Decidable Effectively Closed Sets and Acceptably Equivalent Numberings

KATIE BRODHEAD
Florida A&M University
Department of Mathematics
Tallahassee, Florida 32307
USA
katiebrodhead@gmail.com

Abstract: A Π^0_1 class is an effectively closed set of reals. One way to view it is as the set of infinite paths through a computable tree. We consider the notion of *acceptably equivalent* numberings of Π^0_1 classes. We show that a permutation exists between any two acceptably equivalent numberings that preserves the computable content. Furthermore the most commonly used numberings of the Π^0_1 classes are acceptably equivalent. We also consider decidable Π^0_1 classes in enumerations. A decidable Π^0_1 class may be represented by a unique computable tree *without dead ends*, but we show that this tree may not show up in an enumeration of uniformly computable trees which gives rise to all Π^0_1 classes. In fact this is guaranteed to occur for some decidable Π^0_1 class. These results are motivated by structural questions concerning the upper semi-lattice of enumerations of Π^0_1 classes where notions such as acceptable equivalence arise.

Key-Words: Effectively closed sets, Numberings

1 Introduction

Many results in classical computability theory are derived from a study of the indices of partial computable functions. For example, the Enumeration Theorem allows indices to be treated as arguments. Conversely, the S^m_n Theorem allows arguments to be treated as indices. So it is desirable that these and other results be independent of the chosen system of indices.

It is known that if a system of indices is *acceptable* then it has same structure theory as any system that satisfies the Enumeration and S^m_n theorems. A system of indices ϕ is a family of surjective maps $\phi^n\colon \omega \to$ {n-ary partial recursive functions} [12]. Let ϕ be a system of indices that satisfies the Enumeration and S^m_n theorems and call it the *standard system* [9]. A system of indices φ is *acceptable* if, for every n, there are total computable functions f and g such that $\psi^n_e \simeq \phi^n_{f(e)}$ and $\phi^n_e \simeq \psi^n_{g(e)}$ [11]. For a greater treatment on acceptable systems of indices for partial recursive functions, see [9]. In this paper we develop a notion of acceptability for Π^0_1 classes.

A Π^0_1 class is an effectively closed set of reals in ω^ω, although we shall restrict our attention to classes in 2^ω. Alternatively we may also consider a Π^0_1 class to be the set of infinite paths in through a computable tree in $\omega^{<\omega}$. One way to enumerate them is $P_e = \omega^\omega \setminus \bigcup_{n\in W_e} I(\sigma_n)$ [1]. (Here W_e is the e^{th} c.e. set in the standard system, σ_n is the n^{th} string in the enumeration $\sigma_0, \sigma_1, \sigma_2, \cdots$ of all strings in $\omega^{<\omega}$, and $I(\sigma_n)$ is the set of elements in ω^ω that extend σ_n.) As a result, Π^0_1 classes have index-argument related properties inherited from the Enumeration and S^m_n theorems. We shall use an alternate enumeration method which takes advantage of this property and justifiably call this the standard numbering of Π^0_1 classes.

Our work follows in the path of previous work done by Jockusch, Rogers, and Soare [13, p. 25] for acceptably equivalent number-

-ings of the partial recursive functions, and hence of the c.e. sets. A permutation exists between any two acceptably equivalent numberings which preserves the original computable content. We use the standard numbering for c.e. sets to extend this result to Π_1^0 classes. Furthermore we show that the most frequently used numberings in Π_1^0 classes are acceptable with respect to the standard numbering. We develop these notions below.

In the mid-1950s, initiated under Kolmogorov, work began on generalized theory of numberings and continued under the direction of Mal'tsev and Ershov [4]. A *numbering* of a collection C of objects is a surjective map $\mathrm{F}: \omega \rightarrow C$. An *enumeration without repetition* is an injective numbering. Given two numberings ν and u, we say that u is *acceptable with respect to* ν, denoted $\nu \leq u$, iff there is a total computable function f such that $\nu = u \circ f$. Then u is *acceptable* if it acceptable with respect to all numberings. We say that ν and u are *acceptably equivalent*, denoted $\nu \equiv u$, iff $\nu \leq u$ and $u \leq \nu$. Note that $\equiv$ is an equivalence relation and let $\mathcal{L}(C)$ denote the set of all numberings of C modulo $\equiv$. It is easy to verify that $\mathcal{L}(C)$ is an upper semilattice under $\leq$. Furthermore enumerations without repetition occur only in the minimal elements of this semilattice and acceptable enumerations occur only in the greatest element of the semilattice. It is well established that these types of enumerations do exist.

In 1958 Friedburg [5] showed that an enumeration of the c.e. sets exists without repetition. Goncharov, Lempp, and Solomon [6] further generalized this result for *n*-c.e. sets. An interesting result by Suzuki [14] shows that an enumeration of the computable sets exists without repetition. However our goal is a set of corresponding results for Π_1^0 classes.

Recently Raichev [10] proved that an enumeration of the Π_1^0 classes exists without repetition. Using a modification of the Friedberg's proof for c.e. sets, he gives an enumeration of the Σ_1^0 sets without repetition. The corresponding result related to the Goncharov-Lempp-Solomon theorem concern- -ing differences of Σ_1^0 classes remains unsolved. Concerning the Suzuki theorem, we turn to decidable Π_1^0 classes.

A decidable Π_1^0 class is the set of infinite paths through a computable tree *without dead ends*. In one way, decidable Π_1^0 classes resemble the recursive sets in the same way that Π_1^0 classes resemble the c.e. sets. In c.e. sets it is unknown immediately whether an element will show up in an enumeration. In Π_1^0 classes it is also unknown if a branch in the corresponding tree will eventually end up as a dead end. However in computable sets and decidable Π_1^0 classes (given the proper representation) such things are known. Given the result of Suzuki, it seems plausible that decidable Π_1^0 classes can be enumerated without repetition.

To show such an enumeration exists it is natural to follow Friedburg's approach, utilized by Odifreddi, Goncharov, Lempp, Solomon, Raichev, and others. We attempt to do so but with surprising results. Under the assumption that every computable tree without dead ends shows up in an enumeration of uniformly computable trees representing all the Π_1^0 classes, the proof appears to succeed. However diagonalization immediately provides for a computable tree without dead ends not in the enumeration. Therefore although a decidable Π_1^0 class may be represented by a computable tree without dead ends, its tree may not show up in an enumeration of uniformly computable trees representing all the Π_1^0 classes. For some decidable Π_1^0 class this is guaranteed to happen. Subsequently complexity results for index sets for decidable Π_1^0 classes and for computable trees without dead ends are distinct. We note that the results on index sets for decidable Π_1^0 classes in [2] use the convention that a class P_e is decidable iff the corresponding tree T_e has no dead ends. In light of our new theorem, those results need to be revisited. We generalize these enumeration results to subfamilies of Π_1^0 classes and to trees with $\leq n$ dead ends, for fixed n. It remains open whether decidable Π_1^0 classes may be enumerated without repetition.

2 Numberings of Effectively Closed Sets

In this section, present some basic notation and facts about Π_1^0 classes which lead to different methods of enumerating them. Finally, we present six different enumerations of them.

Basic Notations; Facts about Π_1^0 classes

The partial computable $\{0,1\}$-valued functions are indexed as $\{\phi_e\}_{e\in\omega}$ and the primitive recursive functions as $\{\pi_e\}_{e\in\omega}$. As usual $\phi_{e,s}$ denotes that portion of function ϕ_e defined by stage s. We use $\phi_e(x)\downarrow$ to mean that ϕ_e is defined on input x. Similarly $\phi_e(x)\uparrow$ signifies that the function is undefined. We shall use σ and τ to represent strings in $\omega^{<\omega}$. Let $\langle\tau\rangle\in\omega$ denote the usual code for a finite string. Recall that $T\sqsubseteq\omega^{<\omega}$ is a *tree* iff it is closed under initial segments. Let $[T]$ be the set of infinite paths through the tree T. P is a *Π_1^0 class* iff $P=[T]$ for some computable tree T. We have the following result from [2].

Proposition 1. *For any class $P\subset\omega^\omega$ the following are equivalent:*

(a) $P=[T]$ for some computable tree $T\subset\omega^{<\omega}$;
(b) $P=[T]$ for some primitive recursive tree T;
(c) $P=\{x:(\forall n)R(n,x)\}$, for some computable relation R;
(d) $P=[T]$ *for some Π_1^0 tree $T\subset\omega^{<\omega}$.*

Following this proposition, Cenzer and Remmel mention two possible numberings of the Π_1^0 classes that occur as a consequence. We develop these concepts here.

Numbering 1: Primitive Recursive Functions

For each e, $U_e=\{\emptyset\}\cup\{\sigma:(\forall\tau\sqsubseteq\sigma)\,\pi_e(\langle\tau\rangle)=1\}$ defines a primitive recursive tree. To see that this enumeration contains all primitive recursive trees, observe that if $\{\sigma:\pi_e(\sigma)=1\}$ is a tree then U_e is that tree. By part (b), $e\mapsto U_e$ is a tree enumeration of the Π_1^0 classes.

Numbering 2: Total Computable Functions

Since the complexity of the set Tot of indices for total computable functions is Π_2^0, any numbering $\omega\to$ Tot must naturally be non-effective. We include such a result as such an example.

Let $\Lambda=\{e\in\text{Tot and }T_e=\{\sigma:\phi_e(\sigma)=1\}\text{ is a tree}\}$. By part (a), $\Lambda\subseteq\omega$ is an indexing of all Π_1^0 classes. To obtain a numbering, we will define a map on all of ω by defining the mapping on $\overline{\Lambda}$. We consider the method of proving (a) → (b) in Proposition 1. One can show that if $P=[T_e]$ with computable T_e then $[T_e]=[S_e]$ with primitive recursive

$$S_e=\{\sigma:(\forall n<|\tau|)\neg\phi_{e,|\tau|}(\tau\restriction n)=0\}.$$

Now consider the following proposition.

Proposition 2. [2, p. 9]

(i) There exists a primitive recursive function ϕ such that if ϕ_e defines a computable tree T_e then $S_e=U_{\phi(e)}$. So $[T_e]=[U_{\phi(e)}]$.
(ii) There is a primitive recursive function π such that, for each e, $U_e=T_{\pi(e)}$.

The following is a numbering of the Π_1^0 classes based on an indexing of trees

$$T_e=\{\sigma:\phi_e(\sigma)=1\}$$

arising solely from the total {0,1}-computable functions in $\{\phi_e\}_{e\in\omega}$.

$$e\mapsto\begin{cases}T_e & \text{if } e\in\text{Tot and } T_e \text{ is a tree}\\ T_{\pi(\phi(e))} & \text{otherwise}\end{cases}$$

In [1], Cenzer describes two other methods of enumerating the Π_1^0 classes.

Numbering 3: Computably Enumerable Sets

Utilizing part (d) of Proposition 1,

$$P_e=\omega^\omega\setminus\bigcup_{n\in W_e}I(\sigma_n)$$

gives an enumeration of the Π_1^0 classes. We officially denote it by

$$e\mapsto\{\sigma:\forall\langle m,s\rangle[\phi_{e,s}(m)\downarrow\Rightarrow\ \sigma\not\sqsupseteq\sigma_{\phi_{e,s}(m)}]\}.$$

Numbering 4: C.E. Sets (Primitive Recursive Version)

Modifying the previous numbering we can get an numbering that has the dual feature of being a enumeration of uniformly primitive recursive trees and being based on the c.e. sets. This numbering is given by

$$e\mapsto\{\sigma:(\forall\tau\sqsubseteq\sigma)\langle\tau\rangle\notin W_{e,|\sigma|}\}.$$

We call this the standard numbering of the Π_1^0 classes.

Another method commonly found in the literature (see [8], for example) utilizes a version of Halting Problem concerned with diagonal computation with oracles.

Numbering 5: The Halting Problem

Consider the mapping $\psi: \omega \to$ {class of all Π_1^0 trees} given by $e \mapsto \{\sigma: (\forall s)\phi_{e,s}^{\sigma}(e)\uparrow\}$. From part (d) of the proposition, $\psi(n)$ codes a Π_1^0 class for all n. To show that Im(ψ) codes all Π_1^0 classes, let φ be any numbering of the Π_1^0 classes given by trees. We show that there is a computable function g such that $\varphi = \psi \circ g$. For all n let $\phi_{g(e)}^{\sigma}(n)$ be defined only if $\sigma \notin \varphi(e)$. Then $\sigma \in (\psi \circ g)(e) \Leftrightarrow \phi_{g(e)}^{\sigma}(g(e))\uparrow$ $\Leftrightarrow \sigma \in \varphi(e)$.

Numbering 6: Universal Π_1^0 Relation

There is a universal Π_1^0 relation $U \subseteq \omega \times 2^{\omega}$ such that if $D(x)$ is a Π_1^0 relation then there is an $e \in \omega$ such that $D(x) \leftrightarrow U(e,x)$ [7, p. 78]. Therefore by part (c), $e \mapsto \{x: U(e,x)\}$ is a numbering of the Π_1^0 classes.

We may obtain a tree numbering as follows. Suppose that

$$U(e,x) = (\forall n)R(n,e,x)$$

where R is a computable relation. There is a computable function ν and a computable functional $\Phi_{\nu(e)}$ such that $R(n,e,x) \Leftrightarrow \Phi_{\nu(e)}^{x}(n) = 1$ and $\neg R(n,e,x) \Leftrightarrow \Phi_{\nu(e)}^{x}(n) = 0$. Define the tree

$$S_{\nu(e)} = \{\sigma: (\forall n < |\sigma|)\Phi_{\nu(e)}^{\sigma}(n) = 1\}.$$

Then $\{x: U(e,x)\} = [S_{\nu(e)}]$ and we obtain the numbering $e \mapsto S_{\nu(e)}$.

We used each part of Proposition 1 to give different numberings for the Π_1^0 Classes. Numbering 2 has the distinct feature of being non-effective. Collectively, however, each shared the common feature that they could ultimately be considered numberings of trees. This is due to the very definition of a Π_1^0 class as the set of infinite paths through a computable tree. In this next section we consider which of these are numberings are acceptably equivalent to one another.

3 Acceptable Numberings of Effect--ively Closed Sets

In this section we consider the notion of acceptably equivalent numberings of Π_1^0 classes and show that all of the enumerations given in the previous section are acceptably equivalent, up to the complexity of a given numbering. This expands upon the corresponding work for partial computable functions. We have the following.

Theorem 3. [13, p. 25] *Consider the standard numbering φ of the partial computable functions $\{\phi_e\}_{e\in\omega}$ which represents an effective listing of all Turing programs. Let ψ be any acceptably equivalent numbering. Then there is a computable permutation p of ω such that $\varphi = \psi \circ p$.*

The proof is similar to our result in Theorem 5. It uses the following proposition, also found in [13, p 25], whose proof utilizes the same construction used to prove the Myhill Isomor--phism Theorem.

Proposition 4. *Let $\omega = \bigcup_n A_n = \bigcup_n B_n$ where the sequences $\{A_n\}_{n\in\omega}$ and $\{B_n\}_{n\in\omega}$ are each pairwise disjoint. Let f and g be injective computable functions such that $f(A_n) \subseteq B_n$ and $g(B_n) \subseteq A_n$ for all n. Then there is a computable permutation p such that $p(A_n) = B_n$ for all n.*

So any two acceptably equivalent numberings yield the same computable content since there is a computable permutation that can switch back and forth between the indices. The same is true in Π_1^0 classes.

Theorem 5. *Let φ be the standard numbering of the Π_1^0 classes. Let ψ be any acceptably equivalent numbering. Then there is a comput--able permutation p of ω such that $\varphi = \psi \circ p$.*

Proof: Recall that φ is represented by $e \mapsto P_e = \{\sigma: (\forall \tau \sqsubseteq \sigma)\langle\tau\rangle \notin W_{e,|\sigma|}\}$. We shall represent ψ by $e \mapsto Q_e$. Since φ and ψ are acceptably equivalent there are total computable functions f and g such that for all x, $P_{f(e)} = Q_e$ and

$Q_{g(e)} = P_e$. Let $k_0 = 0$ and let k_n be the least a such that $P_a \neq P_{k_m} (\forall m < n)$. Define $G_n = \{e: P_e = P_{k_n}\}$ and $H_n = \{e: Q_e = P_{k_n}\}$. Then $\omega = \bigcup_n G_n = \bigcup_n H_n$ and the sequences $\{G_n\}_{n\in\omega}$ *and* $\{H_n\}_{n\in\omega}$ are each pairwise disjoint. Furthermore $f(H_n) \subseteq G_n$ and $g(G_n) \subseteq H_n$. To complete the proof it suffices by Proposition 4 to convert f and g into injective computable functions f_1 and g_1 satisfying the same property.

Convert f to f_1:

f satisfies $P_{f(e)} = Q_e$ and $f(H_n) \subseteq G_n$. Now f may not be injective, but since $f(e)$ is in the standard numbering, the Padding Lemma for c.e. sets applies. Therefore there is a computable function h such that $W_a = W_{h(i,a)}$ for all i and a, and if $i \neq j$ then $h(i,a) \neq h(j,b)$ for any a or b. Let $f_1(e) = h(e, f(e))$. Then f_1 satisfies $P_{f_1(e)} = Q_e$ and $f_1(H_n) \subseteq G_n$. Furthermore f_1 is injective.

Convert g to g_1:

To define g_1 we must be able (uniformly in e) to effectively generate an infinite set S_e of indices such that for each $i \in S_e$ we have that $Q_i = Q_{g(e)}$. We can then ensure that g_1 is injective, similar to the argument as for f_1. We cannot use the Padding Lemma since that requires the standard numbering. So we use a different approach.

Take any two disjoint computably inseparable c.e. sets A and B. Let $a_0, a_1, a_2, \ldots$ be an enumeration of A without repetition. Let A_n and B_n denote the sets A and B, respectively, up to stage n. Also let $T_0, T_1, T_2, \ldots$ be a tree enumeration of the Π^0_1 classes. \space For any $\sigma \in \omega^\omega$, let $E_\sigma = 1$ if $|\sigma|$ is even and 0 otherwise. Now let $e, i \in \omega$. Consider the computable relation $P(e,i,\sigma)$ defined by: $P(e,i,\sigma) \leftrightarrow$

$$\sigma \in T_e \text{ or}$$
$$(\sigma \sqsubseteq 0^{a_0+1}1^{a_1+1}0^{a_2+1}\ldots E_\sigma^{a_{|\sigma|}+1} \text{ and } i \notin A_{|\sigma|})$$

Define computable trees $T_{k(e,i)}$ and $T_{l(e,i)}$ as follows:

$$\sigma \in T_{k(e,i)} \leftrightarrow P(e,i,\sigma) \text{ AND } i \notin B_{|\sigma|}$$
$$\sigma \in T_{l(e,i)} \leftrightarrow P(e,i,\sigma)$$

It follows that $P_{k(e,i)}$ and $P_{l(e,i)}$ are given by:

$$P_{k(e,i)} = \begin{cases} P_e & \text{if } i \in A \\ \emptyset & \text{if } i \in B \\ P_e \cup \{0^{a_0+1}1^{a_1+1}\ldots\} & \text{otherwise} \end{cases}$$
$$P_{l(e,i)} = \begin{cases} P_e & \text{if } i \in A \\ P_e \cup \{0^{a_0+1}1^{a_1+1}\ldots\} & \text{otherwise} \end{cases}$$

Let $C_e = \{k(e,i): i \in A\}$ and $D_e = \{l(e,i): i \in A\}$. We claim that for each e, $S_e = g(C_e) \cup g(D_e)$ is infinite, thereby completing the proof. To show this, we shall prove that either $g(C_e)$ or $g(D_e)$ is infinite. There are two cases.

Case I: ($P_e \neq \emptyset$). It follows that for some distinct m and n, that

$\{k(e,i): i \in A\} \subseteq \{a: P_a = P_e\} \subseteq G_n$ and
$\{k(e,i): i \in B\} \subseteq \{a: P_a = \emptyset\} \subseteq G_m$

are disjoint. Now since $Q_{g(e)} = P_e$ for all e, after applying g to each set the new sets remain disjoint. If $g(C_e)$ is finite, say $g(C_e) = \{c_1, c_2, \ldots, c_l\}$, then

$$C'_e = \{i \in \omega: g(k(e,i)) \in \{c_1, c_2, \ldots, c_l\}\}$$

is computable and $A \subseteq C'_e$ and $B \cap C'_e = \emptyset$, contrary to A and B being computably inseparable. Therefore $g(C_e)$ is infinite.

Case II: ($P_e = \emptyset$). It follows that

$$\{0^{a_0+1}1^{a_1+1}\ldots\} \notin P_e$$

so that $P_e \neq P_e \cup \{0^{a_0+1}1^{a_1+1}\ldots\}$. By a similar argument to that above, $g(D_e)$ is infinite. QED

Next we show that all of our numberings are acceptably equivalent up to the complexity of a given numbering. We use all the same notation as before and use $e \mapsto T_e$ to denote a specific tree numbering of the Π^0_1 classes.

Theorem 6. *In the notation of the previous section, each of the following is a numbering of the Π^0_1 classes:*

(1) ***Primitive Recursive Functions***
$$e \mapsto \{\emptyset\} \cup \{\sigma: (\forall \tau \sqsubseteq \sigma)\pi_e(\langle\tau\rangle) = 1\}$$
(2) ***Total Computable Functions***
$$e \mapsto \begin{cases} T_e & \begin{array}{l} \textit{if } \phi_e \textit{ is total } \& \\ \{\sigma: \phi_e(\sigma) = 1\} \\ \textit{is a tree} \end{array} \\ T_{\pi(\phi(e))} & \textit{otherwise} \end{cases}$$
(3) ***Computably Enumerable (C.E.) Sets***
$$e \mapsto \{\sigma: \forall\langle m,s\rangle\ [\phi_{e,s}(m)\downarrow\ \Rightarrow \sigma \not\sqsupseteq \sigma_{\phi_{e,s}(m)}]\}$$

(4) ***C.E. Sets (Partial Recursive Version)***
$e \mapsto \{\sigma: (\forall\tau \sqsubseteq \sigma)\ \langle\tau\rangle \notin W_{e,|\sigma|}\}$
(5) ***The Halting Problem***
$e \mapsto \{\sigma: (\forall s)\ \phi^{\sigma}_{e,s}(e) \uparrow\}$
(6) ***Universal Π^0_1 Relation***
$e \mapsto \{x: U(e,x)\}$

Any of these can be considered to be the standard numbering in the following sense. If φ and ψ are two distinct numberings, then there exists a permutation p such that $\varphi = \psi \circ p$. The permutation is Δ^0_3 if either φ or ψ is the numbering given in (2). Otherwise the permut--ation is computable.

Proof: We use the notation (i) → (j) to mean that if φ and ψ are the corresponding numberings for (i) and (j) respectively, then there is a total φ -computable function f such that $\varphi = \psi \circ f$. We show that (i) ↔ (j) for $i \neq j$. Then by Theorem 5 we have our result for $i, j \neq 2$. However the same proof given in that theorem demonstrates that if $i = 2$ then the permutation is Π^0_2. Our proof closely models the proof, as given in [2], of Proposition 1. Note that according to this result, (2) is of form (a), ((1), (4)) are of form (b), (6) is of form (c), and ((3), (5)) are of form (d). Accordingly, we show (2) → ((1), (4)) → (6) → ((3), (5)) → (2). To obtain the result for $i \neq 2$ we also show ((3), (5)) → ((1), (4)).

(2) → (1), (4). Let φ, ψ, and γ be the numberings for (2), (1), and (4) respectively. Let $\delta(e)$ denote the index of the tree $\varphi(e) = T_{\delta(e)}$. For each $e \in \omega$, define the primitive recursive tree

$$S_e = \{\sigma: (\forall\tau \sqsubseteq \sigma)\neg\phi_{\delta(e),|\sigma|}(\langle\tau\rangle) = 0\}.$$

We show that $[\varphi(e)] = [S_e]$. Now $S_e \subseteq \varphi(e)$, so that $[S_e] \subseteq [\varphi(e)]$. Now suppose that $x \notin [S_e]$. Then for some n, $x \upharpoonright n \notin S_e$. So there is some m such that $\phi_{\delta(e),m}(x \upharpoonright n) = 0$. Then for any $k > \max\{m,n\}$, we have that $x \upharpoonright k \notin \varphi(e)$. It follows that $x \notin \varphi(e)$.

Now use the S^m_n Theorem to get a Δ^0_3-function g such that $\pi_{g(e)}(\langle\sigma\rangle) = 1 \Leftrightarrow (\forall\tau \sqsubseteq \sigma)\neg\phi_{\delta(e),|\sigma|}(\langle\tau\rangle) = 0$. Then $\varphi = \psi \circ g$. We also have that $\varphi = \gamma \circ \delta$.

(1), (4) → (6). Let φ, ψ, and γ be the number--ings for (1), (4), and (6) respectively. Define the relation R_φ by $R_\varphi(n,e,x) \Leftrightarrow x \upharpoonright n \in \varphi(e)$. Let f_φ be a computable function such that $(\forall n)R(n,e,x) \Leftrightarrow U(f_\varphi(e),x)$. Then $\varphi = \gamma \circ f_\varphi$. Defining R_ψ and f_ψ similarly we obtain $\psi = \gamma \circ f_\psi$.

(6) → (3), (5). We obtained (6) → (5) in discussing Numbering (5). Now let φ and ψ be numberings for (6) and (3) respectively. Define $\phi_{g(\nu(e))}(\langle\sigma\rangle)$

$$= \begin{cases} 1 & \text{if } \exists\langle n,s\rangle\left(n < |\sigma|\ \&\ \Phi^{\sigma}_{\nu(e),s}(n) = 0\right) \\ \uparrow & \text{otherwise} \end{cases}$$

Then $\varphi = \psi \circ (g \circ \nu)$.

(3), (5) → (1), (2), (4). Let φ, ψ, γ, ς, and ι be numberings for (3), (5), (2), (1), and (4), respectively. We have, for all e,

$$\varphi(e) = \{\sigma: (\forall n)R_\varphi(n,e,\sigma)\}$$

with R_φ a recursive relation. Define the computable tree

$$T_{f(e)} = \{\sigma: (\forall m,n \leq |\sigma|)R_\varphi(m,e,\sigma \upharpoonright n)\}.$$

Define $T_{g(e)}$ similarly utilizing the recursive relation R_ψ. Then $\varphi = \gamma \circ f$ and $\psi = \gamma \circ g$.

Now utilize the methods of (2) → (1),(4) with $T_{f(e)}$, $T_{g(e)}$ in place of $T_{\delta(e)}$ to obtain *computable* f', g' such that $\varphi = \varsigma \circ f'$ and $\psi = \varsigma \circ g'$; note that $\varphi = \iota \circ f$ and $\psi = \iota \circ g$. QED

It remains open whether these enumerations only occur in the greatest element of the semi-lattice $\mathcal{L}(\mathcal{P})$, where $\mathcal{P}$ is the class of all Π^0_1 classes. We already have a nice example of an element occuring in a minimal element if this semilattice, namely an enumeration of all Π^0_1 classes without repetition. The next section is motivated by the result of Suzuki that there is an enumeration without repetition of the computable sets. We will study decidable Π^0_1 classes occuring in enumerations of Π^0_1 classes.

4 Decidable Effectively Closed Sets in Numberings

A Π^0_1 class may be represented by many different computable trees. However decidable Π^0_1 classes are unique in that each decidable class D has a unique computable tree without dead ends that represents it. Although every enumeration of the Π_1^0 classes necessarily contains every decidable Π_1^0 class, the unique tree without dead ends does not have to show up in the enumeration. In fact this is guaranteed to occur for some decidable Π_1^0 class in an effective enumeration of uniformly computable trees giving rise to all Π_1^0 classes. As a result, index sets for decidable Π_1^0 classes and for computable trees without dead ends are distinct both as sets and in complexity. Previous results in [2] make no such distinction and consequently must be revisted. We generalize the enumeration results to subfamilies of Π_1^0 classes and to trees with $\leq n$ dead ends. We devote the rest of this paper towards proving these results.

Definition 7. *A tree* $T \sqsubseteq 2^{<\omega}$ *and a set* $[T]$ *are clopen iff there is a nonempty finite set* $S \subseteq \omega$ *such that* $T = \emptyset$ *or* $T = \{\sigma : \sigma \sqsubseteq \sigma_i \text{ or } \sigma_i \sqsubseteq \sigma \text{ for some } i \in S\}$.

Clearly a clopen tree T has no dead ends. Moreover a Π_1^0 class $[T] \subseteq 2^\omega$ is clopen if $2^\omega \backslash [T]$ is clopen. That is $P = [T]$ is clopen iff P is a finite union of intervals $I(\sigma_n)$. Clopen sets will play the role for Π_1^0 classes that finite sets play for c.e. sets.

Theorem 8. *Given any effective enumeration of uniformly computable trees, there exists an enumeration without repetition containing all clopen trees along with all computable trees without dead ends that occur in the enumer- -ation.*

Proof: Friedberg [5] uses in his construction of c.e. sets without repetition the notion of one c.e. set *following* another, so that in the end the constructed set will be the followed set. We use the same term terminology here except in the context of one tree following another.

Let $T_0, T_1, ...$ be an effective enumeration of uniformly computable trees. Take, for example, the standard enumeration of trees corresponding to an effective listing of the Π_1^0 classes. Although we don't require $\{T_e\}_{e\in\omega}$ to contain all clopen trees, we assume, without loss of generality, that they already contain them. We will construct, in stages, a sequence of *follower* trees $S_0, S_1, ...$ to prove the theorem.

At stage i we will ensure that we have $i+1$ trees $S_0, S_1, ..., S_i$, constructed up to level 2^i, following trees $T_{(S_0,k_i)}, \dots, T_{(S_0,k_i)}$ $(k_i \in \{m,n\})$ which are each pairwise distinct at level 2^i. Also, at stage i, initially some of the S_i will have the status of being marked $(k_i = m)$ in which case S_i will continue to follow $T_{(S_i,m)}$ forever. If not, then S_i is not marked $(k_i = n)$ and we determine for each i, if S_i should be marked. If an S_i needs to be marked then we determine a tree $T_{(S_i,m)}$ that it shall hereafter follow. Otherwise each S_i continues to follow $T_{(S_i,n)}$ and the stage is complete.

Construction:

Stage 0. Find the first tree T_i such that $T_i \cap \{0,1\}^{2^0} \neq \emptyset$. Denote this tree as $T_{(S_0,n)}$, and define $S_0 = T_{(S_0,n)} \cap \{0,1\}^{\leq 2^0}$.

Stage j+1. S_0, ..., S_j have already been constructed up to level 2^j and are already following trees $T_{(S_0,k_j)}, \dots, T_{(S_j,k_j)}$. We perform the following two actions at this stage:

(1) Construct S_0, ..., S_j up to level 2^{j+1} by determining the trees $T_{(S_0,k_{j+1})}, \dots, T_{(S_j,k_{j+1})}$ they shall follow, and
(2) Construct a new tree S_{j+1} up to level 2^{j+1}

Action (1). Let $U_{j+1} = \{(S_i,k_j) : k_j = n$ and $T_{(S_i,k_j)}$ has dead ends at level $2^{j+1}\}$. All S_i such that $(S_i,k_j) \notin U_{j+1}$ keep their status as marked or unmarked, so $k_j = k_{j+1}$, and continue to follow $T_{(S_i,k_{j+1})}$. Those S_i such that $(S_i,k_j) \in U_{j+1}$ will hereafter be marked and will now follow the tree $T_{(S_i,m)}$ given by $T_{(S_i,m)} = \{\sigma : \tau \sqsubseteq \sigma \text{ or } \sigma \sqsubseteq \tau \text{ for some } \tau \in T_{(S_i,n)}$ of length $2^j\}$. Note that each marked S_i follows a clopen tree $T_{(S_i,m)}$.

Action (2). Let (S_{j+1},n) be the least i such that T_i is distinct from all $T_{(S_i,k_{j+1})}$ $(i \leq j)$ at level 2^{j+1} and such that T_i has no dead ends.

Define $S_{j+1} = T_{(S_{j+1},n)} \cap \{0,1\}^{\leq j+1}$. This comp- -letes the construction.

Verification:
We now verify that:

(i) For each i, $\lim_j T_{(S_i,k_j)} \downarrow = S_i = T_{n_i}$ for some T_{n_i} without dead ends

(ii) $(\forall i)(T_i$ has no dead ends $\to \exists c\, T_i = S_c)$

(iii) $i \neq j \longrightarrow S_i \neq S_j$

Verification of (i). For all j, $k_j = n$ or $k_j = m$. Fix i. By Action (2), at stage i, $(S_i, k_i) = (S_i, n)$. By Action (1), $k_l = k_{l+1} = n$ for all $l > i$ if S_i is never marked. If S_i is marked at stage $r > i$, then for all $s \geq r$, $k_s = k_{s+1} = m$. In either case $\lim_{j\geq i} k_j \downarrow$ so that $\lim_j (S_i, k_j)$ converges to (S_i, n) or (S_i, m). If it converges to (S_i, m) then S_i never diverges from following the clopen tree $T_{(S_i,m)}$. Otherwise S_i is never marked and continually follows $T_{(S_i,n)}$. Since it is never marked it means that $T_{(S_i,n)}$ never has dead ends up to level 2^r, for all $r > i$. So $T_{(S_i,n)}$ is a tree without dead ends. Either way $\lim_j T_{(S_i,k_j)} \downarrow = T_{n_i}$ for some tree T_{n_i} without dead ends. Now for all n, $S_i \cap \{0,1\}^{\leq n} = T_{(S_i,k_n)} \cap \{0,1\}^{\leq n}$ and $T_{(S_i,k_n)} \subseteq T_{(S_i,k_{n+1})}$. Therefore $S_i = \lim_j T_{(S_i,k_j)} = T_{n_i}$.

Verification of (ii). Let T_i be a tree without dead ends. There are two cases. If there is a stage j and a c such that $T_i = T_{(S_c,m)}$ at stage j, then by the constrction $T_i = S_c$. If not, let $\hat{\imath}$ equal the least k such that $T_k = T_i$. Let j be large enough so that $T_{\hat{\imath}}$ differs from T_e at level 2^j for all $e < \hat{\imath}$. If at stage j there already exists a c such that $T_{\hat{\imath}} = T_{(S_c,n)}$ then clearly $T_i = S_c$. Otherwise, by Action (2), some tree S_c follows $T_{\hat{\imath}}$ by no later than stage $j + \hat{\imath}$.

Verification of (iii). By Action (2), S_i is distinct from all S_j $(j < i)$ at level 2^i and from all S_j $(j > i)$ at level 2^j; $S_i \neq S_j$ if $i \neq j$. QED

Corollary 9. In any enumeration of uniformly computable trees, there is a computable tree without dead ends that does not occur in the enumeration.

Proof: Suppose not. Theorem 8 provides for an enumeration S_0, S_1, S_2, ... without repetition of all computable trees without dead ends. We use a diagonalization argument to construct a tree T so that for all n,

$$T \cap \{0,1\}^{n+1} \neq S_n \cap \{0,1\}^{n+1}.$$

At stage 0 let $T \cap \{0,1\}^0 = \{\emptyset\}$. At stage $n+1$ we are given that $T \cap \{0,1\}^n$ is nonempty. Therefore there are at least 2 subtrees of $\{0,1\}^{n+1}$ extending $T \cap \{0,1\}^n$. Define $T \cap \{0,1\}^{n+1}$ to be an extension which is different from $S_n \cap \{0,1\}^{n+1}$. QED

Corollary 10. Let $\{[\mathrm{T}_e]\}_{e\in\omega}$ be the standard enumeration of the Π^0_1 classes. Then there is a decidable Π^0_1 class P such that $P \neq [T_e]$ for any T_e without dead ends.

As a result of this corollary, $\{e$: T_e has no dead ends$\} \neq \{\, e$: $P_e = [T_e]$ is decidable$\}$. In fact both have distinct complexities. Let $\mathrm{Ext}(P_e) = \{\sigma\colon (\forall\sigma \in T_e)(\forall n)(\exists \tau \in \{0,1\}^n)\ \sigma \parallel \tau \in T_e\}$ where $\parallel$ denotes concatenation. By König's Lemma, since the trees are subsets of $2^{<\omega}$, this set is Π^0_1. Therefore $\{e$: T_e has no dead ends$\} = \{e\colon T_e = \mathrm{Ext}(P_e)\}$ is Π^0_1. However, $\{e$: P_e is decidable$\}$

$= \{e$: $P_e = [T]$ for some computable T without dead ends$\}$

$= \{e$: $(\exists a)\ \phi_a$ is a characteristic function for $\mathrm{Ext}(P_e)\}$

Therefore this latter set is $\sum^0_2$. In [2], no distinction is made between these sets or their complexities. In light of these surprising results, the results of [2] must be revisited. We generalize Theorem 8.

Corollary 11. Let $\mathcal{P}_n = \{P = [T]$ is a Π^0_1 class and T has $\leq n$ dead ends$\}$. Then in any enumeration (of a subfamily) of Π^0_1 classes by uniformly computable trees, there is a Π^0_1 class $[T] \in \mathcal{P}_n$ such that there is no e such that T_e has $\leq n$ dead ends and $[T_e] = [T]$.

Proof: Modify the proof of Theorem 8 so that for fixed n, trees become marked only if they are discovered to have $> n$ dead ends. We leave details to the reader. QED

In particular, the previous result is true for the standard numbering and also the numbering done via the primitive recursive functions. Future research in this area will include the enumeration of differences of Π_1^0 classes as well as the complexity of index sets for decidable Π_1^0 classes.

References:

[1] D. Cenzer, *Effectively Closed Sets*, ASL Lecture Notes in Logic, to appear.

[2] D. Cenzer and J. Remmel, Index Sets for Π_1^0 Classes, *Annals of Pure and Applied Logic*, Vol. 93, 1998, pp. 3-61.

[3] D. Cenzer and J. Remmel, Index Sets for Computable Real Functions, *Proceed--ings of CCA03, Cincinnati,* Preprint, 2003, pp. 163-182.

[4] Y. Ershov, Theory of Numberings, *Handbook of Computability Theory (Edward R. Griffor, ed.)*, North-Holl--and, Amsterdam, 1999.

[5] R. Friedberg, Three Theorems on Recursive Enumeration, *Journal of Symbolic Logic*, Vol. 23, 1958, pp. 309-316.

[6] S. Goncharov, S. Lempp, and R. Solomon, Friedburg Numberings of Families of *n*-Computably Enumerable Sets, *Algebra and Logic*, Vol. 41, 2002, pp. 81-86.

[7] P. Hinman, *Recursion-Theoretic Hier--archies*, Springer-Verlag, 1978.

[8] C. Jockusch and R. Soare, Π_1^0 Classes and Degrees of Theories, *Transactions of the American Mathematical Society*, Vol. 173, 1972, pp. 35-56.

[9] P. Odifreddi, *Classical Recursion The--ory, Vol. 1*, North-Holland, Amster--dam- New York, 1989.

[10] A. Raichev, *Relative Randomness via RK-Reducibility*, Ph.D. thesis, Uni--versity of Wisconsin-Madison, 2006.

[11] H. Rogers, Gödel numberings of partial recursive functions, *Journal of Symbolic Logic*, Vol. 23, 1958, pp. 331-341.

[12] H. Rogers, *Theory of recursive functions and effective computability*, McGraw Hill, 1967.

[13] R. Soare, *Recursively Enumerable Sets and Degrees*, Springer-Verlag, 1987.

[14] Y. Suzuki, Enumerations of recursive sets, *Journal of Symbolic Logic*, Vol. 24, 1959, pp. 311.