 Electronic Notes in Theoretical Computer Science 167 (2007) 289–301 

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

Enumerations of Π0

1

Classes:

Acceptability and Decidable Classes

Paul Brodhead[1](#_bookmark0) ,[2](#_bookmark0)

*Department of Mathematics University of Florida Gainesville, U.S.A.*

Abstract

A Π0 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 Π0 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

1

1

Π0 classes are acceptably equivalent. We also consider decidable Π0 classes in enumerations. A

1 1

decidable Π0 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

1

to all Π0 classes. In fact this is guaranteed to occur for some decidable Π0 class. These results are

1 1

motivated by structural questions concerning the upper semilattice of enumerations of Π0 classes

1

where notions such as acceptable equivalence arise.

*Keywords:* Computability, Π0 Classes, Enumerations.

1

# Introduction

Many results in classical computability theory are derived from a study of the indices of partial computable functions. For example, the Enumeration The- orem allows indices to be treated as arguments. Conversely, the *Sm* 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.

*n*

1 The author wishes to thank Douglas Cenzer for his example and helpful conversations.

2 Email: [brodhead@math.ufl.edu](mailto:brodhead@math.ufl.edu)

1571-0661 © 2007 Elsevier B.V. Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

doi:10.1016/j.entcs.2006.08.017

It is known that if a system of indices is *acceptable* then it has same struc- ture theory as any system that satisfies the Enumeration and *Sm* theorems. A system of indices *φ* is a family of surjective maps *φn* : *ω* → {*n*-ary partial recursive functions} [[12](#_bookmark15)]. Let *φ* be a system of indices that satisfies the Enu- meration and *Sm* theorems and call it the *standard system* [[9](#_bookmark12)]. A system of indices *ψ* is *acceptable* if, for every *n*, there are total computable functions *f*

*n*

*n*

and *g* such that *ψn* *φn* and *φn* *ψn*

[[11](#_bookmark14)]. For a greater treatment on

*e f* (*e*) *e g*(*e*)

acceptable systems of indices for partial recursive functions, see [[9](#_bookmark12)]. In this

paper we develop a notion of acceptability for Π0 classes.

1

A Π0 class is an effectively closed set of reals in *ωω*, although we shall

1

restrict our attention to classes in 2*ω*. Alternatively we may also consider a

Π0 class to be the set of infinite paths in through a computable tree in *ω<ω*.

1

One way to enumerate them is *Pe* = *ωω* \ *n*∈*W I*(*σn*) [[1](#_bookmark5)]. (Here *We* is the

e

*eth* c.e. set in the standard system, *σn* is the *nth* string in the enumeration

*σ*0*, σ*1*, σ*2*,...* of all strings in *ω<ω*, and *I*(*σn*) is the set of elements in *ωω* that

extend *σn*.) As a result, Π0

1

classes have index-argument related properties

inherited from the Enumeration and *Sm* theorems. We shall use an alternate

*n*

enumeration method which takes advantage of this property and justifiably call this the standard numbering of Π0 classes.

1

Our work follows in the path of previous work done by Jockusch, Rogers, and Soare [[13](#_bookmark16), p 25] for acceptably equivalent numberings of the partial re- cursive functions, and hence of the c.e. sets. A permutation exists between any two acceptably equivalent numberings which preserves the original com- putable content. We use the standard numbering for c.e. sets to extend this result to Π0 classes. Furthermore we show that the most frequently used num- berings in Π0 classes are acceptable with respect to the standard numbering. We develop these notions below.

1

1

In the mid-1950s, initiated under Kolmogorov, work began on general- ized theory of numberings and continued under the direction of Mal’tsev and Ershov [[4](#_bookmark7)]. A *numbering* of a collection *C* of objects is a surjective map *F* : *ω* → *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 *ν* ≤ *u*, iff there is a total computable function *f* such that *ν* = *u* ◦ *f* . Then *u* is *acceptable* if it acceptable with respect to all numberings. We say that *ν* and *u* are *acceptably equivalent*, denoted *ν* ≡ *u*, iff *ν* ≤ *u* and *u* ≤ *ν*. Note that ≡ is an equivalence relation and let L(*C*) denote the set of all num- berings of *C* modulo ≡. It is easy to verify that L(*C*) is an upper semilattice under ≤. Furthermore enumerations without repetition occur only in the min- imal 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](#_bookmark8)] showed that an enumeration of the c.e. sets exists without repetition. Goncharov, Lempp, and Solomon [[6](#_bookmark9)] further generalized this result for *n*-c.e. sets. An interesting result by Suzuki [[14](#_bookmark17)] shows that an enumeration of the computable sets exists without repetition. However our goal is a set of corresponding results for Π0 classes.

1

Recently Raichev [[10](#_bookmark13)] proved that an enumeration of the Π0 classes exists without repetition. Using a modification of the Friedberg’s proof for c.e. sets, he gives an enumeration of the Σ0 sets without repetiton. The corresponding result related to the Goncharov-Lempp-Solomon theorem concerning differ-

1

1

ences of Σ0

1

classes remains unsolved. Concerning the Suzuki theorem, we

turn to decidable Π0 classes.

1

A decidable Π0 class is the set of infinite paths through a computable tree

1

*without dead ends*. In one way, decidable Π0

1

classes resemble the recursive

sets in the same way that Π0 classes resemble the c.e. sets. In c.e. sets it is

1

unknown immediately whether an element will show up in an enumeration.

In Π0

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

Π0 classes (given the proper representation) such things are known. Given the result of Suzuki, it seems plausible that decidable Π0 classes can be enumerated

1

1

without repetition.

To show such an enumeration exits it is natural to follow Friedburg’s ap- proach, utilized by Odifreddi, Goncharov, Lempp, Solomon, Raichev, and oth- ers. 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 Π0 classes, the proof appears to succeed. However diagonalization immediately provides for a computable tree without dead ends not in the enumeration. Therefore although a decid-

1

able Π0

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 Π0 classes. For some decidable Π0 class this is guaranteed

1 1

to happen. Subsequently complexity results for index sets for decidable Π0

1

classes and for computable trees without dead ends are distinct. We note that

the results on index sets for decidable Π0

1

classes in [[2](#_bookmark6)] use the convention

that a class *Pe* is decidable iff the corresponding tree *Te* has no dead ends. In

light of our new theorem, those results need to be revisited. We generalize these enumeration results to subfamilies of Π0 classes and to trees with ≤ *n* dead ends, for fixed *n*. It remains open whether decidable Π0 classes may be

1

1

enumerated without repetition.

# Enumerations of Π0

1

Classes

In this section, present some basic notation and facts about Π0 classes which lead to different methods of enumerating them. Finally we present six different enumerations of them.

1

Basic Notations; Facts about Π0 classes

1

The partial computable {0*,* 1}–valued functions are indexed as {*φe*}*e*∈*ω* and the primitive recursive functions as {*π**e*}*e*∈*ω*. As usual *φe,s* denotes that portion of function *φe* defined by stage *s*. We use *φe*(*x*)↓ to mean that *φe* is defined on input *x*. Similarly *φe*(*x*)↑ signifies that the function is undefined. We shall use *σ* and *τ* to represent strings in *ω<ω*. Let ⟨*τ* ⟩∈ *ω* denote the usual code for a finite string. Recall that *T* ⊆ *ω<ω* 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 Π0 *class* iff *P* = [*T* ] for some computable tree *T* . We have the following result from [[2](#_bookmark6)]:

1

Proposition 2.1 *For any class P* ⊂ *ωω, the following are equivalent:*

1. *P* = [*T* ] *for some computable tree T* ⊂ *ω<ω;*
2. *P* = [*T* ] *for some primitive recursive tree T;*
3. *P* = {*x* : (∀*n*)*R*(*n, x*)}*, for some computable relation R;*
4. *P* = [*T* ] *for some* Π0 *tree T* ⊂ *ω<ω;*

1

Following this proposition, Cenzer and Remmel mention two possible num- berings of the Π0 classes that occur as a consequence. We develop these con-

1

cepts here.

Numbering 1: Primitive Recursive Functions

For each *e*, *Ue* = {∅} ∪ {*σ* : (∀*τ* ± *σ*) *πe*(⟨*τ* ⟩) = 1} defines a primitive recursive tree. To see that this enumeration contains all primitive recursive trees, observe that if {*σ* : *πe*(*σ*) = 1} is a tree then *Ue* is that tree. By part (b), *e* '→ *Ue* is a tree enumeration of the Π0 classes.

1

Numbering 2: Total Computable Functions

Since the complexity of the set Tot of indices for total computable functions is Π0, any numbering *ω* → Tot must naturally be non-effective. We include such a result as such an example.

2

Let Λ = {*e* : *e* ∈ Tot and *Te* = {*σ* : *φe*(*σ*) = 1} is a tree}. By part (a), Λ ⊆ *ω* is an indexing of all Π0 classes. To obtain a numbering, we will define a map on all of *ω* by defining the mapping on Λ. We consider the method of proving (*i*) → (*ii*) in Theorem 2.1. One can show that if *P* = [*Te*] with computable *Te* then [*Te*] = [*Se*] with primitive recursive *Se* = {*σ* : (∀*n <*

1

|*τ* |)¬*φe,*|*τ*|(*τ*T*n*)= 0}. Now consider the following proposition.

Proposition 2.2 ([[2](#_bookmark6), p 9])

* 1. *There exists a primitive recursive function φ such that if φe deﬁnes a computable tree Te then Se* = *Uφ*(*e*)*. So* [*Te*]= [*Uφ*(*e*)]*.*
  2. *There is a primitive recursive function π such that, for each e, Ue* = *Tπ*(*e*)*.*

The following is a numbering of the Π0

1

classes based on an indexing of

trees *Te* = {*σ* : *φe*(*σ*) = 1} arising solely from the total {0*,* 1}–computable

functions in {*φe*}*e*∈*ω*.

*e* '→ ⎨ *Te* if *φe* is total and *Te* is a tree

⎧

⎩ *Tπ*(*φ*(*e*)) otherwise

In [[1](#_bookmark5)], Cenzer describes two other methods of enumerating the Π0 classes.

1

Numbering 3: Computably Enumerable Sets

Utilizing part (d) of [2.1](#_bookmark1), *Pe* = *ωω* \ *n*∈*W I*(*σn*) gives an enumeration of

e

the Π0 classes. We officially denote it by *e* '→ {*σ* : ∀⟨*m, s*⟩ [*φe,s*(*m*)↓ ⇒ *σ* /±

1

*σφ*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* '→ {*σ* : (∀*τ* ± *σ*) ⟨*τ* ⟩ /∈ *We,*|*σ*|}. We call this the standard numbering of the Π0 classes.

1

Another method commonly found in the literature (see [[8](#_bookmark11)], for example) uti- lizes a version of Halting Problem concerned with diagonal computation with oracles.

Numbering 5: The Halting Problem

Consider the mapping *ψ* : *ω* → {class of all Π0 trees} given by *e* '→ {*σ* :

1

(∀*s*)*φσ* (*e*)↑}. From part (d) of the theorem, *ψ*(*n*) codes a Π0 class for all *n*.

*e,s*

1

To show that Im(*ψ*) codes all Π0 classes, let *ϕ* be any numbering of the Π0

1 1

classes given by trees. We show that there is a computable function *g* such

that *ϕ* = *ψ* ◦ *g*. For all *n* let *φσ* (*n*) be defined only if *σ* /∈ *ϕ*(*e*). Then

*g*(*e*)

*σ* ∈ (*ψ* ◦ *g*)(*e*) ⇐⇒ *φσ* (*g*(*e*))↑ ⇐⇒ *σ* ∈ *ϕ*(*e*).

*g*(*e*)

Numbering 6: Universal Π0 Relation

1

There is a universal Π0 relation *U* ⊆ *ω* × 2*ω* such that if *D*(*x*) is a Π0

1

1

relation then there is an *e* ∈ *ω* such that *D*(*x*) ↔ *U* (*e, x*) [[7](#_bookmark10), p 73]. Therefore

by part (c), *e* '→ {*x* : *U* (*e, x*)} is a numbering of the Π0 classes.

1

We may obtain a tree numbering as follows. Suppose that *U* (*e, x*) = (∀*n*)*R*(*n, e, x*) where *R* is a computable relation. There is a computable func-

tion *ν* and a computable functional Φ*ν*(*e*) such that *R*(*n, e, x*) ⇐⇒ Φ*x*

*ν*(*e*)

(*n*) =

1 and ¬*R*(*n, e, x*) ⇐⇒ Φ*x* (*n*) = 0. Define the tree *Sν*(*e*) = {*σ* : (∀*n <*

*ν*(*e*)

|*σ*|) Φ*σ* (*n*) = 1}. Then {*x* : *U* (*e, x*)} = [*Sν*(*e*)] and we obtain the numbering

*ν*(*e*)

*e* '→ *Sν*(*e*).

We used each part of Theorem [2.1](#_bookmark1) to give different numberings for the Π0 Classes. Numbering 2 has the distinct feature of being non-effective. Collec- tively, however, each shared the common feature that they could ultimately be considered numberings of trees. This is due to the very definition of a Π0 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.

1

1

# Acceptable Enumerations of Π0

1

Classes

In this section we consider the notion of acceptably equivalent numberings of Π0 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:

1

Theorem 3.1 ([[13](#_bookmark16), p 25]) *Consider the standard numbering ϕ of the partial* *computable functions* {*φe*}*e*∈*ω 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 ϕ* = *ψ* ◦ *p.*

The proof is similar to our result in Theorem [3.3](#_bookmark3). It uses the following proposition, also found in [[13](#_bookmark16), p 25], whose proof utilizes the same construction used to prove the Myhill Isomorphism Theorem.

Proposition 3.2 *Let ω* = *n An* = *n Bn where the sequences* {*An*}*n*∈*ω and*

{*Bn*}*n*∈*ω are each pairwise disjoint. Let f and g be injective computable func- tions such that f* (*An*) ⊆ *Bn and g*(*Bn*) ⊆ *An for all n. Then there is a computable permutation p such that p*(*An*)= *Bn 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 Π0 classes.

1

Theorem 3.3 *Let ϕ be the standard numbering of the* Π0 *classes. Let ψ be*

1

*any acceptably equivalent numbering. Then there is a computable permutation*

*p of ω such that ϕ* = *ψ* ◦ *p.*

Proof. Recall that *ϕ* is represented by *e* '→ *Pe* = {*σ* : (∀*τ* ± *σ*) ⟨*τ* ⟩ /∈ *We,*|*σ*|}. We shall represent *ψ* by *e* '→ *Qe*. Since *ϕ* and *ψ* are acceptably equivalent there are total computable functions *f* and *g* such that for all *x*, *Pf*(*e*) = *Qe* and *Qg*(*e*) = *Pe*. Let *k*0 = 0 and let *kn* = least *a* s.t. *Pa* /= *Pk*m (∀*m < n*). Define *Gn* = {*e* : *Pe* = *Pk*n } and *Hn* = {*e* : *Qe* = *Pk*n }. Then *ω* = *n Gn* = *n Hn* and the sequences {*Gn*}*n*∈*ω* and {*Hn*}*n*∈*ω* are each pairwise disjoint. Furthermore *f* (*Hn*) ⊆ *Gn* and *g*(*Gn*) ⊆ *Hn*. To complete the proof it suffices by Proposition [3.2](#_bookmark2) 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 *Pf*(*e*) = *Qe* and *f* (*Hn*) ⊆ *Gn*. 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 *Wa* = *Wh*(*i,a*) for all *i* and *a*, and if *i* /= *j* then *h*(*i, a*) /= *h*(*j, b*) for any *a* or

*b*. Let *f*1(*e*) = *h*(*e, f* (*e*)). Then *f*1 satisfies *Pf*1(*e*) = *Qe* and *f*1(*Hn*) ⊆ *Gn*. 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 *Se* of indices such that for each *i* ∈ *Se* we have that *Qi* = *Qg*(*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*, ...* be an enumeration of *A* without repetition. Let *An* and *Bn* de- note the sets *A* and *B*, respectively, up to stage n. Also let *T*0*, T*1*, T*2*, ...* be

a tree enumeration of the Π0 classes. For any *σ* ∈ *ωω*, let *Eσ* = 1 if |*σ*|

1

is even and 0 otherwise. Now let *e, i* ∈ *ω*. Consider the computable rela-

*a*|σ|+1

tion *P* (*e, i, σ*) which holds iff *σ* ∈ *Te* or (*σ* ± 0*a*0 +11*a*1 +10*a*2 +1*...E* and *i* /∈

*σ*

*A*|*σ*|). Define the computable trees *Tk*(*e,i*) = {*σ* : *P* (*e, i, σ*) and *i* /∈ *B*|*σ*|}

and *Tl*(*e,i*) = {*σ* : *P* (*e, i, σ*)}. It follows that *Pk*(*e,i*) =

⎧⎪⎪⎨

*Pe* if *i* ∈ *A*

∅ if *i* ∈ *B* and

⎪⎪⎩ *Pe* ∪ {0*a*0 +11*a*1 +1*...*} otherwise

*Pl*(*e,i*)

= ⎧⎨ *Pe* if *i* ∈ *A*

⎩ *Pe* ∪ {0*a*0 +11*a*1 +1*...*} otherwise

Let *Ce* = {*k*(*e, i*): *i* ∈ *A*} and *De* = {*l*(*e, i*): *i* ∈ *A*}. We claim that for each

*e*, *Se* = *g*(*Ce*) ∪ *g*(*De*) is infinite, thereby completing the proof. To show this,

we shall prove that either *g*(*Ce*) or *g*(*De*) is infinite.

Case I: (*Pe* /= ∅). It follows that for some distinct *m* and *n*, that {*k*(*e, i*) : *i* ∈ *A*} ⊆ {*a* : *Pa* = *Pe*} ⊆ *Gn* and {*k*(*e, i*) : *y* ∈ *B*} ⊆ {*a* : *Pa* = ∅} ⊆ *Gm* are disjoint. Now since *Qg*(*e*) = *Pe* for all *e*, after applying *g* to each set the new sets remain disjoint. If *g*(*Ce*) is finite, say *g*(*Ce*) = {*c*1*, c*2*, ..., cl*}, then

' '

*Ce* = {*i* ∈ *ω* : *g*(*k*(*e, i*)) ∈ {*c*1*, c*2*, ..., cl*}} is computable and *A* ⊆ *Ce* and

'

*B* ∩ *Ce* = ∅, contrary to *A* and *B* being computably inseparable. Therefore

*g*(*Ce*) is infinite.

Case II: (*Pe* = ∅). It follows that {0*a*0+11*a*1 +1*...*} /∈ *Pe* so that *Pe* /= *Pe* ∪

{0*a*0+11*a*1+1*...*}. By a similar argument to that above, *g*(*De*) is infinite.

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* '→ *Te* to denote a specific tree numbering of the Π0 classes.

1

Theorem 3.4 *In the notation of the previous section, each of the following is a numbering of the* Π0 *classes:*

1

1. *Prim. Rec. Functions e* '→ {∅} ∪ {*σ* : (∀*τ* ± *σ*) *πe*(⟨*τ* ⟩)= 1}

⎧⎪ *Te if φe is total & Te* =

1. *Total Comp. Functions e* '→ ⎪⎨

{*σ* : *φe*(*σ*)= 1} *is a tree*

⎪⎪⎩ *Tπ*(*φ*(*e*)) *otherwise*

1. *Comp. Enum. Sets e* '→ {*σ* : ∀⟨*m, s*⟩[*φe,s*(*m*)↓ ⇒ *σ* /± *σφ*e,s(*m*)]}
2. *C.E. Sets* (*P.R. Vers.*) *e* '→ {*σ* : (∀*τ* ± *σ*) ⟨*τ* ⟩ /∈ *We,*|*σ*|}
3. *The Halting Problem e* '→ {*σ* : (∀*s*) *φσ*

*e,s*

1. ↑}
2. *Universal* Π0 *Relation e* '→ {*x* : *U* (*e, x*)}

1

*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 ϕ* = *ψ* ◦ *p. The permutation is* Δ0 *if either ϕ or ψ is the numbering given in (2). Otherwise the permutation is computable.*

3

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 *ϕ* = *ψ* ◦ *f* . We show that (*i*) ↔ (*j*) for *i* /= *j*. Then by Theorem [3.3](#_bookmark3) we have our result for *i, j* /= 2. However the same proof given in that theorem demonstrates that if *i* = 2 then the permutation is Π0. Our proof closely models the proof, as given in [[2](#_bookmark6)], of Theorem [2.1](#_bookmark1). Note that according to this theorem, (2) is of form (a), ((1), (4)) are of form

2

(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* /= 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 *δ*(*e*) denote the index of the tree *ϕ*(*e*)= *Tδ*(*e*). For each *e* ∈ *ω*, define the primitive recursive tree *Se* = {*σ* : (∀*τ* ± *σ*)¬*φδ*(*e*)*,*|*σ*|(⟨*τ* ⟩)= 0}.

We show that [*ϕ*(*e*)] = [*Se*]. Now *Se* ⊆ *ϕ*(*e*), so that [*Se*] ⊆ [*ϕ*(*e*)]. Now suppose that *x* /∈ [*Se*]. Then for some *n*, *x*T*n* /∈ *Se*. So there is some *m* such that *φδ*(*e*)*,m*(*x*T*n*)= 0. Then for any *k > max*{*m, n*}, we have that *x*T*k* /∈ *ϕ*(*e*). It follows that *x* /∈ *ϕ*(*e*).

Now use the *Sm* Theorem to get a Δ0–function *g* such that *πg*(*e*)(⟨*σ*⟩) =

*n* 3

1 ⇐⇒ (∀*τ* ± *σ*)¬*φδ*(*e*)*,*|*σ*|(⟨*τ* ⟩) = 0. Then *ϕ* = *ψ* ◦ *g*. We also have that

*ϕ* = *γ* ◦ *δ*.

(1), (4) → (6). Let *ϕ, ψ,* and *γ* be the numberings for (1), (4), and (6) respectively. Define the relation *Rϕ* by *Rϕ*(*n, e, x*) ⇐⇒ *x*T*n* ∈ *ϕ*(*e*). Let *fϕ* be a computable function such that (∀*n*)*R*(*n, e, x*) ⇐⇒ *U* (*fϕ*(*e*)*, x*). Then *ϕ* = *γ* ◦ *fϕ*. Defining *Rψ* and *fψ* similarily we obtain *ψ* = *γ* ◦ *fψ*.

(6) → (3), (5). We obtained (6) → (5) in discussing Numbering (5). Now let *ϕ* and *ψ* be numberings for (6) and (3) respectively. Define

⎧⎨ 1 if ∃⟨*n, s*⟩(*n <* |*σ*| & Φ*σ* (*n*)= 0)

*φg*(*ν*(*e*))(⟨*σ*⟩)=

⎩ ↑ otherwise

*ν*(*e*)*,s*

Then *ϕ* = *ψ* ◦ (*g* ◦ *ν*).

(3), (5) → (1), (2), (4). Let *ϕ, ψ, γ, ζ,* and *ι* be numberings for (3), (5),

(2), (1), and (4), respectively. We have, for all *e*, *ϕ*(*e*)= {*σ* : (∀*n*)*Rϕ*(*n, e, σ*)}

with *Rϕ* a recursive relation. Define the computable tree *Tf*(*e*) = {*σ* : (∀*m, n* ≤

|*σ*|)*Rϕ*(*m, e, σ*T*n*)}. Define *Tg*(*e*) similarily utilizing the recursive relation *Rψ*. Then *ϕ* = *γ* ◦ *f* and *ψ* = *γ* ◦ *g*.

Now utilize the methods of (2) → (1)*,* (4) with *Tf*(*e*)*, Tg*(*e*) in place of *Tδ*(*e*) to obtain *computable f* '*, g*' such that *ϕ* = *ζ* ◦ *f* ' and *ψ* = *ζ* ◦ *g*'. Note also that *ϕ* = *ι* ◦ *f* and *ψ* = *ι* ◦ *g*.

It remains open whether these enumerations only occur in the greatest

element of the semilattice L(P), where P is the class of all Π0

1

classes. We

already have a nice example of an element occuring in a minimal element if this

semilattice, namely an enumeration of all Π0 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 Π0 classes

1

1

occuring in enumerations of Π0 classes.

1

# Decidable Π0

1

Classes in Enumerations

A Π0 class may be represented by many different computable trees. However decidable Π0 classes are unique in that each decidable class *D* has a unique computable tree without dead ends that represents it. Although every enu-

1

1

meration of the Π0 classes necessarily contains every decidable Π0 class, the

1 1

unique tree without dead ends does not have to show up in the enumeration.

In fact this is guaranteed to occur for some decidable Π0 class in an effective enumeration of uniformly computable trees giving rise to all Π0 classes. As a result, index sets for decidable Π0 classes and for computable trees without dead ends are distinct both as sets and in complexity. Previous results in [[2](#_bookmark6)] make no such distinction and consequently must be revisted. We generalize

1

1

1

the enumeration results to subfamilies of Π0 classes and to trees with ≤ *n*

1

dead ends. We devote the rest of this paper towards proving these results.

Definition 4.1 A tree *T* ⊆ 2*<ω* and a set [*T* ] are clopen iff there is a nonempty finite set *S* ⊆ *ω* such that *T* = ∅ or *T* = {*σ* : *σ* ± *σi* or *σi* ± *σ* for some *i* ∈ *S*}.

Clearly a clopen tree *T* has no dead ends. Moreover a Π0 class [*T* ] ⊆ 2*ω* is clopen if 2*ω* \ [*T* ] is clopen. That is *P* = [*T* ] is clopen iff *P* is a finite union of intervals *I*(*σn*). Clopen sets will play the role for Π0 classes that finite sets

1

1

play for c.e. sets.

Theorem 4.2 *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 enumera- tion.*

Proof. Friedberg [[5](#_bookmark8)] 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*1*, T*2, ... be an effective enumeration of uniformly computable trees. Take, for example, the standard enumeration of trees corresponding to an ef- fective listing of the Π0 classes. Although we don’t require {*Te*}*e*∈*ω* to contain all clopen trees, we assume, without loss of generality, that they already con- tain them. We will construct, in stages, a sequence of *follower* trees *S*1*, S*2, ... to prove the theorem.

1

At stage *i* we will ensure that we have *i*+ 1 trees *S*0*, S*1*,... , Si*, constructed up to level 2*i*, following trees *T*(*S*0 *,k*i)*,... , T*(*S*i*,k*i) (*ki* ∈ {*m, n*}) which are each

pairwise distinct at level 2*i*. Also, at stage *i*, initially some of the *Si* will have the status of being marked (*ki* = *m*) in which case *Si* will continue to follow *T*(*S*i*,m*) forever. If not, then *Si* is not marked (*ki* = *n*) and we determine for each *i*, if *Si* should be marked. If an *Si* needs to be marked then we determine a tree *T*(*S*i*,m*) that it shall hereafter follow. Otherwise each *Si* continues to follow *T*(*S*i*,n*) and the stage is complete.

*Construction.*

Stage 0. Find the first tree *Ti*

such that *Ti*

∩ {0*,* 1}20 /= ∅, denote this tree as

*T*(*S*0*,n*)

, and define *S*0

= *T*(*S*0*,n*)

∩ {0*,* 1}≤20 .

Stage j+1. *S*0*,... , Sj* 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*,... , Sj* up to level 2*j*+1 by determining the trees *T*(*S ,k* ),

0 j+1

*...* , *T*(*S*j *,k*j+1) they shall follow, and

* 1. Construct a new tree *Sj*+1 up to level 2*j*+1

*Action* (1). Let *Uj*+1 = {(*Si, kj*) : *kj* = *n* and *T*(*S*i*,k*j) has dead ends at level 2*j*+1}. All *Si* such that (*Si, kj*) /∈ *Uj*+1 keep their status as marked or unmarked, so *kj* = *kj*+1, and continue to follow *T*(*S*i*,k*j+1). Those *Si* such that (*Si, kj*) ∈ *Uj*+1 will hereafter be marked and will now follow the tree *T*(*S*i*,m*)

given by *T*(*S ,m*) = {*σ* : *τ* ± *σ* or *σ* ± *τ* for some *τ* ∈ *T*(*S ,n*) of length 2*j*}. Note

i i

that each marked *Si* follows a clopen tree *T*(*S*i*,m*).

*Action* (2). Let (*Sj*+1*, n*) be the least *i* such that *Ti* is distinct from all

*T*(*S ,k*

) (*i* ≤ *j*) at level 2*j*+1 and such that *Ti* has no dead ends. Define

i j+1

*Sj*+1 = *T*(*S*

j+1

*,n*) ∩ {0*,* 1}≤*j*+1. This completes the construction.

*Verification.*

We now verify that:

* + 1. For each *i*, lim*jT*(*S*i*,k*j)↓ = *Si* = *Tn*i for some *Tn*i without dead ends
    2. (∀*i*)(*Ti* has no dead ends −→ (∃*c*) *Ti* = *Sc*)
    3. *i* /= *j* −→ *Si* /= *Sj*

Verification of (i). For all *j*, *kj* = *n* or *kj* = *m*. Fix *i*. By Action (2), at stage *i*, (*Si, ki*) = (*Si, n*). By Action (1), *kl* = *kl*+1 = *n* for all *l > i* if *Si* is never marked. If *Si* is marked at stage *r > i*, then for all *s* ≥ *r*, *ks* = *ks*+1 = *m*. In either case lim*j*≥*ikj*↓ so that lim*j*(*Si, kj*) converges to (*Si, n*) or (*Si, m*). If it converges to (*Si, m*) then *Si* never diverges from following the clopen tree *T*(*S*i*,m*). Otherwise *Si* 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 ,n*) is a tree without dead ends. Either way lim*jT*(*S*i*,k*j) ↓= *Tn*i for some tree *Tn*i without dead ends. Now for all *n*,

i

*Si* ∩ {0*,* 1}≤*n* = *T*(*S ,k* ) ∩ {0*,* 1}≤*n* and *T*(*S ,k* ) ⊆ *T*(*S ,k* ). Therefore *Si* =

i n

lim*j T*(*S*i*,k*j) = *Tn*i .

i n i

n+1

Verification of (ii). Let *Ti* be a tree without dead ends. There are two cases. If there is a stage *j* and a *c* such that *Ti* = *T*(*S*c*,m*) at stage *j*, then by the constrction *Ti* = *Sc*. If not, let *i* equal the least *k* such that *Tk* = *Ti*. Let *j* be large enough so that *T*b*i* differs from *Te* at level 2*j* for all *e < i*. If at stage *j* there already exists a *c* such that *T*b*i* = *T*(*S*c*,n*) then clearly *Ti* = *Sc*. Otherwise,

^

^

by Action (2), some tree *Sc* follows *T*b*i* by no later than stage *j* + ^*i*.

Verification of (iii). By Action (2), *Si* is distinct from all *Sj* (*j < i*) at level

2*i* and from all *Sj* (*j > i*) at level 2*j*. So *Si* /= *Sj* if *i* /= *j*.

Corollary 4.3 *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 [4.2](#_bookmark4) provides for an enumeration *S*0*, S*1*, S*2*,...* without repetition of all computable trees without dead ends. We use a diag- onalization argument to construct a tree *T* so that for all *n*, *T* ∩ {0*,* 1}*n*+1 /= *Sn* ∩ {0*,* 1}*n*+1. At stage 0 let *T* ∩ {0*,* 1}0 = {∅}. At stage *n* +1 we are given that *T* ∩ {0*,* 1}*n* is nonempty. Therefore there are at least 2 subtrees of

{0*,* 1}*n*+1 extending *T* ∩ {0*,* 1}*n*. Define *T* ∩ {0*,* 1}*n*+1 to be an extension which is different from *Sn* ∩ {0*,* 1}*n*+1.

Corollary 4.4 *Let* {[*Te*]}*e*∈*ω be the standard enumeration of the* Π0 *classes.*

1

*Then there is a decidable* Π0 *class P such that P* /= [*Te*] *for any Te without*

1

*dead ends.*

As a result of this corollary, {*e* : *Te* has no dead ends} /= {*e* : *Pe* = [*Te*] is decidable}. In fact both have distinct complexities. Let

Ext(*Pe*)= {*σ* : (∀*σ* ∈ *Te*)(∀*n*)(∃*τ* ∈ {0*,* 1}*n*) *σ-τ* ∈ *Te*}

By Konig’s Lemma, since the trees are subsets of 2*<ω*, this set is Π0. Therefore

1

{*e* : *Te* has no dead ends} = {*e* : *Te* = Ext(*Pe*)} is Π0. However,

1

{*e* : *Pe* is decidable} = {*e* : *Pe* = [*T* ] for some comp. *T* without dead ends}

= {*e* : (∃*a*) *φa* is a char. function for Ext(*Pe*)}

Therefore this latter set is Σ0. In [[2](#_bookmark6)], no distinction is made between these sets or their complexities. In light of these surprising results, the results of [[2](#_bookmark6)] must be revisited. We generalize Theorem [4.2](#_bookmark4).

2

Corollary 4.5 *Let* P*n* = {*P* : *P* = [*T* ] *is a* Π0

1

*class and T has* ≤ *n dead*

*ends*}*. Then in any enumeration (of a subfamily) of* Π0 *classes by uniformly computable trees, there is a* Π0 *class* [*T* ] ∈ P*n such that there is no e such that Te has* ≤ *n dead ends and* [*Te*]= [*T* ]*.*

1

1

Proof. Modify the proof of Theorem [4.2](#_bookmark4) so that for fixed *n*, trees become marked only if they are discovered to have *> n* dead ends. We leave details to the reader.

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 Π0 classes as well as

1

the complexity of index sets for decidable Π0 classes.

1

# References

1. Cenzer, D., *Effectively Closed Sets*, “ASL Lecture Notes in Logic,” to appear.
2. Cenzer, D. and J. Remmel, *Index Sets for* Π0 *Classes*, Annals of Pure and Applied Logic, 93

1

(1998), 3–61.

1. Cenzer, D. and J. Remmel, *Index Sets for Computable Real Functions*, “Proceedings of CCA03,” Cincinnati, July 2003, 163–182.
2. Ershov, Y., *Theory of Numberings*, “Handbook of Computability Theory (Edward R. Griffor, ed.),” North-Holland, Amsterdam, 1999, 473–503.
3. Friedberg, R., *Three Theorems on Recursive Enumeration*, Journal of Symbolic Logic, 23

(1958), 309–316.

1. Goncharov, S., S. Lempp, and R. Solomon, *Friedburg Numberings of Families of n-Computably Enumerable Sets*, Algebra and Logic, 41 (2002), 81–86.
2. Hinman, P., “Recursion-Theoretic Hierarchies,” Springer-Verlag, 1978.
3. Jockusch, C. and R. Soare, Π0 *Classes and Degrees of Theories*, Transactions of the American Mathematical Society, 173 (1972), 35–56.

1

1. Odifreddi, P., “Classical Recursion Theory”, Vol. 1, North-Holland, Amsterdam–New York, 1989.
2. Raichev, A., “Relative Randomness via RK-Reducibility”, Ph.D. thesis, University of Wisconsin–Madison, 2006.
3. Rogers, H., *G*o¨*del numberings of partial recursive functions*, Journal of Symbolic Logic, 23

(1958), 331–341.

1. Rogers, H., “Theory of recursive functions and effective computability”, McGraw Hill, 1967.
2. Soare, R., “Recursively Enumerable Sets and Degrees”, Springer-Verlag, 1987.
3. Suzuki, Y., *Enumerations of recursive sets*, Journal of Symbolic Logic, 24 (1959), 311.