# Almost everything looks ordinary somewhere: <br> Reals $n$-generic relative to some perfect tree 

Bernard Anderson

Gordon State College
November 18, 2013

## Introduction

## Computability Theory

- Computability Theory (also called Recursion Theory) is a branch of mathematical logic.
- We study sets of numbers, looking at their properties, patterns, and things in common with other sets that share these properties.
- We also study their computational power, what other sets of numbers they compute (and which compute them)


## Introduction

## Subsets of the natural numbers

In this talk, we will work with subsets of the natural numbers.
Earlier work
My work was based on earlier work done by Reimann and Slaman on relatively random sets.

Random sets are similar to generic sets. Random sets are unpredictable, generic sets are unpatterned.

## Countable and uncountalbe sets

## Definitions

A set is countable if we can list off its elements

For example, the natural numbers are countable: $1,2,3, \ldots$

A set is uncountable if it is not countable.

The set of infinite binary strings is not countable.

## Countable and uncountalbe sets

## Proof

Suppose $R_{1}, R_{2}, R_{3}, \ldots$ lists off all infinite binary strings.

Define the string $R$ by $R(n)=1-R_{n}(n)$.

For example, if $R_{1}=101 \ldots, R_{2}=100 \ldots, R_{3}=110 \ldots$ then $R=011 \ldots$.
$R$ always disagrees with $R_{n}$ at the $n$-th spot.

## Countable and uncountalbe sets

## Proof (continued)

Since $R$ is an infinite binary string, it must be listed, so $R=R_{m}$ for some $m$.

But then $R(m)=R_{m}(m)$ and $R(m)=1-R_{m}(m)$ for a contradiction.

We conclude the set of infinite binary strings is uncountable. $\square$

## Countable and uncountalbe sets

## Countable sets are small

Countable sets are much, much smaller than uncountable sets.

Joining together a finite number of countable sets still gives a countable set.

In fact, a countably infinite collection of countable sets is still countable.

We demonstrate how to count such a collection in the following diagram:

## Countable and uncountable sets

List 1: $1 \begin{array}{llllllllllllll} & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$

List 2: $\begin{array}{lllllllllllllll} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$
$\begin{array}{lllllllllllllll}\text { List 3: } & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$

List 4: $1 \begin{array}{llllllllllllll} & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$

List 5: $1 \begin{array}{llllllllllllll} & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$

## Countable and uncountable sets

List 1: $1-\frac{7}{-1} \begin{array}{lllllllllllll} & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$

List 2: $\quad 1 \begin{array}{lllllllllllll} & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$
List 3: $\sqrt{2} \begin{array}{lllllllllllll} & 4 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$

List 4: $1 \begin{array}{llllllllllllll} & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$

List 5: $1 \begin{array}{llllllllllllll} & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14\end{array}$

## Countable and uncountable sets

List 1 :

List 2:

List 3 :

List 4:

List 5:


## Countable and uncountable sets

List 1 :

List 2:

List 3:

List 4:

List 5:




891011121810
 : 7177 List List 5: $\sqrt{2} / 3 / 4{ }^{5}$

## Sets as binary strings

Infinite binary strings
We will often identify subsets of $\mathbb{N}$ with infinite binary strings

If $n \in A$ then we say $A(n)=1$. If $n \notin A$ then we say $A(n)=0$.

## Sets as binary strings

## Infinite binary strings

We will often identify subsets of $\mathbb{N}$ with infinite binary strings

If $n \in A$ then we say $A(n)=1$. If $n \notin A$ then we say $A(n)=0$.

We use the notation $A \Uparrow n$ to denote the elements of $A$ less than or equal to $n$.

For example, if 2,3 , and 5 are the elements of $A \| 5$, then the string for $A$ starts $01101 \ldots$

## Overview

In this talk we will look at how many sets of numbers inherently have patterns,
and how many can appear to have no patterns when viewed in the right context.

We start with another story ...

## Binary strings as paths

## Visual representation

We can also depict an infinite binary string (and hence a subset of $\mathbb{N}$ ) visually.

We draw a path through a tree, going left for 0 and right for 1 .

For example, consider $A=01101 \ldots$ (see next slide).

## Binary Strings as paths

$A=$


## Binary Strings as paths

$$
A=0
$$



## Binary Strings as paths

$A=01$


## Binary Strings as paths

$$
A=011
$$



## Binary Strings as paths

$$
A=0110
$$



## Binary Strings as paths

$$
A=01101 \ldots
$$



## Computability

## Computable Sets

We say a set is computable if a sufficiently powerful computer can determine if any number is in the set, given arbitrarily large finite amounts of time and memory space.

## Definition of computable

Although the above definition is vague, there are several precise definitions of a set being computable. These definitions have been shown to be equivalent.

## Computability

## Church's Thesis

- Church's Thesis states that this intuitive definition of Computable is correct.
- If we can describe an effective procedure to calculate a function, then it is computable.
- This hypothesis has held for well over half a century of research in mathematics and computer science.


## Computable Enumerability

## Computable Enumerability

We say a set $A$ is computably enumerable (c.e.) if we can computably (effectively) list the elements of the set.

## Computable Enumerability

## Computable Enumerability

We say a set $A$ is computably enumerable (c.e.) if we can computably (effectively) list the elements of the set.

This list may not be in order, so at any point in time we don't know if an unlisted number is not in the set or if it will be listed later.

## Computable Enumerability

Example
We wait while elements of the c.e. set $B$ are listed off.
$347 \in B, \quad 192 \in B, \quad 13 \in B, \quad 5882 \in B, \ldots$

## Computable Enumerability

Example
We wait while elements of the c.e. set $B$ are listed off.
$347 \in B, \quad 192 \in B, \quad 13 \in B, \quad 5882 \in B, \ldots$

We know 13 is in $B$. We don't know if 12 is in $B$.

Perhaps if we wait a minute longer we will see 12 added to $B$. Perhaps 12 isn't in $B$ and we could wait forever for it to appear.

## Definition of 1-Generic

Definition (Friedberg, 1957)
A set $A$ is 1-generic if for every c.e. set of finite strings $S$ there is a number $n$ such that either $A \| n \in S$ or $[A \| n] \cap S=\varnothing$.

## Definition of 1-Generic

## Definition (Friedberg, 1957)

A set $A$ is 1-generic if for every c.e. set of finite strings $S$ there is a number $n$ such that either $A \| n \in S$ or $[A \| n] \cap S=\varnothing$.

## Generic sets

Generic sets are typical, in the sense that they don't belong to any "small" set we can easily define.

If a set $A$ has an easy to describe pattern (i.e. all even numbers are in $A$, or no prime numbers are in $A$ ) then $A$ is not 1-generic.

## Generic sets and the binary tree

## Visual representation

We can represent generic sets visually on the binary tree (see next 3 slides).

If $A$ is 1 -generic then for every c.e. set $S$, we see at some finite initial segment that either $A \| n \in S$, or $A \| n$ cannot be extended to meet $S$.

## $A \| 2 \in S$

$A \Uparrow 2$ meets $S$.


## $[A \| 1] \cap S=\varnothing$

$A \| 1$ cannot be extended to meet $S$.


## Neither

Neither holds. If $S$ is c.e., then $A$ is not 1-generic.


## For mathematicians only

## Connection to Analysis

We can place a canonical topology on the Cantor Space to interpret genericity in terms of analysis.

A comeager set is effective if we can computably list its neighborhoods.

A set $A \in 2^{\mathbb{N}}$ is 1 -generic if it is in every effective comeager set.

A set $A \in 2^{\mathbb{N}}$ is 1 -random if it is in every effective measure one set.

## Building a generic set

## Building a generic set

Building a set $A$ which works for any given c.e. set of finite strings $S$ is easy.

Building an $A$ which simultaneously works for all c.e. sets of finite strings is not.

## Building a generic set

## Building a generic set

Building a set $A$ which works for any given c.e. set of finite strings $S$ is easy.

Building an $A$ which simultaneously works for all c.e. sets of finite strings is not.

Since generic sets have no easy to describe patterns, the sets themselves are hard to describe

To build a generic set, we need some more definitions.

## Relative computability

Oracle machines
Let $A$ be a set, and suppose our computer can obtain information about $A$ as part of its computation process.

If this computer can calculate a set $B$ then we say $A$ can compute B.

We denote this by $A \geq_{T} B$.

Computing a generic set
We will define a set zero jump, $\varnothing^{\prime}$, and show that it computes a 1-generic set.

## Programs

## Computer Programs

- We view computer programs as strings of symbols
- Many, like "alsdjfjpiel3fne!lneij;tgieja" don't do anything.
- Others, like "Input $x$, Output $x+1$ " work as intended.


## Programs

## Definition

If a program $P$ run with input $x$ comes to a halt with meaningful output, we say it converges and write $P(x) \downarrow$.

Definition
We say a program is total if for every input the program comes to a halt and provides meaningful output.

## Enumeration

Listing programs

- We want to find a way to list off all possible computer programs.
- We let $P_{n}$ denote the $n$-th program in our list.
- One example is. . .


## Enumeration

One possible list
$P_{1}$ : a
$P_{2}: \mathrm{b}$
$P_{3}$ : aa
$P_{4}$ : c
$P_{5}: a b$
$P_{6}$ : ba
$P_{7}$ : aaa
引
$P_{9738928}$ : Input $x$, Output $x+1$

## Enumeration

## Other lists

Clearly, there are better lists available.
However, in Computability Theory, which list we use doesn't matter.

## Previous talk (aside)

Last year we saw that no matter how we list the programs, there will always be a program that references itself.

For example program 537 would output 537 on any input.

## The halting set

Zero jump
We define zero jump (also called the halting set) to be the set of numbers $n$ such that the $n$th program halts when run with input $n$.

Formally, $\varnothing^{\prime}=\left\{n \mid P_{n}(n) \downarrow\right\}$.

## The halting set (continued)

Properties of zero jump

- A diagonalization argument can be used to show that $\varnothing^{\prime}$ is not computable.


## The halting set (continued)

Properties of zero jump

- A diagonalization argument can be used to show that $\varnothing^{\prime}$ is not computable.
- $\varnothing^{\prime}$ can answer all $\Sigma_{1}$ questions (all "Does there exist" questions about c.e. sets).

This can be shown by writing a program that halts iff the answer to the question is "Yes".

Similarly, $\varnothing^{\prime}$ can compute all c.e. sets.

## Building a generic set (continued)

Computing a generic set
$\varnothing^{\prime}$ can compute a 1-generic set.

Theorem (Friedberg, 1957)
There is a 1 -generic set $G \leq_{T} \varnothing^{\prime}$.

## $\varnothing^{\prime}$ computes a 1-generic set

## Proof sketch(see slides 2-4 down)

We start with $G$ as an empty string, $G_{0}=\langle \rangle$.

We consider the c.e. sets of finite strings ( $S_{1}, S_{2}, S_{3}, \ldots$ ) one at a time.

At stage $i$, we will extend $G_{i-1}$ to a $G_{i}$ that works for $S_{i}$.

## $\varnothing^{\prime}$ computes a 1-generic set

## Proof sketch (continued)

Each time, we ask $\varnothing^{\prime}$, "Can $G_{i-1}$ be extended to meet the next c.e. set $S_{i}$ ?"

## $\varnothing^{\prime}$ computes a 1-generic set

## Proof sketch (continued)

Each time, we ask $\varnothing^{\prime}$, "Can $G_{i-1}$ be extended to meet the next c.e. set $S_{i}$ ?"

If yes, we extend $G_{i-1}$ to a $G_{i} \in S_{i}$.

## $\varnothing^{\prime}$ computes a 1-generic set

## Proof sketch (continued)

Each time, we ask $\varnothing^{\prime}$, "Can $G_{i-1}$ be extended to meet the next c.e. set $S_{i}$ ?"

If yes, we extend $G_{i-1}$ to a $G_{i} \in S_{i}$.

If no, we do nothing, $G_{i}=G_{i-1}$, so we have $\left[G_{i}\right] \cap S_{i}=\varnothing$.

## $\varnothing^{\prime}$ computes a 1-generic set

Can $G_{0}$ be extended to meet $S_{1}$ ?


## $\varnothing^{\prime}$ computes a 1-generic set

Yes. Extend $G_{0}$ to $G_{1} \in S_{1}$.


## $\varnothing^{\prime}$ computes a 1-generic set

Can $G_{1}$ be extended to meet $S_{2}$ ?


## $\varnothing^{\prime}$ computes a 1-generic set

No. Let $G_{2}=G_{1}$ so $\left[G_{2}\right] \cap S_{2}=\varnothing$.


## $\varnothing^{\prime}$ computes a 1-generic set

Can $G_{2}$ be extended to meet $S_{3}$ ?


## $\varnothing^{\prime}$ computes a 1-generic set

Yes. Extend $G_{2}$ to $G_{3} \in S_{3}$.


## Joins of sets

## Definition

If $A$ and $B$ are strings, we let $A \oplus B$ be the string obtained by alternating elements from $A$ and $B$.

For example, if $A=01101 \ldots$ and $B=00000 \ldots$ then $A \oplus B=0010100010 \ldots$

## Joins of sets (continued)

$A \oplus 00000 \ldots$
For any set $A$, the set $A \oplus 00000 \ldots$ is not 1-generic (there is an obvious pattern).

## Joins of sets (continued)

## $A \oplus 00000 \ldots$

For any set $A$, the set $A \oplus 00000 \ldots$ is not 1-generic (there is an obvious pattern).

This is witnessed by the c.e. set $S$ of finite strings (see diagram next slide):
$S=\{\sigma \mid \exists n<$ length $(\sigma)[n$ is even and $\sigma(n)=1]\}$.

## Joins of sets (continued)

$S$ witnesses $01101 \ldots \oplus 00000 \ldots$ is not 1 -generic.


## Joins of sets (continued)

"Almost" generic sets
We've seen that even if $G$ is 1 -generic, $G \oplus 00000 \ldots$ is not 1 -generic.

## Joins of sets (continued)

"Almost" generic sets
We've seen that even if $G$ is 1 -generic, $G \oplus 00000 \ldots$ is not 1-generic.

In fact if $C$ is any non-generic set (even a computable set), then $G \oplus C$ is not 1 -generic.

## Joins of sets (continued)

"Almost" generic sets
We've seen that even if $G$ is 1 -generic, $G \oplus 00000 \ldots$ is not 1-generic.

In fact if $C$ is any non-generic set (even a computable set), then $G \oplus C$ is not 1 -generic.

Finally, $G \oplus G$ is not 1-generic.

## Generic relative to some perfect tree

A more general version of generic
Many of the sets in the previous examples were "essentially" generic, but had a single pattern applied to them.

We can consider if there is a broader notion of genericity.

In particular, we can ask which sets seem generic from some perspective, and which sets inherently lack any genericity?

We introduce some new definitions to consider this.

## Generic relative to some perfect tree (continued)

Trees
A tree is a set of strings $T$ closed under initial segments.

For example, if $1011 \in T$ then we must have $1 \in T, 10 \in T$, and $101 \in T$.

The full binary tree we have seen so far is the largest possible tree.

We can make smaller trees by trimming off branches (see board for examples).

## Generic relative to some perfect tree (continued)

## Paths

An infinite string $A$ is a path through a tree $T$ if for all $n$, we have $A \| n \in T$.

Informally, we can draw the string without leaving the tree (see board).

For example, every infinite string is a path through the full binary tree.

## Generic relative to some perfect tree (continued)

## Perfect trees

We say a tree $T$ is perfect if it has no isolated paths, i.e. every node in the tree has two incomparable extensions (see diagram next 2 slides).

## Generic relative to some perfect tree (continued)

## Perfect trees

We say a tree $T$ is perfect if it has no isolated paths, i.e. every node in the tree has two incomparable extensions (see diagram next 2 slides).

Every path through a perfect tree has infinitely many places it can branch off.

The full binary tree $2^{<\mathbb{N}}$ is perfect.

## Generic relative to some perfect tree (continued)

Example of a perfect tree (no isolated paths)


## Generic relative to some perfect tree (continued)

## Example of a non-perfect tree



## Generic relative to some perfect tree (continued)

## Definition (Slaman)

Let $A$ be a set and $T$ be a perfect tree. Then $A$ is 1-generic relative to $T$ if $A$ is a path through $T$ and for every c.e. in $T$ set of finite strings $S \subseteq T$, there is a number $n$ such that either $A \| n \in S$ or $[A \| n] \cap S=\varnothing$.

## Generic relative to some perfect tree (continued)

Definition (Slaman)
Let $A$ be a set and $T$ be a perfect tree. Then $A$ is 1-generic relative to $T$ if $A$ is a path through $T$ and for every c.e. in $T$ set of finite strings $S \subseteq T$, there is a number $n$ such that either $A \| n \in S$ or $[A \| n] \cap S=\varnothing$.

A set $A$ is generic relative to some perfect tree if there exists a perfect tree $T$ such that $A$ is generic relative to $T$.

## Generic relative to some perfect tree (continued)

Example
We saw earlier that even for a 1-generic $G$ we have that $G \oplus 00000 \ldots$ is not 1-generic.

## Generic relative to some perfect tree (continued)

## Example

We saw earlier that even for a 1-generic $G$ we have that $G \oplus 00000 \ldots$ is not 1-generic.

However if we let
$T=\left\{\sigma \in 2^{<\mathbb{N}} \mid \forall n<\right.$ length $(\sigma)[n$ even $\left.\rightarrow \sigma(n)=0]\right\}$
Then $G \oplus 00000 \ldots$ is 1 -generic relative to $T$ (see diagram next 2 slides).

Hence $G \oplus 00000 \ldots$ is 1-generic relative to some perfect tree.

## Generic relative to some perfect tree (continued)

$S$ witnesses $G \oplus 00000 \ldots$ is not 1 -generic.


## Generic relative to some perfect tree (continued)

$G \oplus 00000 \ldots$ is 1-generic relative to $T$.


## Generic relative to some perfect tree (continued)

## More examples

In the same way we can see that if $G$ is 1 -generic and $C$ is computable, then $G \oplus C$ and $G \oplus G$ are 1-generic relative to some perfect tree.

On the other hand any c.e. set (in fact any $n$-c.e. set), including $\varnothing^{\prime}$, is not 1 -generic relative to any perfect tree.

## Notes on definition

## Importance of c.e. in $T$

The requirement that $A$ work for all sets $S$ that are c.e. in $T$ (rather than just c.e.) is important.

It means that if we chose a more complicated $T$, it must satisfy more complicated sets $S$.

This prevents us from making many sets $A 1$-generic relative to some perfect tree just by choosing a highly complicated $T$.

## Notes on definition (continued)

Importance of perfect tree
The requirement that $T$ be a perfect tree is also important.

It prevents us from making many sets $A 1$-generic relative to some perfect tree by just making $A$ an isolated path on $T$ (see diagram next slide).

## Notes on definition (continued)

Isolated path is generic relative to non-perfect $T$


## First main result

Recall, the set of all subsets of $\mathbb{N}$ has cardinality of the continuum.

We can ask, how many sets are 1-generic relative to some perfect tree?

Theorem (Anderson)
Only countably many sets are not 1-generic relative to any perfect tree

## Definitions

2-generic
We say a finite set of strings $S$ is c.e. $\left(\varnothing^{\prime}\right)$ if a computer that can access $\varnothing^{\prime}$ can list the elements of $S$.

We say a set is 2-generic if for every c.e. $\left(\varnothing^{\prime}\right)$ set of finite strings $S$ there is a number $n$ such that either $A \| n \in S$ or $[A \| n] \cap S=\varnothing$.

This is the definition of 1-generic except c.e. has been repalced by c.e. $\left(\varnothing^{\prime}\right)$.

This is a stronger notion of genericity. Every 2-generic set is 1-generic.

## Definitions (continued)

## n-generic

We can define $n$-generic and $n$-generic relative to some perfect tree similarly for any number $n$.

The larger $n$ is, the larger and more complicated the set of patterns that must be avoided.

## First main result (continued)

Our result still holds for stronger notions of genericity

Theorem (Anderson)
For every n, only countably many sets are not n-generic relative to any perfect tree

## Reverse Mathematics

For the rest of the talk, we look at how the previous result applies to an area of math known as Reverse Mathematics.

But first, back to our story ...

## Reverse mathematics (continued)

## Introduction to Reverse Mathematics

For any theorem of mathematics, we can ask "What axioms are necessary to prove the theorem?"

We can then classify the axiomatic strength of theorems. Stronger theorems require stronger axioms to prove them.

## Reverse mathematics (continued)

Reverse mathematics (continued)
There is a hierarchy of five axioms, each of which proves the axioms below it.

Extensive work has shown that many theorems of classical mathematics have the strength of one of these axioms (see diagram next slide).

## Hierarchy with examples

$\Pi_{1}^{1}$-CA (i.e. Which computable trees in the Baire space $\mathbb{N}^{<\mathbb{N}}$ are well founded)
$\mathbf{A T R}_{0}$ (i.e. Given ordinals $\alpha$ and $\beta$, either $\alpha \leq \beta$ or $\beta \leq \alpha$.)
$\mathbf{A C A}_{0}$ (i.e. Existence of maximal ideals, Range of every $f: \mathbb{N} \rightarrow \mathbb{N}$ exists.)
$\mathbf{W K L}_{0}$ (i.e. Existence of prime ideals, Cantor space is compact)
$\mathbf{R C A}_{0}$ (i.e. Most theorems dealing with finite objects)

## Abbreviations

## $\mathrm{GRPT}_{n}$ and GRPT

Let $\mathrm{GRPT}_{n}$ denote the statement "Only countably many sets are not $n$-generic relative to any perfect tree."

Let GRPT denote the statement "For all $n$, only countably many sets are not $n$-generic relative to any perfect tree."

## Second main result

Our earlier theorems have unusually high axiomatic strength.

Theorem (Anderson)
$\Pi_{1}^{1}$-CA fails to prove $G R P T_{2}$.
Theorem (Anderson)
ZFC ${ }^{-}$and existence of finitely many iterations of the power set of $(\mathbb{N})$ fail to prove GRPT.

The later result is at an axiom strength far above $\Pi_{1}^{1}$-CA; few theorems outside of set theory are this strong.

## Conclusion

Further study

- There are a lot of other interesting results in Computability Theory.
- The field is still fairly new and expanding.
- The objects we study are relatively tangible.


## Conclusion

## References

1. B. A. Anderson. Reals $n$-generic relative to some perfect tree. J. Symbolic Logic, 73(2):401-411, 2008.
2. R. I. Soare. Recursively Enumerable Sets and Degrees. Springer-Verlag, 1987.

## Conclusion

Thank you

