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 $\Pi^0_1$ 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 $\Pi^0_1$ 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 $\Pi^0_1$ classes are acceptably equivalent. We also consider decidable $\Pi^0_1$ classes in enumerations. A decidable $\Pi^0_1$ 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 $\Pi^0_1$ classes. In fact this is guaranteed to occur for some decidable $\Pi^0_1$ class. These results are motivated by structural questions concerning the upper semi-lattice of enumerations of $\Pi^0_1$ 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 $\phi$ is a family of surjective maps $\phi^n: \omega \rightarrow \{n\text{-ary partial recursive functions}\}$ [12]. Let $\phi$ 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 $\psi$ is acceptable if, for every $n$, there are total computable functions $f$ and $g$ such that $\psi^n_e \equiv \phi^n(f(e))$ and $\phi^n_e \equiv \psi^n(g(e))$ [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 $\Pi^0_1$ classes.

A $\Pi^0_1$ class is an effectively closed set of reals in $\omega^\omega$, although we shall restrict our attention to classes in $2^\omega$. Alternatively we may also consider a $\Pi^0_1$ 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, $\sigma_n$ 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 $\omega^\omega$ that extend $\sigma_n$.) As a result, $\Pi^0_1$ 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 $\Pi^0_1$ 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 $\Pi^0_1$ classes. Furthermore we show that the most frequently used numberings in $\Pi^0_1$ 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 $F: \omega \rightarrow C$. An enumeration without repetition is an injective numbering. Given two numberings $v$ and $u$, we say that $u$ is acceptable with respect to $v$, denoted $v \leq u$, iff there is a total computable function $f$ such that $v = u \circ f$. Then $u$ is acceptable if it acceptable with respect to all numberings. We say that $v$ and $u$ are acceptably equivalent, denoted $v \equiv u$, iff $v \leq u$ and $u \leq v$. 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 Friedberg [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 $\Pi^0_1$ classes.

Recently Raichev [10] proved that an enumeration of the $\Pi^0_1$ classes exists without repetition. Using a modification of the Friedberg's proof for c.e. sets, he gives an enumeration of the $\Sigma^0_1$ sets without repetition. The corresponding result related to the Goncharov-Lempp-Solomon theorem concerning differences of $\Sigma^0_n$ classes remains unsolved.

Concerning the Suzuki theorem, we turn to decidable $\Pi^0_1$ classes.

A decidable $\Pi^0_1$ class is the set of infinite paths through a computable tree without dead ends. In one way, decidable $\Pi^0_1$ classes resemble the recursive sets in the same way that $\Pi^0_1$ classes resemble the c.e. sets. In c.e. sets it is unknown immediately whether an element will show up in an enumeration. In $\Pi^0_1$ 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 $\Pi^0_1$ classes (given the proper representation) such things are known. Given the result of Suzuki, it seems plausible that decidable $\Pi^0_1$ 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 $\Pi^0_1$ 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 $\Pi^0_1$ 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 $\Pi^0_1$ classes. For some decidable $\Pi^0_1$ class this is guaranteed to happen. Subsequently complexity results for index sets for decidable $\Pi^0_1$ classes and for computable trees without dead ends are distinct. We note that the results on index sets for decidable $\Pi^0_1$ 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 $\Pi^0_1$ classes and to trees with $\leq n$ dead ends, for fixed $n$. It remains open whether decidable $\Pi^0_1$ classes may be enumerated without repetition.

2 Numberings of Effectively Closed Sets
In this section, present some basic notation and facts about \( \Pi^0_1 \) classes which lead to different methods of enumerating them. Finally, we present six different enumerations of them.

**Basic Notations; Facts about \( \Pi^0_1 \) 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 \( \phi_e \) defined by stage \( s \). We use \( \phi_e(x) \downarrow \) to mean that \( \phi_e \) is defined on input \( x \). Similarly \( \phi_e(x) \uparrow \) signifies that the function is undefined. We shall use \( \sigma \) and \( \tau \) to represent strings in \( \omega^{<\omega} \). Let \( \langle \tau \rangle \in \omega \) denote the usual code for a finite string.

Recall that \( T \subseteq \omega^{<\omega} \) is a tree iff it is closed under initial segments.

Let \( \langle T \rangle \) be the set of infinite paths through the tree \( T \). \( P \) is a \( \Pi^0_1 \) class iff \( P = \langle T \rangle \) 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) \( P = \langle T \rangle \) for some computable tree \( T \subset \omega^{<\omega} \);

(b) \( P = \langle T \rangle \) for some primitive recursive tree \( T \);

(c) \( P = \{x : (\forall n)R(n,x)\} \) for some computable relation \( R \);

(d) \( P = \langle T \rangle \) for some \( \Pi^0_1 \) tree \( T \subset \omega^{<\omega} \).

Following this proposition, Cenzer and Remmel mention two possible numberings of the \( \Pi^0_1 \) classes that occur as a consequence. We develop these concepts here.

**Numbering 1: Primitive Recursive Functions**

For each \( e \), \( U_e = \emptyset \cup \{\sigma : (\forall \tau \subseteq \sigma) \pi_e(\langle \tau \rangle) = 1\} \) 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 \( \Pi^0_1 \) classes.

**Numbering 2: Total Computable Functions**

Since the complexity of the set \( \text{Tot} \) of indices for total computable functions is \( \Pi^0_2 \), any numbering \( \omega \to \text{Tot} \) must naturally be non-effective. We include such a result as such an example.

Let \( \Lambda = \{e \in \text{Tot} \mid T_e = \{\sigma: \phi_e(\sigma) = 1\} \) is a tree\( \} \). By part (a), \( \Lambda \subseteq \omega \) is an indexing of all \( \Pi^0_2 \) classes. To obtain a numbering, we will define a map on all of \( \omega \) by defining the mapping on \( \Lambda \).

We consider the method of proving (a) \( \rightarrow \) (b) in Proposition 1. One can show that if \( P = \langle T_e \rangle \) with computable \( T_e \) then \( [T_e] = [S_e] \) with primitive recursive

\[ S_e = \{\sigma : (\forall n < |\tau|) \neg \phi_e(\sigma \uparrow n) = 0\} \]

Now consider the following proposition.

**Proposition 2.** [2, p. 9]

(i) There exists a primitive recursive function \( \pi \) such that, for each \( e \), \( U_e = T_{\pi(e)} \).

(ii) There is a primitive recursive function \( \pi \) such that, for each \( e \), \( U_e = T_{\pi(e)} \) is a tree.

The following is a numbering of the \( \Pi^0_1 \) classes based on an indexing of trees

\[ T_e = \{\sigma: \phi_e(\sigma) = 1\} \]

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

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

In [1], Cenzer describes two other methods of enumerating the \( \Pi^0_1 \) classes.

**Numbering 3: Computably Enumerable Sets**

Utilizing part (d) of Proposition 1,

\[ P_e = \omega^\omega \setminus \bigcup_{n \in \omega} I(\sigma_n) \]

gives an enumeration of the \( \Pi^0_1 \) classes. We officially denote it by

\[ e \mapsto \{\sigma : \forall m, s \exists \phi_{e,s}(m) \} \]

**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

\[ e \mapsto \{\sigma : (\forall \tau \subseteq \sigma)(\tau) \notin W_e,|\sigma|\} \]
We call this the standard numbering of the $\Pi^0_1$ 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 $\psi: \omega \rightarrow \{\text{class of all } \Pi^0_1 \text{ trees}\}$ given by $e \mapsto \{\sigma: (\forall n)\phi_{e,s}(n) \uparrow\}$. From part (d) of the proposition, $\psi(n)$ codes a $\Pi^0_1$ class for all $n$. To show that $\text{Im}(\psi)$ codes all $\Pi^0_1$ classes, let $\varphi$ be any numbering of the $\Pi^0_1$ classes given by trees. We show that there is a computable function $g$ such that $\varphi = \psi \circ g$. For all $n$ let $\phi_{g(e)}(n)$ be defined only if $\sigma \notin \varphi(e)$. Then $\sigma \in (\psi \circ g)(e) \iff \phi_{g(e)}(g(e)) \uparrow \iff \sigma \in \varphi(e)$.

**Numbering 6: Universal $\Pi^0_1$ Relation**

There is a universal $\Pi^0_1$ relation $U \subseteq \omega \times 2^\omega$ such that if $D(x)$ is a $\Pi^0_1$ 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 $\Pi^0_1$ 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 $v$ and a computable functional $\Phi_v(e)$ such that $R(n, e, x) \iff \Phi_v(e)(n) = 1$ and $\neg R(n, e, x) \iff \Phi_v(e)(n) = 0$. Define the tree $S_v(e) = \{\sigma: (\forall n < |\sigma|)\Phi_v(e)(n) = 1\}$. Then $\{x: U(e, x)\} = [S_v(e)]$ and we obtain the numbering $e \mapsto S_v(e)$.

We used each part of Proposition 1 to give different numberings for the $\Pi^0_1$ 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 $\Pi^0_1$ 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 Effectively Closed Sets

In this section we consider the notion of acceptably equivalent numberings of $\Pi^0_1$ 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 $\varphi$ of the partial computable functions $\{\phi_e\}_{e \in \omega}$ which represents an effective listing of all Turing programs. Let $\psi$ be any acceptably equivalent numbering. Then there is a computable permutation $p$ of $\omega$ 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 Isomorphism Theorem.

**Proposition 4.** Let $\omega = U_n A_n = U_n B_n$ 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 $\Pi^0_1$ classes.

**Theorem 5.** Let $\varphi$ be the standard numbering of the $\Pi^0_1$ classes. Let $\psi$ be any acceptably equivalent numbering. Then there is a computable permutation $p$ of $\omega$ such that $\varphi = \psi \circ p$.

**Proof:** Recall that $\varphi$ is represented by $e \mapsto P_e = \{\sigma: (\forall \tau \subseteq \sigma)(\tau) \notin W_{\varphi, |\sigma|}\}$. We shall represent $\psi$ by $e \mapsto Q_e$. Since $\varphi$ and $\psi$ are acceptably equivalent there are total computable functions $f$ and $g$ such that for all $x$, $P_{f(e)} = Q_e$ and...
Let\( k_0 = 0 \) and let \( k_n \) be the least \( a \) such that \( P_{k_0} \neq P_{k_n}(\forall m < n) \). Define \( G_n = \{ e : P_e = P_{k_n} \} \) and \( H_n = \{ e : Q_e = P_{k_n} \} \). Then \( \omega = \bigcup_{n \in \omega} G_n \cup \bigcup_{n \in \omega} H_n \) 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 \( f_1 \):**

\[ f \text{ satisfies } P_f(e) = Q_e \text{ and } f(H_n) \subseteq G_n. \]

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 \( \Pi^0_1 \) classes. 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:

\[
P(e,i,\sigma) \iff \begin{cases}
\sigma \in T_e & \text{or} \\
(\sigma \subseteq 0^{a_0+1}_i 1^{a_1+1}_i 0^{a_2+1}_i \ldots E^\sigma_{i+1} \text{ and } i \not\in A_{\sigma^i})
\end{cases}
\]

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

\[
\begin{align*}
\sigma &\in T_{k(e,i)} \iff P(e,i,\sigma) \text{ AND } i \not\in B_{\sigma^i} \\
\sigma &\in T_{l(e,i)} \iff P(e,i,\sigma)
\end{align*}
\]

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

\[
P_{k(e,i)} = \begin{cases}
\emptyset & \text{if } i \in A \\
P_e \cup \{0^{a_0+1}_i 1^{a_1+1}_i 0^{a_2+1}_i \ldots \} & \text{otherwise}
\end{cases}
\]

\[
P_{l(e,i)} = \begin{cases}
P_e & \text{if } i \in A \\
P_e \cup \{0^{a_0+1}_i 1^{a_1+1}_i 0^{a_2+1}_i \ldots \} & \text{otherwise}
\end{cases}
\]

Let \( C_e = \{ (k,e) : i \in A \} \) and \( D_e = \{ (l,e) : i \in A \} \). 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 \( n \),

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

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

\[
\{ 0^{a_0+1}_i 1^{a_1+1}_i 0^{a_2+1}_i \ldots \} \not\in P_e
\]

so that \( P_e = P_e \cup \{ 0^{a_0+1}_i 1^{a_1+1}_i 0^{a_2+1}_i \ldots \} \). By a similar argument to that above,

\[g(D_e) \text{ is infinite. QED}\]

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 \( \Pi^0_1 \) classes.

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

1. **Primitive Recursive Functions**
   \[
   e \mapsto (\emptyset) \cup \{ \sigma : (\forall \tau \in \sigma) \pi_e(\langle \tau \rangle) = 1 \}
   \]
2. **Total Computable Functions**
   \[
   e \mapsto \begin{cases}
   T_e & \text{if } \phi_e \text{ is total} \\
   \{ \sigma : \phi_e(\sigma) = 1 \} & \text{is a tree}
   \end{cases}
   \]
3. **Computably Enumerable (C.E.) Sets**
   \[
   e \mapsto \{ \sigma : \forall \langle m,s \rangle [\phi_{e,s}(m) \downarrow \Rightarrow \sigma \sqsubseteq \sigma_{\phi_{e,s}(m)}] \}
   \]
(4) **C.E. Sets (Partial Recursive Version)**
\[ e \mapsto \{ \sigma : (\forall \tau \subseteq \sigma) \tau \notin W_{e, \sigma} \} \]

(5) **The Halting Problem**
\[ e \mapsto \{ \sigma : (\forall s) \phi_{e,s}(e) \} \]

(6) **Universal \( \Pi^0_2 \) Relation**
\[ e \mapsto \{ x \in U(e, x) \} \]

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

**Proof:** We use the notation (i) \( \rightarrow \) (j) to mean that if \( \varphi \) and \( \psi \) are the corresponding numberings for (i) and (j) respectively, then there is a total \( \varphi \)-computable function \( f \) such that \( \varphi = \psi \circ f \). We show that (i) \( \leftrightarrow \) (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 \( \Pi^0_2 \). 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) \( \rightarrow \) ((1), (4)) \( \rightarrow \) (6) \( \rightarrow \) ((3), (5)) \( \rightarrow \) (2). To obtain the result for \( i \neq 2 \) we also show ((3), (5)) \( \rightarrow \) ((1), (4)).

(2) \( \rightarrow \) (1), (4). Let \( \varphi \), \( \psi \), and \( \gamma \) 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
\[ S_e = \{ \sigma : (\forall \tau \subseteq \sigma) \sim \phi_{\delta(e),[\tau]}(\tau) = 0 \} \]
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 \( n, x \uparrow n \notin S_e \). So there is some \( m \) such that \( \phi_{\delta(e),m}(x \uparrow n) = 0 \). Then for any \( k > \max\{m, n\} \), we have that \( x \uparrow k \notin \varphi(e) \). It follows that \( x \notin \varphi(e) \).

Now use the \( S_n^m \) Theorem to get a \( \Delta^0_2 \)-function \( g \) such that \( \pi_{\delta(e),[\tau]}(\tau) = 0 \) \( \Leftrightarrow \) \( (\forall \tau \subseteq \sigma) \sim \phi_{\delta(e),[\tau]}(\tau) = 0 \). Then \( \varphi = \psi \circ g \). We also have that \( \varphi = \gamma \circ \delta \).

(1), (4) \( \rightarrow \) (6). Let \( \varphi, \psi, \) and \( \gamma \) be the numberings for (1), (4), and (6) respectively. Define the relation \( R_{\varphi} \) by \( R_{\varphi}(n, e, x) \Leftrightarrow x \uparrow n \in \varphi(e) \). Let \( f_{\varphi} \) 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 \( R_{\varphi} \) and \( f_{\psi} \) similarly we obtain \( \psi = \gamma \circ f_{\psi} \).

(6) \( \rightarrow \) (3), (5). We obtained (6) \( \rightarrow \) (5) in discussing Numbering (5). Now let \( \varphi \) and \( \psi \) be numberings for (6) and (3) respectively. Define \( \phi_{\delta(e)}(\sigma) \)
\[ = \begin{cases} 1 & \text{if } \exists (n, s)(n < |\sigma| \land \Phi_{\varphi(n),s}(n) = 0) \\ \uparrow & \text{otherwise} \end{cases} \]
Then \( \varphi = \psi \circ (g \circ \nu) \).

(3), (5) \( \rightarrow \) (1), (2), (4). Let \( \varphi, \psi, \gamma, \zeta, \) and \( \sigma \) 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 \( R_{\varphi} \) a recursive relation. Define the computable tree
\[ T_{\varphi(e)} = \{ \sigma : (\forall n, m \leq |\sigma|)R_{\varphi}(m, e, \sigma \uparrow n) \} \]
Define \( T_{\varphi(e)} \) similarly utilizing the recursive relation \( R_{\varphi} \). Then \( \varphi = \gamma \circ f \) and \( \psi = \gamma \circ g \).

Now utilize the methods of (2) \( \rightarrow \) (1), (4) with \( T_{\varphi(e)} \), \( T_{\varphi(e)} \) in place of \( T_{\delta(e)} \) to obtain computable \( f', g' \) such that \( \varphi = \zeta \circ f' \) and \( \psi = \zeta \circ g' \); 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 semilattice \( \mathcal{L}(\mathcal{P}) \), where \( \mathcal{P} \) is the class of all \( \Pi^0_1 \) classes. We already have a nice example of an element occurring in a minimal element if this semilattice, namely an enumeration of all \( \Pi^0_1 \) 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 \( \Pi^0_1 \) classes occurring in enumerations of \( \Pi^0_1 \) classes.

4 Decidable Effectively Closed Sets in Numberings

A \( \Pi^0_1 \) class may be represented by many different computable trees. However decidable \( \Pi^0_1 \) 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 $\Pi^0_1$ classes necessarily contains every decidable $\Pi^0_1$ 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 $\Pi^0_1$ class in an effective enumeration of uniformly computable trees giving rise to all $\Pi^0_1$ classes. As a result, index sets for decidable $\Pi^0_1$ 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 revisited. We generalize the enumeration results to subfamilies of $\Pi^0_1$ classes and to trees with $\leq n$ dead ends. We devote the rest of this paper towards proving these results.

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

Clearly a clopen tree $T$ has no dead ends. Moreover a $\Pi^0_1$ class $[T] \subseteq 2^{\omega}$ is clopen if $2^{\omega}\setminus[T]$ 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 $\Pi^0_1$ 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 enumeration.

**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 $\Pi^0_1$ 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 $T(S_0, k_1), ..., T(S_0, k_i)$ ($k_i \in \{ m, n \}$) 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 $S_0 = T(S_0, n) \cap \{0,1\}^{2^0}$.

**Stage j+1.** $S_0, ..., S_j$ have already been constructed up to level $2^j$ and are already following trees $T(S_0, k_j), ..., T(S_j, k_j)$. 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 $T(S_0, k_{j+1}), ..., T(S_j, k_{j+1})$ 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 $T(S_i, m) = \{ \sigma: \tau \subseteq \sigma \text{ or } \sigma \subseteq \tau \text{ for some } \tau \in T(S_i, n) \text{ of length } 2^j \}$. 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 comple-
etes 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 \( \rightarrow \exists c T_i = S_c) \)

(iii) \( i \neq j \rightarrow S_i \neq S_j \)

**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), \( k_i = k_{i+1} = n \) 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 \( \lim_j T(S_i, k_j) \downarrow \) so that \( \lim_j T(S_i, k_j) \) 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 \( n \), \( S_i \cap \{0,1\}^{\leq n} = T(S_i, n) \cap \{0,1\}^{\leq n} \) 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 construction \( T_i = S_c \). If not, let \( \ell \) equal the least \( k \) such that \( T_k = T_i \). Let \( \ell \) be large enough so that \( T_i \) differs from \( T_{\ell} \) at level \( 2^j \) for all \( e < \ell \). If at stage \( j \) there already exists a \( c \) such that \( T_i = T(S_c, n) \), then clearly \( T_i = S_c \). Otherwise, by Action (2), some tree \( S_c \) follows \( T_i \) by no later than stage \( j + \ell \).

**Verification of (iii).** By Action (2), \( S_i \) is distinct from all \( S_j \) \((j < i)\) at level \( 2^j \) and from all \( S_j \) \((j > i)\) at level \( 2^i \); \( S_i \neq S_j \) 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 \( S_0, S_1, S_2, \ldots \) 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 \( \{[T_e]\}_{e \in \omega} \) be the standard enumeration of the \( \Pi_1^0 \) classes. Then there is a decidable \( \Pi_1^0 \) 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 \( \text{Ext}(P_e) = \{\sigma: (\forall \tau \in T_e) (\forall n) (\exists \tau \in [0,1]^n) \sigma \upharpoonright \tau \in T_e\} \) where \( \upharpoonright \) denotes concatenation. By König’s Lemma, since the trees are subsets of \( 2^{\omega} \), this set is \( \Pi_1^0 \). Therefore \( \{e: T_e \) has no dead ends\} \( = \{e: T_e = \text{Ext}(P_e)\} \) is \( \Pi_1^0 \). However, \( \{e: P_e \) is decidable\} \( = \{e: T_e = [T] \) for some computable \( T \) without dead ends\} \( = \{e: (\exists \alpha) \phi_\alpha \) is a characteristic function for \( \text{Ext}(P_e)\} \).

Therefore this latter set is \( \Sigma_2^0 \). 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 \( \Pi_1^0 \) class and \( T \) has \( \leq n \) dead ends\} \( \) and \( \mathcal{R}_k = \{T \in \mathcal{P}_k \) classes by uniformly computable trees, there is a \( \Pi_1^0 \) 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 \( \Pi^0_1 \) classes as well as the complexity of index sets for decidable \( \Pi^0_1 \) classes.

References: