10. *A map* $f\colon 2^\omega \to 2^\omega$ *is* $\Delta^1_1(X)$ *for* $X \subseteq \mathbb{N}$ *if there is a* $\Delta^1_1(X)$ *sequence of functions* $(\gamma_\sigma)_{\sigma\in 2^{<\omega}}$ *such that* $\boldsymbol{R}(\gamma_\sigma) = f^{-1}(\widehat{\sigma})$.

The following result is standard (see, e.g., [Mos09, Chapter 2]).

Lemma 11. *A map* $f\colon 2^\omega \to 2^\omega$ *is Borel if and only if* f *is* $\Delta^1_1(X)$ *for some* $X \subseteq \mathbb{N}$.

1.4. **A characterization of feedback computable functions.** We now show that we can isolate Borel codes for those oracles that cause a feedback computation to halt with a tree of subcomputations of height at most α.

Proposition 12. *There is a computable collection of codes for feedback machines,* $e_\downarrow, e_{\downarrow j}, e_\uparrow, e_{\uparrow j}$ *for* $j \in \omega$, *such that*

- *for all countable ordinals* α,
- *for all* $X \subseteq \omega$,
- *for all* $\sigma \in 2^{<\omega}$,
- *for all* $f \in \omega$, *and*
- *for all* $x \in \{\downarrow,\uparrow\} \cup \{\downarrow,\uparrow\} \times \omega$,

$\langle e_x\rangle^{X,\alpha}(f,n,\sigma,\cdot\,)$ *is a Borel code, which we will refer to as* ζ_x.

Further, whenever $Y \in 2^\omega$ *with* $\sigma \prec Y$ *and* $j \in \omega$ *the following properties hold.*

- $Y \in \boldsymbol{R}(\zeta_\downarrow)$ *if and only if* $\langle f\rangle^{X,Y}(n)$ *halts with a tree of subcomputations of height* $\leq \alpha$.
- $Y \in \boldsymbol{R}(\zeta_{\downarrow j})$ *if and only if* $\langle f\rangle^{X,Y}(n)$ *halts with a tree of subcomputations of height* $\leq \alpha$ *and outputs* j.
- $Y \in \boldsymbol{R}(\zeta_\uparrow)$ *if and only if* $\langle f\rangle^{X,Y}(n)$ *does not halt and the tree of subcomputations is of height* $\leq \alpha$.
- $Y \in \boldsymbol{R}(\zeta_{\uparrow j})$ *if and only if* $\langle f\rangle^{X,Y}(n)$ *does not halt after* j*-many steps, and up to the* jth *step the tree of subcomputations has height* $\leq \alpha$.

Finally, we always have $Y \notin \boldsymbol{R}(\zeta_x)$ *if* $\sigma \not\prec Y$.

Proof. We begin with some notation. Enumerate all triples (η,ν,k) such that η and ν are elements of $2^{<\omega}$ and $\{f\}^{X,\eta}_\nu(n)$ does not make any invalid oracle calls

in the first k-many steps, i.e., does not make any oracle queries outside of η or ν. Call this collection B.

For each $\boldsymbol{\eta} = (\eta, \nu, k) \in B$, let $\langle (r_i^{\boldsymbol{\eta}}, m_i^{\boldsymbol{\eta}}) \rangle_{i \leq \ell^{\boldsymbol{\eta}}}$ be the sequence such that $\langle \tau(r_i^{\boldsymbol{\eta}}, m_i^{\boldsymbol{\eta}}) \rangle_{i \leq \ell^{\boldsymbol{\eta}}}$ are the queries made by $\{f\}_\nu^{X,\eta}(n)$ to ν. Note that the length of the sequence is $\ell^{\boldsymbol{\eta}}$. Recall these are called halting queries. In particular if ν is the correct response to every halting query made by $\langle f \rangle^{X,Y}(n)$, i.e., ν agrees with $h_{X,Y}$, then $\{f\}_\nu^{X,Y}(n) \cong \langle f \rangle^{X,Y}(n)$. With this notation we will use the convention that $\nu(\tau(a,b)) = 0$ means ν "believes" $\langle a \rangle^{X,Y}(b)$ halts and $\nu(\tau(a,b)) = 1$ means ν "believes" $\langle a \rangle^{X,Y}(b)$ does not halt.

Our goal will be to define Borel codes $C_{\boldsymbol{\eta}}^\alpha$ for all $\boldsymbol{\eta} \in B$ in such a way that $C_{\boldsymbol{\eta}}^\alpha$ is the Borel code of all Y extending η such that the behavior of halting calls of $\langle f \rangle^{X,Y}(n)$ on the first k-many steps agrees with that of ν, and for which the tree of computations has height at most α.

Let $B_\downarrow$ be the collection of triples $(\eta, \nu, k) \in B$ such that $\{f\}_\nu^{X,\eta}(n)$ halts in at most k steps. For $j \in \omega$, let $B_{\downarrow j}$ be the collection of triples in $B_\downarrow$ with output j. Let $B_{\uparrow j}$ be the collection of triples in $B \setminus B_\downarrow$ whose third coordinate is j.

For each $\boldsymbol{\eta} = (\eta, \nu, k) \in B$ we define a Borel code $C_{\boldsymbol{\eta}}$ as follows.

(i) If $\langle f \rangle^{X,\eta}(n)$ makes no halting queries, then $C_{\boldsymbol{\eta}}^\alpha$ is a code for $\widehat{\eta}$.

(ii) If $\{f\}_\nu^{X,\eta}$ makes any halting queries and $\alpha = 0$, then $C_{\boldsymbol{\eta}}^\alpha$ is a Borel code for $\emptyset$.

(iii) Suppose $\nu(\tau(r_i^{\boldsymbol{\eta}}, m_i^{\boldsymbol{\eta}})) = 0$, i.e., $\{f\}_\nu^{X,\eta}$ "thinks" $\langle r_i^{\boldsymbol{\eta}} \rangle^{X,Y}(m_i^{\boldsymbol{\eta}})$ halts. Then let $D_{\boldsymbol{\eta},i}^\alpha$ be the Borel code which is the union of the Borel codes of $\langle e_h \rangle^{X,\beta}(r_i^{\boldsymbol{\eta}}, m_i^{\boldsymbol{\eta}}, \eta, \cdot\,)$ for $\beta < \alpha$. In other words $D_{\boldsymbol{\eta},i}$ is a Borel code for the collection of those Y such that $\langle r_i^{\boldsymbol{\eta}} \rangle^{X,Y}(m_i^{\boldsymbol{\eta}}) = 0$ and has a tree of subcomputations of rank less than α. Or said another way, $D_{\boldsymbol{\eta},i}^\alpha$ is a Borel code for the collection of those Y such that $\langle f \rangle^{X,Y}(n)$ agrees with ν on the ith halting query and has a tree of subcomputations of rank less than α.

Suppose $\nu(\tau(r_i^{\boldsymbol{\eta}}, m_i^{\boldsymbol{\eta}})) = 1$, i.e., $\{f\}_\nu^{X,\eta}$ "thinks" $\langle r_i^{\boldsymbol{\eta}} \rangle^{X,Y}(m_i^{\boldsymbol{\eta}})$ does not halt. Then for $j \in \omega$ and $\beta < \alpha$, let $E_{\boldsymbol{\eta},i,j,\beta}^\alpha$ be the Borel code of the union of $\langle e_{\uparrow j} \rangle^{X,\beta}(r_i^{\boldsymbol{\eta}}, m_i^{\boldsymbol{\eta}}, \eta, \cdot\,)$. In other words $E_{\boldsymbol{\eta},i,j,\beta}^\alpha$ is a Borel code for those Y such that $\langle r_i^{\boldsymbol{\eta}} \rangle^{X,Y}(m_i^{\boldsymbol{\eta}})$ hasn't halted by step j and which has a tree of subcomputations of height at most β. Now for $\gamma < \alpha$ let $D_{\boldsymbol{\eta},i}^{\alpha,\gamma}$ be the Borel code of the intersection of $E_{\boldsymbol{\eta},i,j,\beta}^\alpha$ over $j \in \omega$ and $\beta < \gamma$. So $D_{\boldsymbol{\eta},i}^{\alpha,\gamma}$ is a Borel code for those Y such that $\langle r_i^{\boldsymbol{\eta}} \rangle^{X,Y}(m_i^{\boldsymbol{\eta}})$ doesn't halt and has a tree of subcomputations of height at most γ. Now let $D_{\boldsymbol{\eta},i}^\alpha$ be the union of $D_{\boldsymbol{\eta},i}^{\alpha,\gamma}$ over $\gamma < \alpha$. So $D_{\boldsymbol{\eta},i}^\alpha$ is a Borel code for those Y such that $\langle r_i^{\boldsymbol{\eta}} \rangle^{X,Y}(m_i^{\boldsymbol{\eta}})$ doesn't halt and has a tree of subcomputations of height at less than α.

Let $C_{\boldsymbol{\eta}}^\alpha$ be the intersection of $D_{\boldsymbol{\eta},i}^\alpha$ for $i \leq \ell^{\boldsymbol{\eta}}$. Note that as $\ell^{\boldsymbol{\eta}}$ is finite the rank of the tree of subcomputations of anything in $C_{\boldsymbol{\eta}}^\alpha$ bounded by the supremum of the tree of subcomputations of $\langle r_i^{\boldsymbol{\eta}} \rangle^{X,Y}(m_i^{\boldsymbol{\eta}})$ plus 1. Therefore

the rank of the tree of subcomputations of $\langle f\rangle^{X,Y}(n)$ is at most α for all $Y \in C^\alpha_{\boldsymbol{\eta}}$.

Finally for $j \in \omega$ we define the codes as follows.

(I) $\langle e_h\rangle^{X,\alpha}(f, n, \eta, \cdot\,)$ is the union of $C^\alpha_{\boldsymbol{\eta}}$ such that $\boldsymbol{\eta} \in B_\downarrow$.
(II) $\langle e_{\downarrow j}\rangle^{X,\alpha}(f, n, \eta, \cdot\,)$ is the union of $C^\alpha_{\boldsymbol{\eta}}$ such that $\boldsymbol{\eta} \in B_{\downarrow j}$.
(III) $\langle e_{\uparrow j}\rangle^{X,\alpha}(f, n, \eta, \cdot\,)$ is the union of $C^\alpha_{\boldsymbol{\eta}}$ such that $\boldsymbol{\eta} \in B_{\uparrow j}$.
(IV) $\langle e_{\uparrow}\rangle^{X,\alpha}(f, n, \eta, \cdot\,)$ is the intersection of the codes given by $\langle e_{\uparrow j}\rangle^{X,\alpha}(f, n, \eta, \cdot\,)$ for $j \in \omega$.

Claim 13. *$C^\alpha_{\boldsymbol{\eta}}$ is a Borel code of the set of all Y extending η such that the behavior of $\langle f\rangle^{X,Y}(n)$ on the first k-many steps agrees with that of ν, and for which the tree of computations has height at most α.*

Proof. Our proof that these codes satisfy our theorem proceeds by induction on α. Notice by conditions (i) and (ii) that if $\alpha = 0$ then $C^\alpha_{\boldsymbol{\eta}}$ satisfies the claim.

But then by induction on α and condition (iii), $D^\alpha_{\boldsymbol{\eta},i}$ is a Borel code for those Y extending η for which the ith halting call agree with ν and the tree of subcomputations has height $< \alpha$. This then implies that $C^\alpha_{\boldsymbol{\eta}}$ satisfies the claim. □

It is then straightforward to check from Claim 13 that these definitions satisfy the proposition. □

Before moving on to the main application of Proposition 12, it is worth taking a moment to highlight the importance of the ordinal α. Specifically, if we did not have a uniform bound on the height of the tree of subcomputations we were considering, we might accidentally make a halting query which would cause our computation to freeze — causing the entire construction to break. However, we show in Lemma 14 that this is not an issue, as there will always be a single bound on all trees of subcomputations, which is itself feedback computable from X.

Lemma 14. *Suppose e is a code (with respect to X) for a feedback computable function from 2^ω to 2^ω. Then there is a countable ordinal α which is feedback computable in X such that for every $Y \in 2^\omega$ and $n \in \mathbb{N}$ the tree of subcomputations of $\langle e\rangle^{X,Y}(n)$ has height bounded by α.*

Proof. Let $P(\beta)$ be the statement "$(\exists Y \in 2^\omega)(\exists n \in \mathbb{N})$ such that β is the height of a tree of subcomputations for $\langle e\rangle^{X,Y}(n)$". Then $P(\,\cdot\,)$ is a $\Sigma^1_1(X)$ predicate. Hence, by [Sac90, Chapter II Ex. 5.9], there is some ordinal α hyperarithmetic in X such that α bounds all β satisfying P. Further, by [AFL15, Theorem 16], α is feedback computable in X because α is hyperarithmetic in X.

However, for all $Y \in 2^\omega$ and $n \in \mathbb{N}$, the feedback computation $\langle e\rangle^{X,Y}(n)$ does not freeze and hence its tree of subcomputations is well-founded. In particular this implies that α bounds the height of the tree of subcomputations of $\langle e\rangle^{X,Y}(n)$ for all $Y \in 2^\omega$ and $n \in \mathbb{N}$. □

Proposition 15. *Suppose $f\colon 2^\omega \to 2^\omega$ is a feedback computable map (with respect to X). Then f is $\Delta^1_1(X)$.*

Proof. Let e be a code (with respect to X) for the map f, and let α be as in Lemma 14. By Proposition 12, there is a uniformly computable (in X and α) collection of Borel codes $\zeta_{\downarrow j}$ such that $\boldsymbol{R}(\zeta_{\downarrow j}) = \{Y : \langle e\rangle^{X,Y}(n) = j \text{ and } \langle e\rangle^{X,Y}(n) \text{ has a tree of subcomputations of height } < \alpha\}$. But then $\boldsymbol{R}(\zeta_{\downarrow j}) = \{Y : \langle e\rangle^{X,Y}(n) = j\}$, as the trees of subcomputations for feedback machines of the form $\langle e\rangle^{X,Y}(n)$ have height $< \alpha$. Hence there is a collection of Borel codes $(\gamma_\sigma)_{\sigma\in 2^{<\omega}}$, uniformly feedback computable in X, such that $\boldsymbol{R}(\gamma_\sigma) = f^{-1}(\widehat{\sigma})$ for each $\sigma \in 2^{<\omega}$. But then by [AFL15, Theorem 16], the functions $\gamma_\sigma\colon \mathbb{N} \to \mathbb{N}$ are $\Delta^1_1(X)$ uniformly in σ. Therefore f is $\Delta^1_1(X)$. □

Proposition 16. *Suppose $f\colon 2^\omega \to 2^\omega$ is $\Delta^1_1(X)$. Then f is feedback computable (with respect to X).*

Proof. There is a sequence $\langle\gamma_\sigma\rangle_{\sigma\in 2^{<\omega}}$ which is in $\Delta^1_1(X)$ such that for $\sigma \in 2^{<\omega}$ the real γ_σ is a Borel code for $f^{-1}(\widehat{\sigma})$. By Theorem 3, we therefore have that $\langle\gamma_\sigma\rangle_{\sigma\in 2^{<\omega}}$ is feedback computable from X.

Then by Lemma 8 there is a feedback machine e such that $\langle e\rangle^{X,Y}(n) = 1$ if there is a σ such that $\mathrm{len}(\sigma) = n+1$, $\sigma(n) = 1$ and $Y \in \boldsymbol{R}(\gamma_\sigma)$, and $\langle e\rangle^{X,Y}(n) = 0$ otherwise. In other words, $\langle e\rangle^{X,Y}(n)$ is the value of $f(Y)(n)$. Hence e is a code (relative to X) for f. □

At this point, we have accomplished our main purpose in this section, to provide a machine model for Borel functions from 2^ω to 2^ω, by showing that the Borel functions are exactly the feedback computable functions. To round this out, we include some other characterizations of feedback computable functions.

If f is feedback computable (mention of the parameter $X \subseteq \omega$ will be suppressed), then $f(Y)$ is in any admissible set containing Y. So f is uniformly Σ_1 definable over all admissible sets, as $f(Y) = Z$ iff within any admissible set containing Y there is a tree witnessing the computation of $f(Y)$ [AFL15, Proposition 4]. In fact, f can trivially be extended to a function on the entire universe V, by letting $f(Y)$ be the empty set whenever Y is not a real. Conversely, suppose $f\colon V \to V$ is uniformly Σ_1 definable over all admissible sets, and f takes reals to reals. Then, as a function on reals, f is Σ^1_1; namely, $f(Y) = Z$ iff there is a real coding an ω-standard admissible set modeling $f(Y) = Z$. (In a little more detail, it costs nothing to say the model is ω-standard, as you can insist that the members of the model's version of ω are given by the evens in their natural order. Furthermore, the ordinal standard part of an admissible set is itself admissible, so an ω-standard admissible set containing Y, even if non-standard, will also contain $f(Y)$.) It is folklore that every Σ^1_1 function is Δ^1_1, as $f(Y) \neq Z$ iff there is a W such that $f(Y) = W$ and $W \neq Z$.

For another characterization of the functions in question, van de Wiele [vdW82] showed that a function is uniformly Σ_1 over all admissible sets iff it is E-recursive.

This was later extended by Slaman [Sla86] (see also [Lub88] for a different proof[1]) to include hereditarily countable parameters. Slaman's result is that for a hereditarily countable parameters p, a function f is uniformly $\Sigma_1(p)$ over all admissible sets iff f is ES_p recursive, where ES_p recursion is E-recursion augmented by selection from p, a schema first identified in [Hoo82] and further studied in [Sla85]. In our case, the parameter is a real X; by Gandy Selection, selection from a real follows from the regular E-recursion schema [Sla85], so that f is uniformly $\Sigma_1(X)$ over all admissible sets iff f is E-recursive in X.

Summarizing the above, we have the following theorem.

Theorem 17. *For any function $f\colon 2^\omega \to 2^\omega$ and any $X \subseteq \omega$ the following are equivalent.*

(a) *f is $\Delta^1_1(X)$.*
(b) *f is feedback computable with respect to X.*
(c) *f can be extended to a function on V which is uniformly $\Sigma_1(X)$ definable over all admissible sets.*
(d) *f can be extended to a function E-recursive in X.*

2. Feedback computability relative to a structure

In this section, we extend the notion of an oracle from a set of natural numbers to a structure (up to isomorphism). If the structure is countable, it can be coded as a set of natural numbers, however this cannot be done if the structure is uncountable. As such, we want our definition of computation from a structure to ultimately be independent of any coding of our structure. This can be seen as a feedback analogue of Medvedev reducibility on isomorphism classes of structures. (For a survey of Muchnik and Medvedev degrees, see [Hin12].) We will make use of the fact (Theorem 17) that Borel functions can be thought of as those that are feedback computable from an oracle, and the fact that the isomorphism classes of countable structures are Borel (see [Kec95, Theorem 16.6]).

Definition 18. *A countable language L is* **feedback computable** *if the sets of relation, function, and constant symbols in L, and their arities, are uniformly feedback computable.*

A particular feedback computable enumeration of such data gives rise to a natural encoding of each countable L-structure with underlying set $\mathbb{N}$, which we will sometimes call its **L-encoding**, and often use implicitly.

We now give two definitions of different kinds of structures that can be computed from a structure independent of any coding.

[1]There is a minor mistake in the latter which can easily be corrected. Slaman's proof uses selection from the parameter p. It is mistakenly claimed in [Lub88] that selection from p is not necessary. In fact, the construction in [Lub88] is perfectly good; it's just that the use of selection from p in the construction was overlooked.

2.1. **Feedback computing expansions.** We now introduce the notion of feedback computing an expansion of a structure.

Definition 19. *Let $L_0 \subseteq L_1$ be feedback computable languages, $\mathcal{M}_0$ be a countable L_0-structure, and $\mathcal{M}_1$ a countable L_1-structure which is an expansion of $\mathcal{M}_0$, i.e., $\mathcal{M}_1|_{L_0} = \mathcal{M}_0$. Then e* **feedback computes the expansion $\mathcal{M}_1$ of oracle $\mathcal{M}_0$,** *written $\langle e\rangle^{\mathcal{M}_0} \succeq^+ \mathcal{M}_1$, if for every L_0-structure $\mathcal{M}_0^* \cong \mathcal{M}_0$ with underlying set $\mathbb{N}$, there is a L_1-structure $\mathcal{M}_1^* \cong \mathcal{M}_1$ such that $\mathcal{M}_1^*|_{L_0} = \mathcal{M}_0^*$ and $\langle e\rangle^{i_0(\mathcal{M}_0^*)}$ is a total function with $\langle e\rangle^{i_0(\mathcal{M}_0^*)} = i_1(\mathcal{M}_1^*)$, where i_j is the natural L_j-encoding of $\mathcal{M}_j^*$ (for $j \in \{0,1\}$).*

As an example, consider the case where $L_0 = (\mathrm{id}, \times)$ and $L_1 = (\mathrm{id}, \times, (\cdot)^{-1})$ with id a constant, $\times$ a binary function, and $(\cdot)^{-1}$ a unary function. Suppose $\mathcal{M}_0$ is a group in the language L_0 and $\mathcal{M}_1$ is the same group but in the language L_1 (i.e., with the inverse function). Then, as we can feedback compute the inverse function when we are passed a group, there is a feedback machine which computes the expansion $\mathcal{M}_1$ of $\mathcal{M}_0$.

Lemma 20. *Whenever there is an injection $f\colon \mathcal{M}_0 \to \mathbb{N}$, the statement $\langle e\rangle^{\mathcal{M}_0} \succeq^+ \mathcal{M}_1$ is absolute among all models of* $\mathrm{ZF} + \mathrm{DC}$ *containing ω_1, $\mathcal{M}_0$, f, and $\mathcal{M}_1$.*

Proof. Suppose that W^+ and W^- are two models of $\mathrm{ZF} + \mathrm{DC}$ containing ω_1, $\mathcal{M}_0$, f and $\mathcal{M}_1$. Further suppose $W^- \models \langle e\rangle^{\mathcal{M}_0} \succeq^+ \mathcal{M}_1$.

Let i_j be the natural encoding of $\mathcal{M}_j$ (for $j \in \{0,1\}$). As f is an injection from $\mathcal{M}_0$ into $\mathbb{N}$ we can assume without loss of generality that the underlying set of $\mathcal{M}_0$ is $\mathbb{N}$. Let $E^+(\mathcal{M}_0, \mathcal{M}_1)$ be the statement "for all countable well-founded $V_0 \models \mathrm{ZFC}^*$ in which $\mathcal{M}_0, \mathcal{M}_1 \in V_0$ with $\mathcal{M}_1|_{L_0} = \mathcal{M}_0$, for all $\mathcal{M}_0^* \cong \mathcal{M}_0$ in V_0 there exists $\mathcal{M}_1^* \cong \mathcal{M}_1$ with $\mathcal{M}_1^*|_{L_0} = \mathcal{M}_0^*$ in V_0 such that $\langle e\rangle^{i_0(\mathcal{M}_0^*)} = i(\mathcal{M}_1^*)$" (where ZFC^* is a finite subset of ZFC large enough for all necessary consequences to follow from ZFC^*). By Lemma 3 and Proposition 4 of [AFL15], for any $\mathcal{M}_0^* \cong \mathcal{M}_0$ with underlying set $\mathbb{N}$, the computation $\langle e\rangle^{i_0(\mathcal{M}_0^*)}$ doesn't depend on the model of ZFC. Hence, as $W^- \models \langle e\rangle^{\mathcal{M}_0} \succeq^+ \mathcal{M}_1$ we also have $W^- \models E^+(\mathcal{M}_0, \mathcal{M}_1)$. But $E^+(\,\cdot\,,\,\cdot\,)$ is Π^1_2 and so by Shoenfield absoluteness we also have $W^+ \models E^+(\mathcal{M}_0, \mathcal{M}_1)$. Hence $W^+ \models \langle e\rangle^{\mathcal{M}_0} \succeq^+ \mathcal{M}_1$. □

It follows from Lemma 20 that the following definition is well-defined and doesn't depend on the specific forcing extension. Note that this can be seen as a feedback computability analogue of relations being uniformly relatively intrinsically (u.r.i.) computable (see [Mon17]).

Definition 21. *Let $L_0 \subseteq L_1$ be feedback computable languages, $\mathcal{M}_0$ be a (not necessarily countable) L_0-structure, and $\mathcal{M}_1$ an L_1-structure which is an expansion of $\mathcal{M}_0$, i.e., $\mathcal{M}_1|_{L_0} = \mathcal{M}_0$. Then e* **feedback computes the expansion $\mathcal{M}_1$ of oracle $\mathcal{M}_0$,** *written $\langle e\rangle^{\mathcal{M}_0} \succeq^+ \mathcal{M}_1$, if there is some forcing extension $V[G]$ of the universe in which $\mathcal{M}_0$ is countable and $V[G] \models \langle e\rangle^{\mathcal{M}_0} \succeq^+ \mathcal{M}_1$.*

We now define what it means for a subset of a structure to be feedback computable.

Definition 22. *Suppose L is a language and $\mathcal{M}$ is a (not necessarily countable) L-structure. Suppose $U \subseteq \mathcal{M}$ is fixed by all automorphisms of $\mathcal{M}$ and $\mathcal{M}_U$ is the expansion of $\mathcal{M}$ which adds U as a new unary predicate. Then U is* **feedback computable** *from $\mathcal{M}$ if there is an $e \in \mathbb{N}$ such that $\langle e \rangle^{\mathcal{M}} \succeq^+ \mathcal{M}_U$.*

Note that by Lemma 20 the specific forcing extension is irrelevant. The reason why we require U to be closed under automorphisms of $\mathcal{M}$ is so that the set U is uniquely defined by $\mathcal{M}_U$. Our main use of this notion is when $\mathcal{M}$ is of the form $(\gamma, \in_\gamma, A)$ for some ordinal γ and finite subset A, where $\in_\gamma$ denotes the relation $\in$ restricted to γ.

2.2. **Feedback computing a structure.** Having defined what it means for an expansion of a structure to be feedback computable, we now define what it means for a structure to be feedback computable from another structure.

Definition 23. *Let L_0, L_1 be feedback computable languages and let $\mathcal{M}_j$ be a countable L_j-structure for $j \in \{0,1\}$. Then e* **feedback computes $\mathcal{M}_1$ from $\mathcal{M}_0$,** *written $\langle e \rangle^{\mathcal{M}_0} \succeq \mathcal{M}_1$, if for every L_0-structure $\mathcal{M}_0^* \cong \mathcal{M}_0$ with underlying set $\mathbb{N}$ there is an L_1-structure $\mathcal{M}_1^* \cong \mathcal{M}_1$ with underlying set $\mathbb{N}$ such that $\langle e \rangle^{i_0(\mathcal{M}_0^*)}$ is a total function with $\langle e \rangle^{i_0(\mathcal{M}_0^*)} = i_1(\mathcal{M}_1^*)$, where i_j is the natural L_j-encoding of $\mathcal{M}_j^*$ (for $j \in \{0,1\}$).*

As an example, let G be a group and H a normal subgroup. Consider the structure given by the group G along with a distinguished relation for H. We can feedback compute the group G/H by simply choosing a representative from each coset of H along with the group multiplication table for these representatives induced by multiplying the corresponding cosets.

Note that for a structure $\mathcal{M}$ with underlying set $\mathbb{N}$ and natural encoding i, the notation $\langle e \rangle^{i(\mathcal{M})}$ denotes the feedback computable (partial) function from $\mathbb{N}$ to $\mathbb{N}$ that takes as an oracle the natural encoding of $\mathcal{M}$. In contrast, $\langle e \rangle^{\mathcal{M}}$ will not be used on its own, and $\langle e \rangle^{\mathcal{M}} \succeq \mathcal{N}$ (or $\langle e \rangle^{\mathcal{M}} \succeq^+ \mathcal{N}$) can be thought of as saying that no matter what copy of $\mathcal{M}$ is passed as an oracle to $\langle e \rangle$, the output is always a copy of $\mathcal{N}$ (or in the case of $\succeq^+$ a copy of $\mathcal{N}$ which is also an expansion of $\mathcal{M}$).

It is worth noting that if $\langle e \rangle^{\mathcal{M}} \succeq^+ \mathcal{N}$ then we also have $\langle e \rangle^{\mathcal{M}} \succeq \mathcal{N}$. However the converse need not hold as if $\langle e \rangle^{\mathcal{M}} \succeq \mathcal{N}$ the output may be a structure whose restriction to $L_{\mathcal{M}}$ (the language of $\mathcal{M}$) is only isomorphic to $\mathcal{M}$ and not equal to it.

Lemma 24. *Whenever there are injections $f_j \colon \mathcal{M}_j \to \mathbb{N}$ (for $j \in \{0,1\}$), the statement $\langle e \rangle^{\mathcal{M}_0} \succeq \mathcal{M}_1$ is absolute among all models of* ZF + DC *containing ω_1, $\mathcal{M}_0$, $\mathcal{M}_1$, f_0, and f_1.*

Proof. Suppose that W^+ and W^- are two models of ZF + DC containing ω_1, $\mathcal{M}_0$, $\mathcal{M}_1$, f_0, and f_1. Further suppose $W^- \models \langle e \rangle^{\mathcal{M}_0} \asymp \mathcal{M}_1$.

For $j \in \{0, 1\}$, let i_j be the natural encoding of $\mathcal{M}_j$. As f_j is an injection from $\mathcal{M}_j$ into $\mathbb{N}$ we can assume without loss of generality that the underlying set of $\mathcal{M}_j$ is $\mathbb{N}$. Let $E(\mathcal{M}_0, \mathcal{M}_1)$ be the statement "for all countable well-founded $V_0 \models \text{ZFC}^*$ in which $\mathcal{M}_0, \mathcal{M}_1, f_0, f_1 \in V_0$, for all $\mathcal{M}_0^* \cong \mathcal{M}_0$ in V_0 there exists $\mathcal{M}_1^* \cong \mathcal{M}_1$ in V_0 such that $\langle e \rangle^{i_0(\mathcal{M}_0^*)} = i(\mathcal{M}_1^*)$" (where again ZFC^* is a finite subset of ZFC large enough for all necessary consequences to follow from ZFC^*). By Lemma 3 and Proposition 4 of [AFL15], for any $\mathcal{M}_0^* \cong \mathcal{M}_0$ with underlying set $\mathbb{N}$, the computation $\langle e \rangle^{i_0(\mathcal{M}_0^*)}$ doesn't depend on the model of ZFC. Hence, as $W^- \models \langle e \rangle^{\mathcal{M}_0} \asymp \mathcal{M}_1$ we also have $W^- \models E(\mathcal{M}_0, \mathcal{M}_1)$. But $E(\,\cdot\,, \cdot\,)$ is Π^1_2 and so by Shoenfield absoluteness we also have $W^+ \models E(\mathcal{M}_0, \mathcal{M}_1)$. Hence $W^+ \models \langle e \rangle^{\mathcal{M}_0} \asymp \mathcal{M}_1$. □

The absoluteness of the relation of feedback reducibility between structures (Definition 23) allows us to make sense of feedback reducibility between structures even when those structures happen to be uncountable by considering the question of reducibility in a forcing extension where the structures are countable.

It follows from Lemma 24 that the following definition is well-defined and doesn't depend on the specific forcing extension.

Definition 25. *Let L_0, L_1 be feedback computable languages, $\mathcal{M}_0$ be a (not necessarily countable) L_0-structure, and $\mathcal{M}_1$ a (not necessarily countable) L_1-structure. Then e* **feedback computes** $\mathcal{M}_1$ **from** $\mathcal{M}_0$*, written $\langle e \rangle^{\mathcal{M}_0} \asymp \mathcal{M}_1$, if there is some forcing extension $V[G]$ of the universe in which $\mathcal{M}_0$ is countable and $V[G] \models \langle e \rangle^{\mathcal{M}_0} \asymp \mathcal{M}_1$.*

The relative computability of uncountable structures was studied using generic extensions and Muchnik degrees in [KMS16]. Our consideration of the feedback reducibility of uncountable structures can be seen as a feedback analogue of these notions, except using the analogue of Medvedev degrees instead of Muchnik degrees, because of the uniformity of our reductions.

Furthermore, both notions of feedback reducibility that take a structure as an oracle (Definitions 19 and 23) allow us to perform computation in a way that ignores the particular instantiations of the structures. This is important, as there are times when there is more computable information that can be obtained by the encoding of the structure than can be obtained intrinsically from the structure.

2.3. **Example: Functions from ω to ω.** There is one example of computing one structure from another which is particularly important. Note that if $(W, \lhd, c)$ is a well-ordering of order type ω and c is a constant in W, then, uniformly in an encoding of an oracle $(W, \lhd, c)$, we can return the element of $\mathbb{N}$ which c represents. Therefore if $\langle e \rangle^{\mathcal{M}} \asymp (W, \lhd, c)$ there is little harm in identifying the output with the number c represents. In particular we can define $\langle e \rangle^{\mathcal{M}}(n) = m$

if $\langle e\rangle^{\mathcal{M}\times(\omega,\in,n)} \succeq (\omega,\in,m)$. Hence we can think of $\langle e\rangle^{\mathcal{M}}(n) = m$ as saying that whenever e is handed a copy of $\mathcal{M}$ as an oracle, along with the natural number n, it outputs the natural number m.

As an example of this, suppose G is a torsion group. There is a feedback computable function $\langle e\rangle$ which takes n, computes the nth prime p and returns the smallest $m > 0$ such that there is a subgroup of size p^m if such an m exists and returns 0 otherwise. While the input depends on the specific group, it does not depend on the encoding of the group.

The following is then immediate from Lemma 24.

Lemma 26. *Let $f\colon \mathcal{M} \to \mathbb{N}$ be an injection. The statement $\langle e\rangle^{\mathcal{M}}(n) = m$ is absolute among all models of* ZF + DC *containing ω_1, $\mathcal{M}$, and f.*

2.4. **Example: α-infinite time Turing machines.** For an admissible ordinal α, the (α,α)-infinite time Turing machines (ITTMs) provide a different model which captures the α-computable sets. We now show how to represent this model via oracle feedback computation.

The following definition is a straightforward generalization of the (∞,ω)-Turing machines of [HL00] and the (α,α)-Turing machines in [KS09].

Definition 27. *Let α,β be ordinals. A (run of a) α-**time, β-space Turing machine** (referred to as an (α,β)-*ITTM*) consists of the following data.*

- *A function $T\colon \alpha\times\beta \to \{0,1\}$, called the* **tape**. *For $\gamma\in\alpha$, the function $T(\gamma,\cdot)\colon \beta\to\{0,1\}$ is called the* **values** *of the tape at time γ. The values at time 0 are called the* **initial values**.
- *A function $H\colon \alpha\to\beta$, called the* **head location**.
- *A function $S\colon \alpha\to E$, called the* **state space**, *where E is a finite linearly ordered set containing a special starting state s and halting state h. For $\gamma\in\alpha$, the value $S(\gamma)$ is called the* **state** *of the machine at time γ.*
- *A function $C\colon E\times\{0,1\}\to E\times\{0,1\}\times\{\text{LEFT},\text{RIGHT},\text{STAY}\}$, called the* **look up table**. *It can be thought of as taking the state of the machine and the symbol written under the tape and returning the new state, the new symbol, and whether to move the head left or right, or to have it stay where it is.*

This data is required to satisfy the following conditions.

- *$C(h,z) = (h,z,\text{STAY})$ for all $z\in\{0,1\}$.*
- *$H(0) = 0$ and $S(0) = s$.*
- *If $\gamma\in\alpha$ is a limit ordinal then $H(\gamma) = \liminf_{\zeta\in\gamma} H(\zeta)$ and $S(\gamma) = \liminf_{\zeta\in\gamma} S(\zeta)$.*
- *If $\gamma+1\in\alpha$ and $(e,z,M) = T\big(S(\gamma),T(\gamma,H(\gamma))\big)$, then the following hold.*
  - *$S(\gamma+1) = e$.*
  - *$T(\gamma+1,H(\gamma)) = z$.*
  - *$T(\gamma+1,\eta) = T(\gamma,\eta)$ for $\eta\neq H(\gamma)$.*

- *If* $M = \text{STAY}$ *then* $H(\gamma+1) = H(\gamma)$.
- *If* $M = \text{RIGHT}$ *then* $H(\gamma+1) = H(\gamma)+1$.
- *If* $M = \text{LEFT}$ *and* $H(\gamma) = p+1$ *then* $H(\gamma+1) = p$.
- *If* $M = \text{LEFT}$ *and* $H(\gamma)$ *is a limit ordinal then* $H(\gamma+1) = 0$.

The **input** *of the machine is* $T(0, \cdot)$. *The machine* **halts** *if there is some* $\gamma < \alpha$ *such that* $S(\gamma) = h$, *and in this case,* $T(\gamma, \cdot)$ *is the* **output** *of the machine (which is well-defined by the first condition).*

We will refer to an (α, α)-ITTM *as simply an* α-ITTM.

We now show how to perform such computations using feedback.

Lemma 28. *There is a feedback machine* **ittm** *such that if*

$$\mathcal{M}_{\alpha,\beta,X,C} := \langle (\alpha, \in_\alpha), (\beta, \in_\beta), X, C \rangle$$

where $X \colon \beta \to \{0,1\}$ *and* C *is a lookup table, then*

$$\langle \mathbf{ittm} \rangle^{\mathcal{M}_{\alpha,\beta,X,C}} \cong \mathcal{N}_{\alpha,\beta,X,C}$$

where $\mathcal{N}_{\alpha,\beta,X,C} = \langle (\alpha, \in_\alpha), (\beta, \in_\beta), T_X, H_X, S_X, C \rangle$, *with* (T_X, H_X, S_X, C) *the (unique)* (α,β)*-ITTM with* $T_X(0, \cdot) = X$ *and code* C.

Proof. Note that given (α, β), a code C, and initial values X, the definition of an (α, β)-ITTM uniquely determines the functions T_X, H_X, and S_X by a transfinite recursion along α that is uniform in α, β, and X. Given a representation of an ordinal, we can feedback computably identify if an element of that representation corresponds to a limit ordinal (and if not find its successor). Hence from any isomorphic copy of $\mathcal{M}_{\alpha,\beta,X,C}$ we can feedback compute $\mathcal{N}_{\alpha,\beta,X,C}$. □

Henceforth all (α, β)-ITTMs will have $\alpha = \beta$. As is standard in this situation, we will imagine that there is an input tape, an output tape, and an extra parameter tape (in which all but finitely many values are 0). This can be encoded into the ITTM in the standard way by interleaving these three tapes.

Definition 29. *A function* $f \colon \mathcal{P}_{<\omega}(\alpha) \to \mathcal{P}_{<\omega}(\alpha)$ *is* **α-ITTM computable** *if there is an* α-ITTM *with a fixed finite extra parameter set such that when the input tape is the characteristic function of* A *for some finite sequence* A *of elements of* α, *then the* α-ITTM *halts with the characteristic function of* $f(A)$ *on the output tape. The notion of* α-ITTM *computability naturally extends to functions* $f \colon \alpha^n \to \alpha^m$ *for* $n, m \in \mathbb{N}$.

Lemma 30. *Let* α *and* γ *be ordinals. Suppose* $(\alpha, \lhd)$ *is well-ordered with order type* γ *where* $\lhd$ *is* α-ITTM *computable. Then for all finite* $B \subseteq \gamma$ *there is a feedback machine* e *such that* $\langle e \rangle^{(\alpha, \in_\alpha, A)} \asymp (\gamma, \in_\gamma, B)$ *for some finite* $A \subseteq \alpha$.

Proof. For any finite lookup table C, from $(\alpha, \in_\alpha, A)$ we can feedback compute $\mathcal{M}_{\alpha,\alpha,A,C}$ (via a feedback machine that intrinsically encodes C). Hence for any such C, the structure $\mathcal{N}_{\alpha,\alpha,A,C}$ is feedback computable from $(\alpha, \in_\alpha, A)$ by Lemma 28.

By assumption, there is some C and some finite $A \in \mathcal{P}_{<\omega}(\alpha)$ such that from $\mathcal{N}_{\alpha,\alpha,A,C}$ we can feedback compute $(\gamma, \in_\gamma, B)$. Hence for some $A \in \mathcal{P}_{<\omega}(\alpha)$, the structure $(\gamma, \in_\gamma, B)$ can be feedback computed from $(\alpha, \in_\alpha, A)$. □

2.5. **Feedback computation from ordinals.** For the remainder of this section, we consider an extended example, feedback computability relative to a countable admissible ordinal.

Proposition 31. *Let α be an admissible ordinal, and suppose $\gamma < \alpha^+$. Let A be a finite subset of γ, and suppose that e is such that the function $\langle e \rangle^{(\gamma, \in_\gamma, A)}$ feedback computes the set $U \subseteq \gamma$. Then $U \in L(\alpha^+)$.*

Proof. First note we can assume without loss of generality that α is countable, as if it isn't we can move to a forcing extension where α is countable. Next note that the partial ordering $(\alpha^{<\omega}, \preceq) \in L(\alpha^+)$ where $a \preceq b$ if a is an initial segment of b. Let G be a generic for $(\alpha^{<\omega}, \preceq)$. Then G is a surjection from ω onto α. By [Ers90, Theorem 1], the set $L(\alpha^+)[G]$ is admissible. Let $G^* \in 2^\omega$ be a real encoding $(\alpha, \in_\alpha)$ such that $G^* \in L(\alpha^+)[G]$.

Note that there must be a real $M \in L(\alpha^+)[G]$ which encodes the structure $(\gamma, \in_\gamma, A)$ and let M_U be the corresponding real encoding $(\gamma, \in_\gamma, A, U)$. Then M_U is feedback computable from M (by assumption). Therefore, by [AFL15, Proposition 16], the structure M_U is in any admissible set containing M and in particular is in $L(\alpha^+)[G]$. In particular this implies that $U \in L(\alpha^+)[G]$. But as G was an arbitrary generic for $(\alpha^{<\omega}, \preceq)$, we must have that $U \in L(\alpha^+)$, as desired. □

Hence there is an upper bound on how complicated a function can be that is feedback computable from an admissible ordinal α. It is an interesting open question to pin down exactly how complicated the sets feedback computable from α can be. This is a question that we completely answer when α is a Gandy ordinal. It is worth pointing out that in Proposition 31 we cannot simply absorb the finite set A into the code of the program, as the specific natural numbers representing the elements of A depend on the particular representation of $(\gamma, \in_\gamma)$.

Proposition 32. *There is an e such that $\langle e \rangle^{(\gamma, \in_\gamma)} \asymp (L(\gamma), \in_{L(\gamma)})$. Further e is independent of γ.*

Proof. Without loss of generality we can assume γ is countable. From $(\gamma, \in_\gamma)$ we can feedback compute the set of all pairs (φ, A, α) where φ is a first order formula, $\alpha \in_\gamma \gamma$, and A is a tuple of ordinals in α of length the number of free variables in φ. We can then feedback compute the relation $(\varphi_0, A_0, \alpha_0) \in_{L(\gamma)} (\varphi_1, A_1, \alpha_1)$ by induction on α_0 and α_1. Next, by induction on α we can compute an equivalence relation $\equiv$ where $(\varphi_0, A_0, \alpha) \equiv (\varphi_1, A_1, \alpha)$ if and only if they contain the same elements. Finally we can let $L(\gamma)$ consist of a single representative of each $\equiv$-class. □

We then have the following corollary.

Corollary 33. *If there is an α-computable well-ordering of α of height γ, then for any finite $B \subseteq \gamma$ there is an $e \in \omega$ and a finite subset $A \subseteq \alpha$ such that $\langle e \rangle^{(\alpha, \in_\alpha, A)} \asymp (\gamma, \in_\gamma, B)$.*

Proof. We can feedback compute $(\alpha + 1, \in_{\alpha+1})$ from $(\alpha, \in_\alpha)$, and so by Proposition 32 we can feedback compute $L(\alpha + 1)$ from α. But every α-computable well-ordering of α is in $L(\alpha + 1)$ so for some finite A we can compute $(\gamma, \in_\gamma)$. □

This suggests the following definition.

Definition 34. *An admissible ordinal α is defined to be a* **feedback Gandy** *ordinal if for all $\gamma < \alpha^+$ there is a well-ordering $(\alpha, \lhd)$ of order type γ which is feedback computable from $(\alpha, \in_\alpha, A)$ for some finite $A \subseteq \alpha$.*

In particular, Corollary 33 shows that all Gandy ordinals are feedback Gandy ordinals. Whereas it has been established that there are admissible ordinals that are not Gandy ordinals [Gos79], it is an open question whether or not every admissible ordinal is a feedback Gandy ordinal.

The following corollary is then immediate.

Corollary 35. *If α is a feedback Gandy ordinal and $\gamma < \alpha^+$, then for all finite $B \subseteq \gamma$ there is a finite $A \subseteq \alpha$ such that $(\gamma, \in_\gamma, B)$ is feedback computable from $(\alpha, \in_\alpha, A)$.*

Proposition 36. *Let $\alpha < \gamma$ be ordinals, let A be a finite subset of α, and let $e \in \omega$. Suppose that $U \in L(\gamma)$ is such that $U \subseteq \alpha$, and suppose that $\langle e \rangle^{(\alpha, \in_\alpha, A)} \asymp (\gamma, \in_\gamma)$. Then there is some finite $A^* \subseteq \alpha$ and $e^* \in \omega$ such that $\langle e^* \rangle^{(\alpha, \in_\alpha, A^*)} \asymp^+ (\alpha, \in_\alpha, U)$.*

Proof. This follows immediately from Proposition 32 by letting $A^* := A \cup \{a\}$, where a is a code for U in a definable bijection from γ to $L(\gamma)$. □

Combining Proposition 31 and Corollaries 35 and 36, we obtain the following.

Theorem 37. *If α is a feedback Gandy ordinal then the following are equivalent for $U \subseteq \alpha$.*

- *$U \in L(\alpha^+)$.*
- *U is feedback computable from $(\alpha, \in_\alpha, A)$ for some $A \in \mathcal{P}_{<\omega}(\alpha)$.*

3. Open questions

We end with several open questions. For each of these questions, let α be an ordinal and A a finite subset of α.

- For what β is there some set $U \in L(\beta + 1) \setminus L(\beta)$ that is feedback computable from $(\alpha, \in_\alpha, A)$?
- If there is some $U \in L(\beta + 1) \setminus L(\beta)$ that is feedback computable from $(\alpha, \in_\alpha, A)$, must $L(\beta + 1)$ be feedback computable from $(\alpha, \in_\alpha, A^*)$ for some finite $A^* \subseteq \alpha$?

- Which ordinals are feedback Gandy? In particular, are there feedback Gandy ordinals that are not Gandy ordinals? Indeed, are there any admissible ordinals that are not feedback Gandy ordinals?

Acknowledgements

The authors would like to thank Julia Knight, Russell Miller, Noah Schweber, and Richard Shore for helpful conversations.

References

[AFL15] N. L. Ackerman, C. E. Freer, and R. S. Lubarsky, *Feedback Turing computability, and Turing computability as feedback*, 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, IEEE, 2015, pp. 523–534.

[AS76] F. G. Abramson and G. E. Sacks, *Uncountable Gandy ordinals*, J. London Math. Soc. (2) **14** (1976), no. 3, 387–392.

[Ers90] Y. L. Ershov, *Forcing in admissible sets*, Algebra and Logic **29** (1990), no. 6, 424–430 (1991).

[Gos79] R. Gostanian, *The next admissible ordinal*, Ann. Math. Logic **17** (1979), no. 1-2, 171–203.

[Hin12] P. G. Hinman, *A survey of Mučnik and Medvedev degrees*, Bull. Symbolic Logic **18** (2012), no. 2, 161–229.

[HL00] J. D. Hamkins and A. Lewis, *Infinite time Turing machines*, J. Symbolic Logic **65** (2000), no. 2, 567–604.

[Hoo82] M. R. R. Hoole, *Recursion on sets*, Ph.D. thesis, Univ. of Oxford, 1982.

[Kec95] A. S. Kechris, *Classical descriptive set theory*, Graduate Texts in Mathematics, vol. 156, Springer-Verlag, New York, 1995.

[KMS16] J. Knight, A. Montalbán, and N. Schweber, *Computable structures in generic extensions*, J. Symb. Log. **81** (2016), no. 3, 814–832.

[KS09] P. Koepke and B. Seyfferth, *Ordinal machines and admissible recursion theory*, Ann. Pure Appl. Logic **160** (2009), no. 3, 310–318.

[Lub88] R. S. Lubarsky, *Another extension of Van de Wiele's theorem*, Ann. Pure Appl. Logic **38** (1988), no. 3, 301–306.

[Mon17] A. Montalbán, *Computable structure theory*, Draft of Part I, https://math.berkeley.edu/~antonio/CSTpart1.pdf, 2017.

[Mos09] Y. N. Moschovakis, *Descriptive set theory*, second ed., Mathematical Surveys and Monographs, vol. 155, American Mathematical Society, Providence, RI, 2009.

[Sac90] G. E. Sacks, *Higher recursion theory*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1990.

[Sla85] T. A. Slaman, *Reflection and forcing in E-recursion theory*, Ann. Pure Appl. Logic **29** (1985), no. 1, 79–106.

[Sla86] ———, *Σ_1-definitions with parameters*, J. Symbolic Logic **51** (1986), no. 2, 453–461.

[vdW82] J. van de Wiele, *Recursive dilators and generalized recursions*, Proc. Herbrand Symposium, Logic Colloquium '81 (J. Stern, ed.), North-Holland, Amsterdam, 1982, pp. 325–332.

Department of Mathematics, Harvard University, Cambridge, MA 02138
E-mail address: nate@math.harvard.edu

Remine, Fairfax, VA 22031
E-mail address: cameron@remine.com

Department of Mathematical Sciences, Florida Atlantic University, Boca Raton, FL 33431
E-mail address: Robert.Lubarsky@alum.mit.edu