Negligibility, depth, and algorithmic randomness
Part 2

Christopher P. Porter
University of Florida

Joint work with Laurent Bienvenu

UF Logic Seminar
September 12, 2014

Last time

- I motivated the study of certain limitations of probabilistic computation using tools from algorithmic randomness and computability theory.
- I introduced the basics of algorithmic randomness and discussed my preferred model of probabilistic computation.
- I introduced the notion of a negligible Π^0_1 class and discussed some basic results about such classes.

Today, I will

1. introduce the notion of a deep Π^0_1 class;
2. prove a number of basic results about deep classes;
3. outline the proof that the collection of consistent completions of Peano arithmetic is a deep class, and
4. provide several other examples of deep classes.

Outline of today's talk

1. Review
2. Introducing deep Π^0_1 classes
3. Examples of Π^0_1 classes

1. Review

Recall...

- Martin-Löf random sequences: Infinite binary sequences that avoid a family of effective null subsets of 2^ω.
- Probabilistic computation: Turing computation with an algorithmically random oracle (usually a Martin-Löf random sequence).
- Π^0_1 classes: Effectively closed classes, or equivalently:
  - the collection of infinite paths through a computable tree; or
  - the collection of infinite paths through a co-c.e. tree.

Further recall...

- $\mathcal{S} \subseteq 2^\omega$ is negligible if there is no probabilistic procedure for computing a member of $\mathcal{S}$ with positive probability.

- A left-c.e. semi-measure can be seen as a super-additive measure that can be computably approximated from below.
  - There is a correspondence between left-c.e. semi-measures and Turing functionals.
  - There is a universal left-c.e. semi-measure M.

- $\mathcal{S} \subseteq 2^\omega$ is negligible if and only if $\overline{M}(\mathcal{S}) = 0$, where $\overline{M}$ is the largest measure such that $\overline{M} \leq M$.

2. Introducing Deep Π^0_1 classes

Deep classes: the idea

Unlike negligibility, we only define depth for Π^0_1 classes.

Depth is a property that is strictly stronger than negligibility for Π^0_1 classes.

Instead of considering how difficult it is to produce a path through a Π^0_1 class $\mathcal{P}$, we can consider how difficult it is to produce an *initial segment* of some path through $\mathcal{P}$, level by level.

Deep Π^0_1 classes are the "most difficult" Π^0_1 classes in this respect.

Some notation

Let $\mathcal{P} \subseteq 2^\omega$ be a Π^0_1 class.

Let $T_\mathcal{P} \subseteq 2^{<\omega}$ be the set of extendible nodes of $\mathcal{P}$,

$$T_\mathcal{P} = \{\sigma \in 2^{<\omega} : [\![\sigma]\!] \cap \mathcal{P} \neq \emptyset\}.$$

Thus $T_\mathcal{P}$ is the canonical co-c.e. tree such that $\mathcal{P} = [T_\mathcal{P}]$ (the set of infinite paths through $T_\mathcal{P}$).

Hereafter T will stand for $T_\mathcal{P}$.

For each $n \in \omega$, T_n consists of all strings in T of length n.

Deep classes: the definition

Let $\mathcal{P}$ be a Π^0_1 class and let T be the canonical co-c.e. tree such that $\mathcal{P} = [T]$.

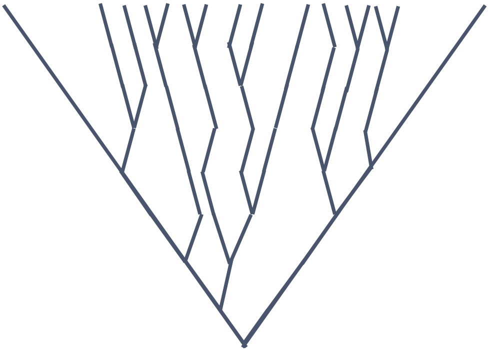
$\mathcal{P}$ is a *deep class* if there is some computable, non-decreasing, unbounded function $h : \omega \to \omega$ such that

$$M(T_n) \leq 2^{-h(n)},$$

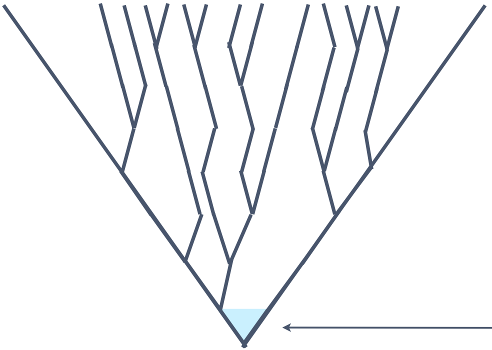
where $M(T_n) = \sum_{\sigma \in T_n} M(\sigma)$.

That is, the probability of producing some initial segment of a path through $\mathcal{P}$ is effectively bounded from above.

Note: Every deep class is negligible.

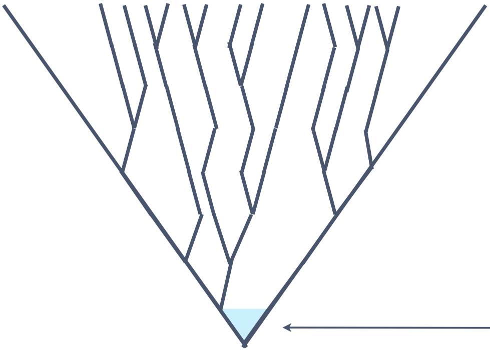


Length Probability

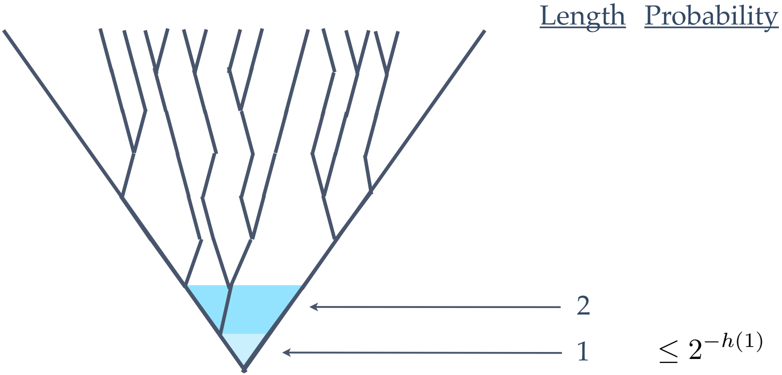


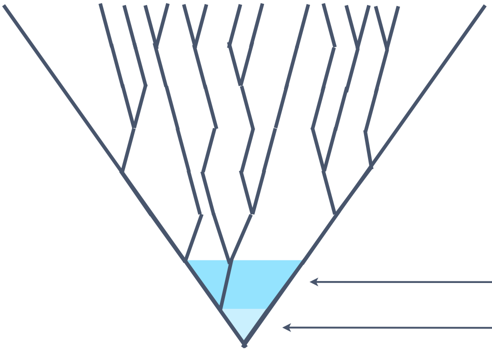
Length Probability

1

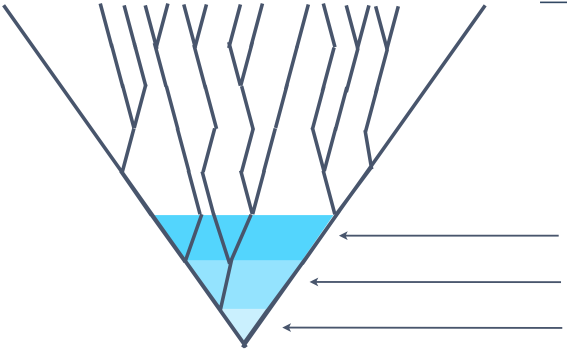


Length	Probability
1	$\leq 2^{-h(1)}$

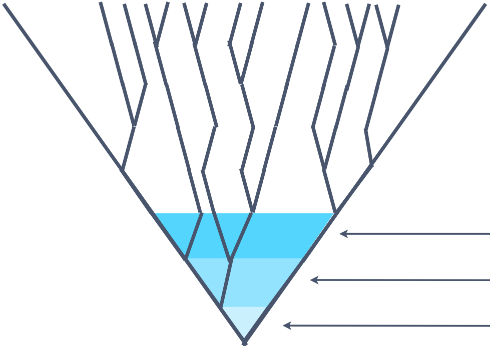

Length Probability
2
1 $\leq 2^{-h(1)}$



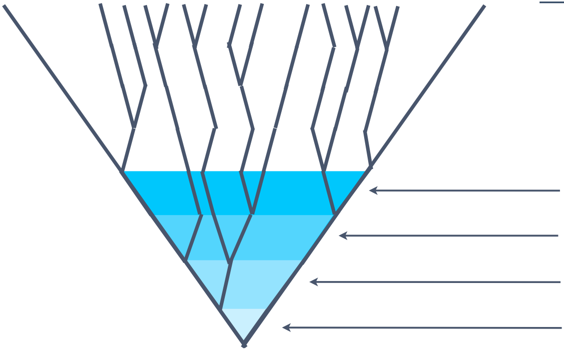
Length	Probability
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$



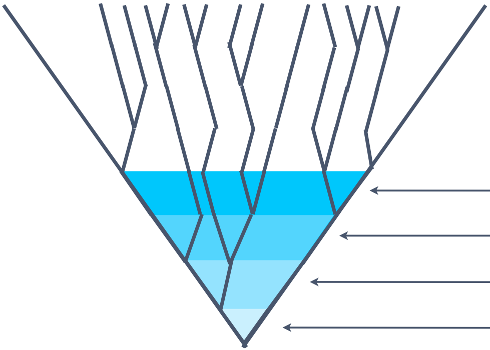
Length	Probability
3	
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$



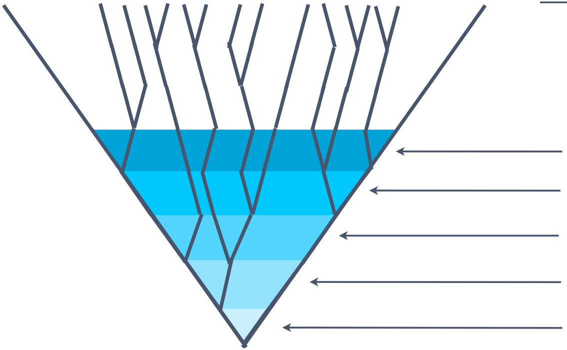
Length	Probability
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$



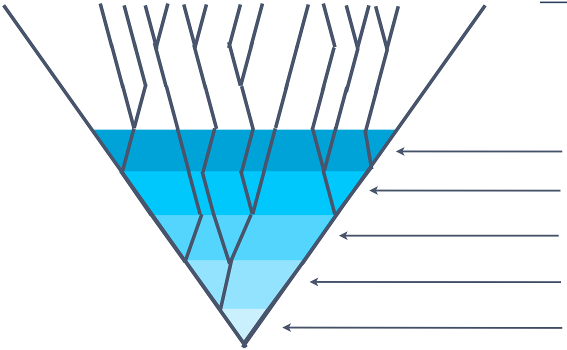
Length	Probability
4	
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$



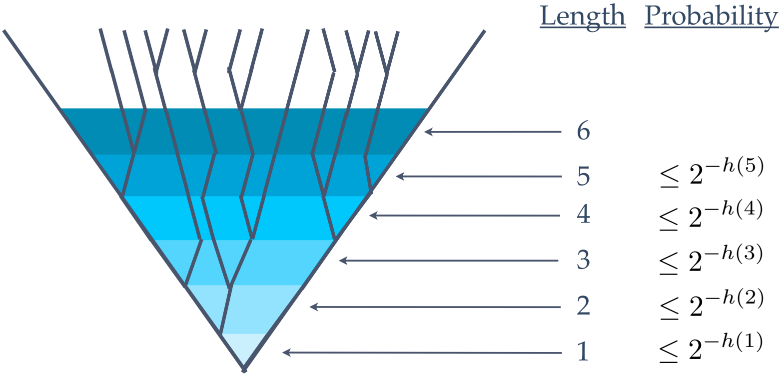
Length	Probability
4	$\leq 2^{-h(4)}$
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$



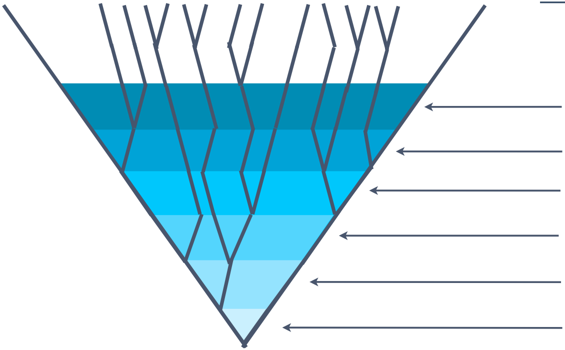
Length	Probability
5	
4	$\leq 2^{-h(4)}$
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$



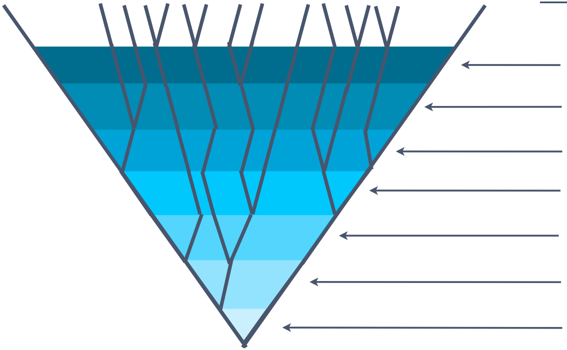
Length	Probability
5	$\leq 2^{-h(5)}$
4	$\leq 2^{-h(4)}$
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$



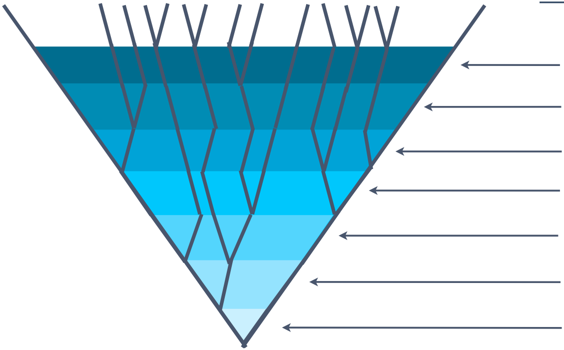
Length
Probability
6
5 $\leq 2^{-h(5)}$
4 $\leq 2^{-h(4)}$
3 $\leq 2^{-h(3)}$
2 $\leq 2^{-h(2)}$
1 $\leq 2^{-h(1)}$



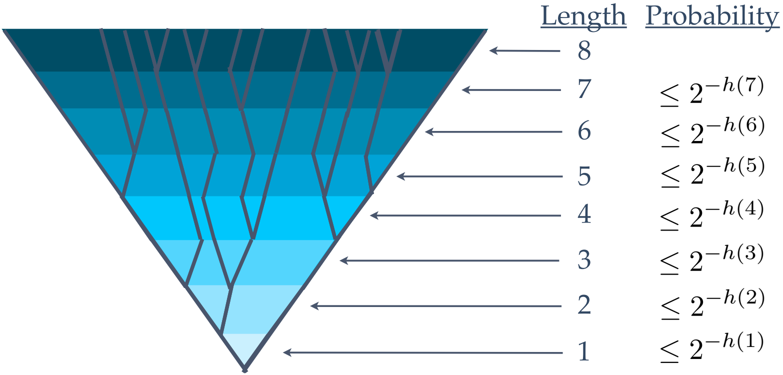
Length	Probability
6	$\leq 2^{-h(6)}$
5	$\leq 2^{-h(5)}$
4	$\leq 2^{-h(4)}$
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$

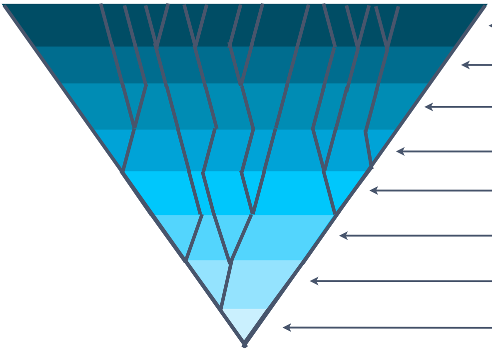


Length	Probability
7	
6	$\leq 2^{-h(6)}$
5	$\leq 2^{-h(5)}$
4	$\leq 2^{-h(4)}$
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$



Length	Probability
7	$\leq 2^{-h(7)}$
6	$\leq 2^{-h(6)}$
5	$\leq 2^{-h(5)}$
4	$\leq 2^{-h(4)}$
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$


Length
Probability
8
7
$\leq 2^{-h(7)}$
6
$\leq 2^{-h(6)}$
5
$\leq 2^{-h(5)}$
4
$\leq 2^{-h(4)}$
3
$\leq 2^{-h(3)}$
2
$\leq 2^{-h(2)}$
1
$\leq 2^{-h(1)}$



Length	Probability
8	$\leq 2^{-h(8)}$
7	$\leq 2^{-h(7)}$
6	$\leq 2^{-h(6)}$
5	$\leq 2^{-h(5)}$
4	$\leq 2^{-h(4)}$
3	$\leq 2^{-h(3)}$
2	$\leq 2^{-h(2)}$
1	$\leq 2^{-h(1)}$

Why use the co-c.e. tree in the definition of depth?

For every Π^0_1 class $\mathcal{P}$ there is a computable tree $S \subseteq 2^{<\omega}$ such that $\mathcal{P} = [S]$.

Why can't we use this computable tree S in the definition of depth?

First, in general, S will contain non-extendible nodes, so even if we can compute some element in S_n, we still may fail to compute an initial segment of a member of $\mathcal{P}$.

But this observation doesn't rule out the possibility that we can define depth in terms of computable trees.

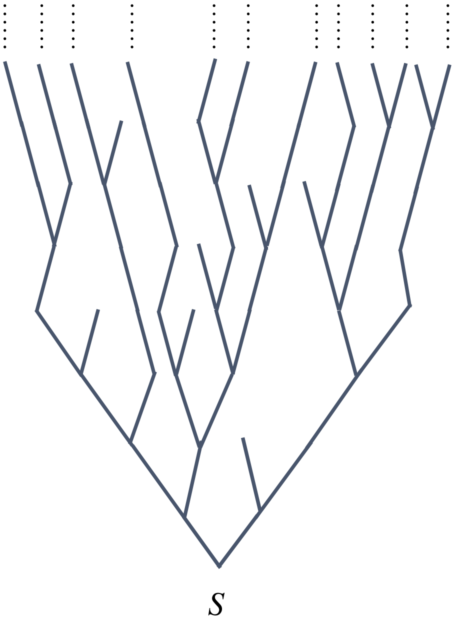
A better reason

Theorem (Bienvenu, Porter)

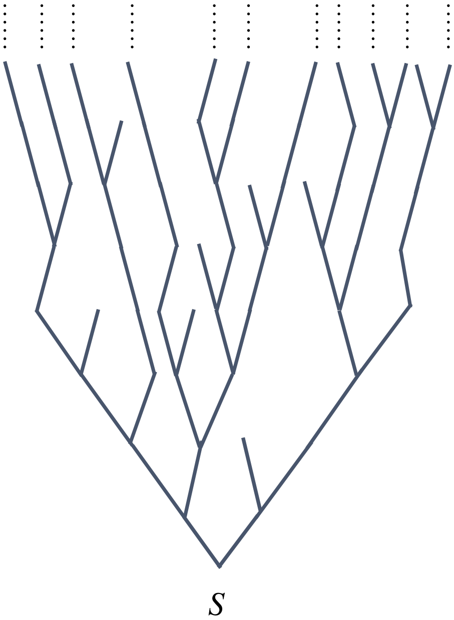
Let S be a computable tree. Then there is no computable order h such that $M(S_n) \leq 2^{-h(n)}$ for every $n \in \omega$.

Corollary (Bienvenu, Porter)

Let S be a tree with a computable sub-tree. Then there is no computable order h such that $M(S_n) \leq 2^{-h(n)}$ for every $n \in \omega$.

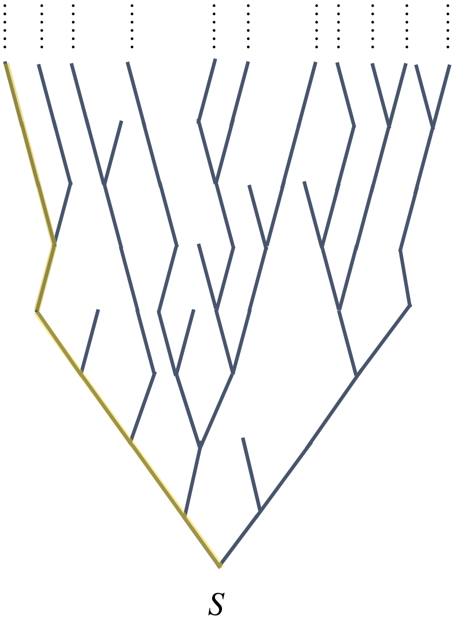

S

Case 1: S has only finitely many non-extendible nodes.

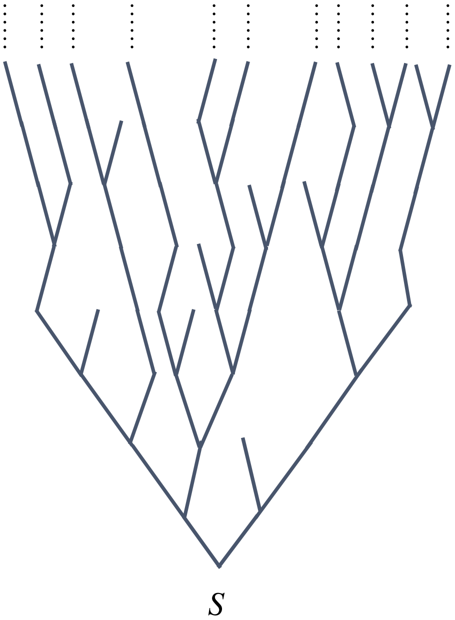


Case 1: S has only finitely many non-extendible nodes.

In this case, the left-most path of S is computable.

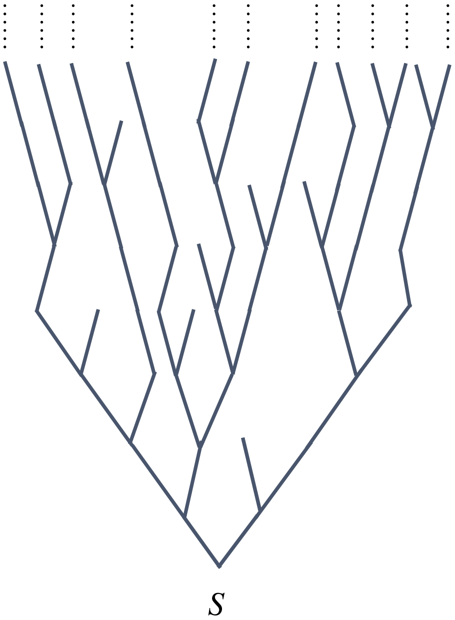


Case 2: S has infinitely many non-extendible nodes.



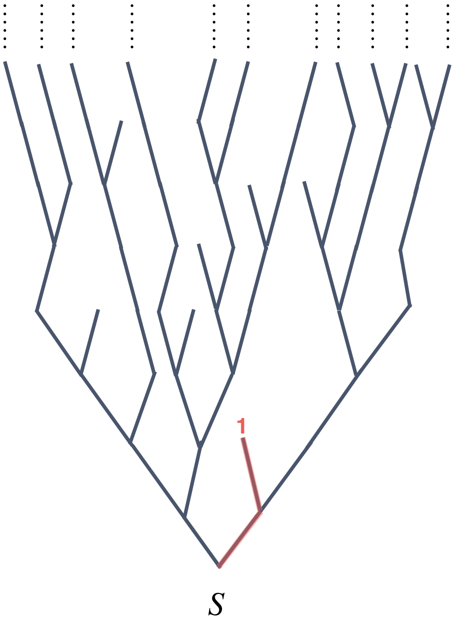
Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.



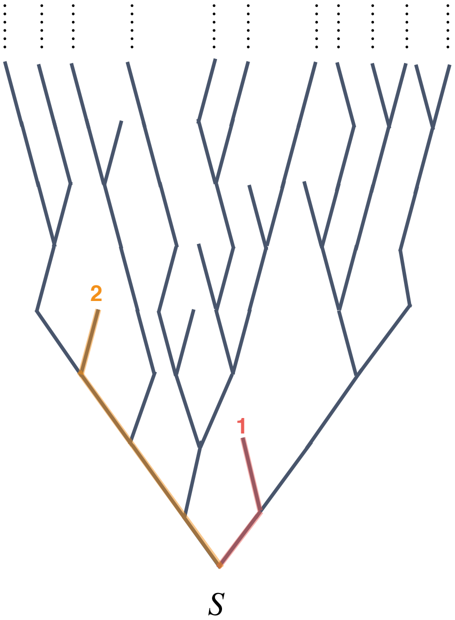
Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.



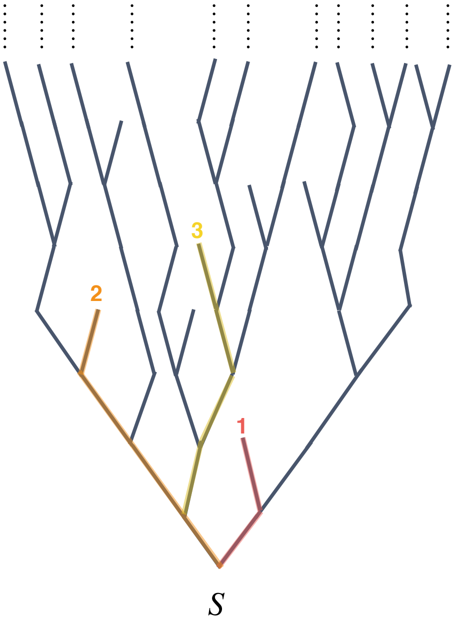
Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.



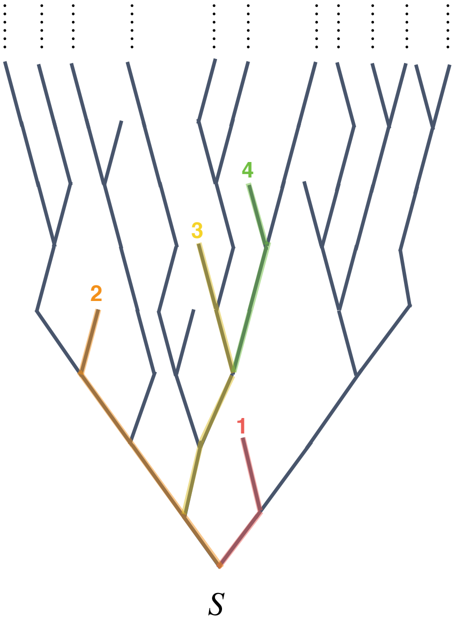
Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.



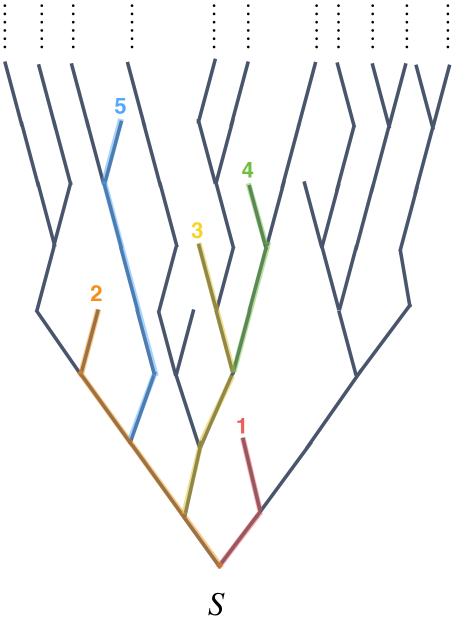
Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.



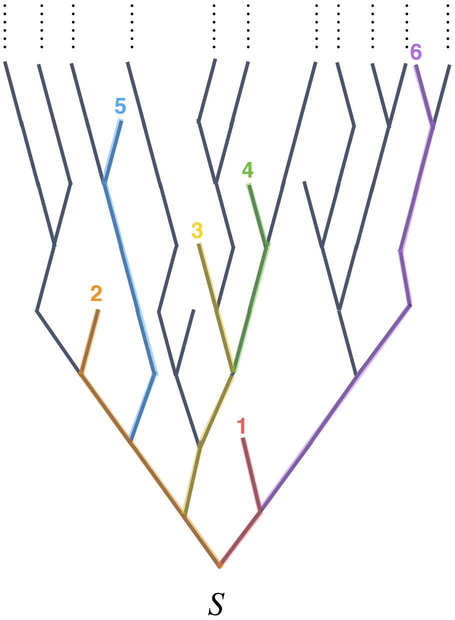
Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.



Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.

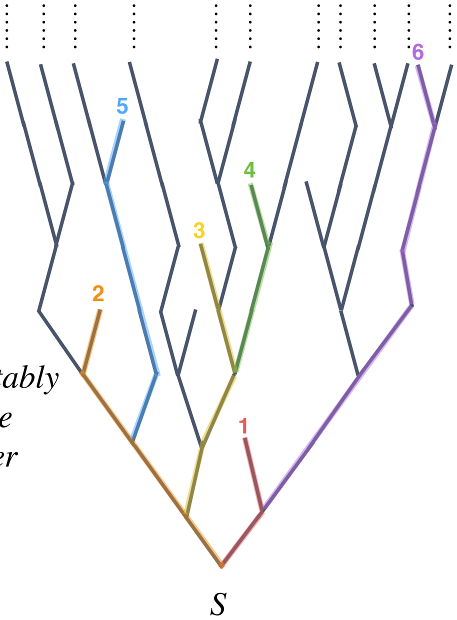


Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.

Next let $f : \mathbb{N} \to \mathbb{N}$ *be a function that is computably approximable from above with no computable lower bound such that*

$$\sum_{n \in \mathbb{N}} 2^{-f(n)} \leq 1.$$

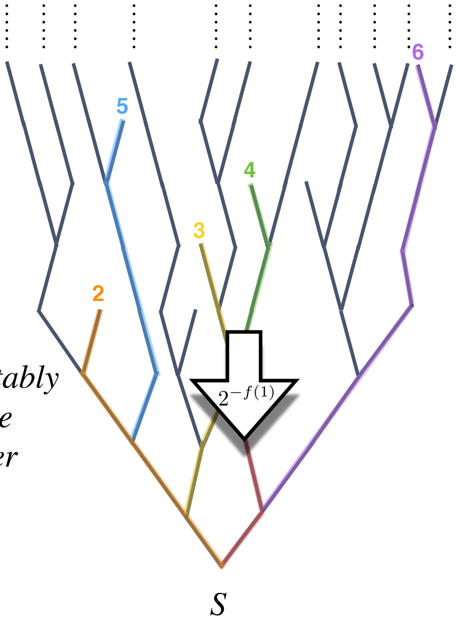


Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.

Next let $f : \mathbb{N} \to \mathbb{N}$ *be a function that is computably approximable from above with no computable lower bound such that*

$$\sum_{n \in \mathbb{N}} 2^{-f(n)} \leq 1.$$

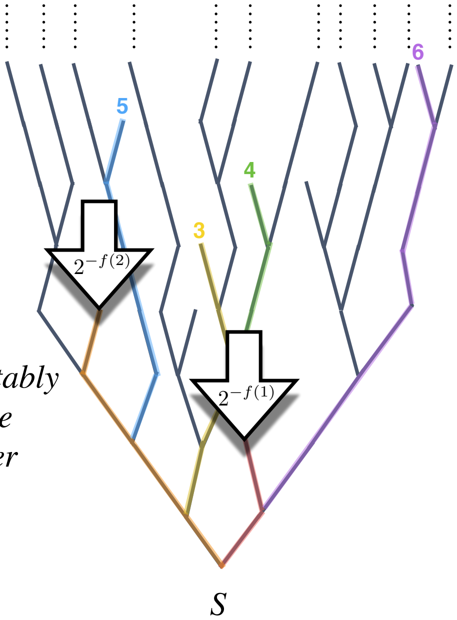


Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.

Next let $f : \mathbb{N} \to \mathbb{N}$ *be a function that is computably approximable from above with no computable lower bound such that*

$$\sum_{n\in\mathbb{N}} 2^{-f(n)} \leq 1.$$

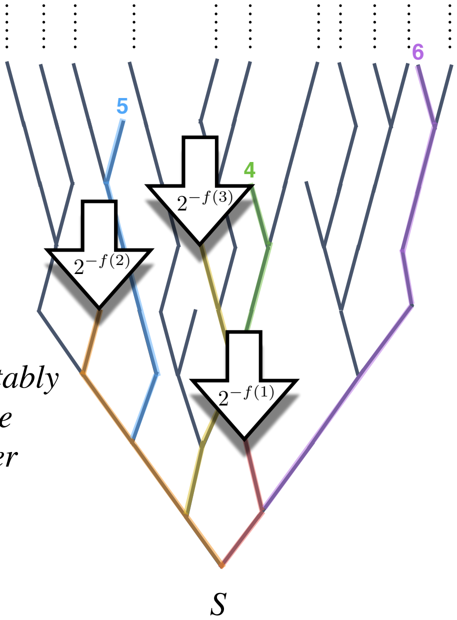


Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.

Next let $f : \mathbb{N} \to \mathbb{N}$ *be a function that is computably approximable from above with no computable lower bound such that*

$$\sum_{n \in \mathbb{N}} 2^{-f(n)} \leq 1.$$

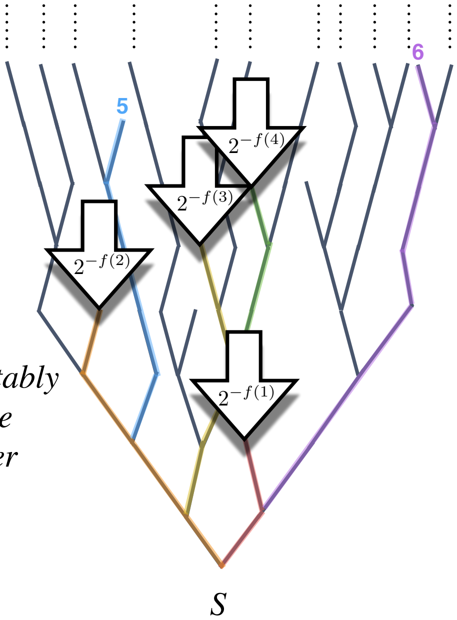


Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.

Next let $f : \mathbb{N} \to \mathbb{N}$ *be a function that is computably approximable from above with no computable lower bound such that*

$$\sum_{n \in \mathbb{N}} 2^{-f(n)} \leq 1.$$

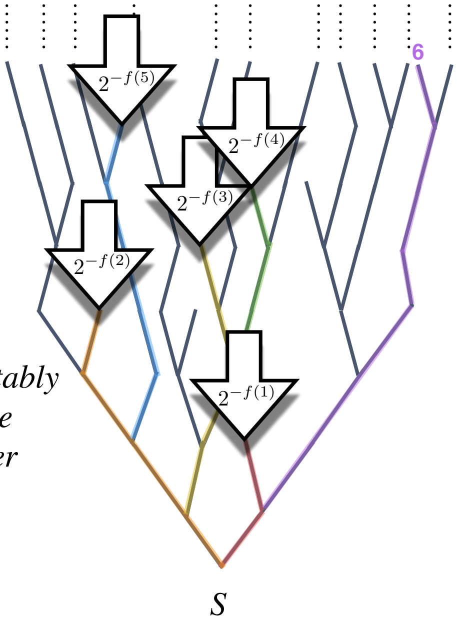


Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.

Next let $f : \mathbb{N} \to \mathbb{N}$ *be a function that is computably approximable from above with no computable lower bound such that*

$$\sum_{n\in\mathbb{N}} 2^{-f(n)} \leq 1.$$

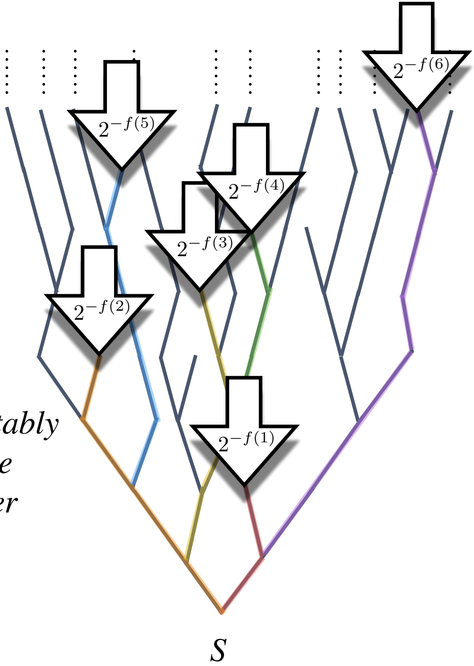


Case 2: S has infinitely many non-extendible nodes.

In this case, first we find a sequence of non-extendible nodes of increasing length.

Next let $f : \mathbb{N} \to \mathbb{N}$ *be a function that is computably approximable from above with no computable lower bound such that*

$$\sum_{n\in\mathbb{N}} 2^{-f(n)} \leq 1.$$



Does such an f exist?

Does such an f exist?

Yes!

Does such an f exist?

Yes!

Let U be a universal, prefix-free Turing machine.

For each $\sigma \in 2^{<\omega}$, the *prefix-free Kolmogorov complexity* of σ is defined to be

$$K(\sigma) := \min\{|\tau| : U(\tau) = \sigma\}.$$

If $(\sigma_i)_{i\in\omega}$ is an enumeration of $2^{<\omega}$ in length-lexicographical order, then

$$f(i) = K(\sigma_i)$$

is the desired function f.

Depth vs. negligibility

It is not hard to show that every deep class is negligible.

Is every negligible class deep?

Depth vs. negligibility

It is not hard to show that every deep class is negligible.

Is every negligible class deep? No.

Depth vs. negligibility

It is not hard to show that every deep class is negligible.

Is every negligible class deep? No.

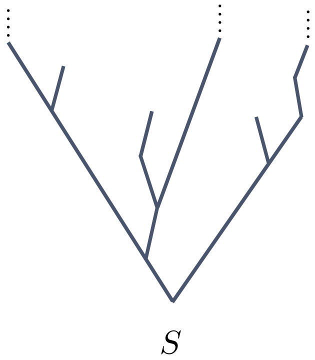
Theorem (Bienvenu, Porter)

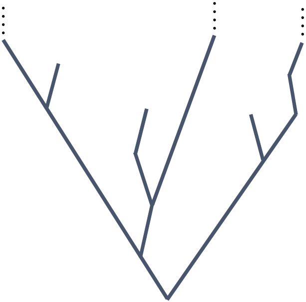
For every deep class $\mathcal{P}$, there is negligible class $\mathcal{Q}$ that is not deep such that

- *for every $X \in \mathcal{P}$, we have $X \in \mathcal{Q}$, and*
- *for every $Y \in \mathcal{Q}$, $Y = \sigma^\frown X$ for some $\sigma \in 2^{<\omega}$ and $X \in \mathcal{P}$.*

In other words, every deep class is Muchnik equivalent to a negligible Π^0_1 class that is not deep.

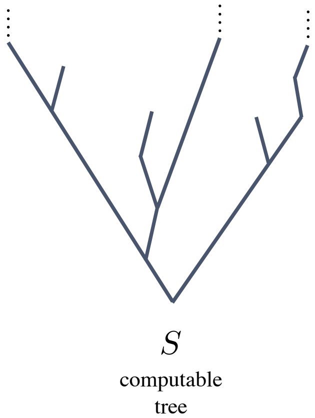
However, it is worth noting that depth is preserved under Medvedev equivalence.

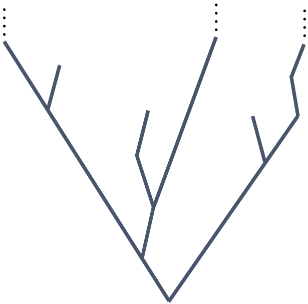

S



S

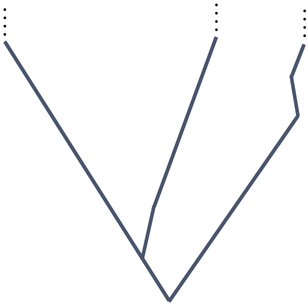
computable tree


S
computable
tree
T



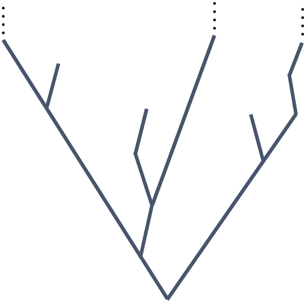
S

computable
tree



T

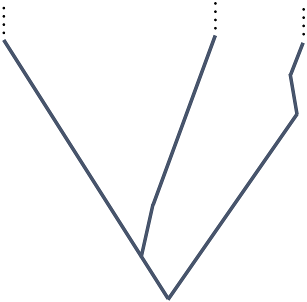
co-c.e.
tree



S

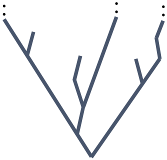
computable
tree

$$[S] = [T]$$

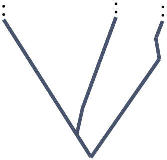


T

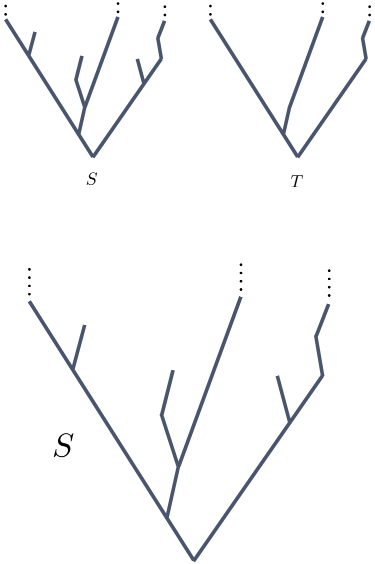
co-c.e.
tree

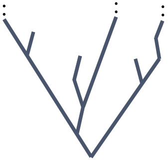


S

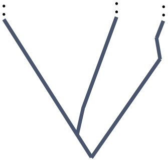


T

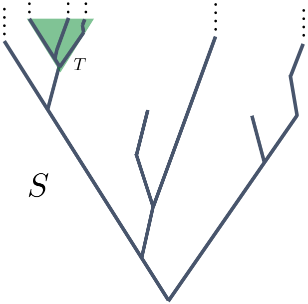

S
T
S

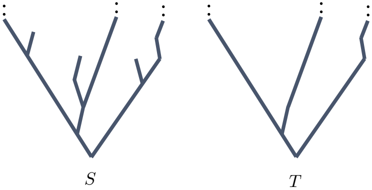


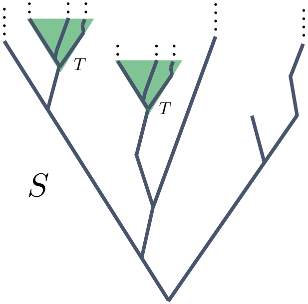
S

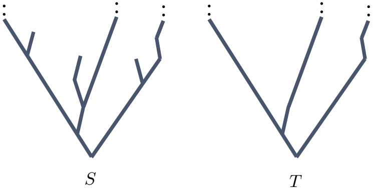


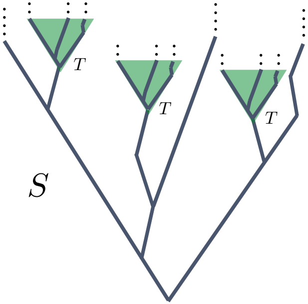
T




S
T


T
T
S


S
T


T
T
T
S

Computing members of deep Π^0_1 classes

What level of randomness $\mathcal{R}$ guarantees that no $\mathcal{R}$-random sequence can compute a member of a deep Π^0_1 class?

The answer is known as *difference randomness*, which is formulated in terms of *difference tests*: a collection of pairs of uniformly effectively open subsets $(\mathcal{U}_i, \mathcal{V}_i)_{i\in\omega}$ of 2^ω such that $\lambda(\mathcal{U}_i \setminus \mathcal{V}_i) \leq 2^{-i}$.

Theorem (Bienvenu, Porter)

If $X \in 2^\omega$ is difference random, then X cannot compute any member of a deep Π^0_1 class.

Note: The difference random sequences are precisely the Martin-Löf random sequences that cannot compute a completion of PA.

3. Examples of deep Π^0_1 classes

Paradigm example: Consistent completions of PA

The following is implicit in work of Levin and Stephan.

Theorem
The Π^0_1 class of consistent completions of PA is a deep class.

What exactly does this tell us?

Not only can we not probabilistically compute some consistent completion of *PA* with positive probability, but we cannot even hope to produce longer and longer initial segments of a consistent completion of *PA* with sufficiently high probability.

Completions of PA proof sketch, 1

To prove that this result, we can consider the class $\mathcal{P}$ of total extensions of a universal partial computable $\{0,1\}$-valued function.

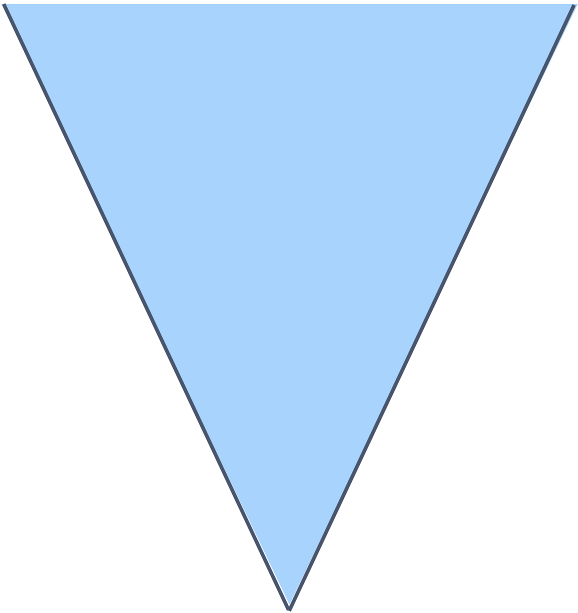
Let $u(\langle e, x\rangle) = \phi_e(x)$, where $(\phi_e)_{e\in\omega}$ is an effective enumeration of all partial computable $\{0,1\}$-valued functions.

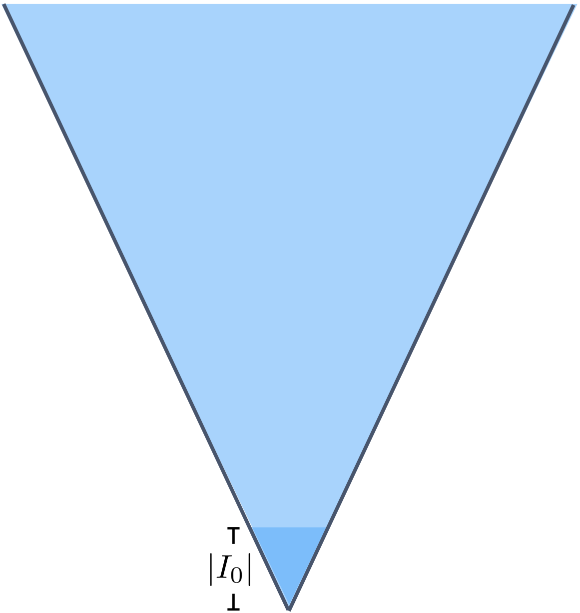
We will define a partial computable $\{0,1\}$-valued function ϕ_e (where we know e in advance by the recursion theorem), and this will allow us to show that $\mathcal{P}$ is deep.

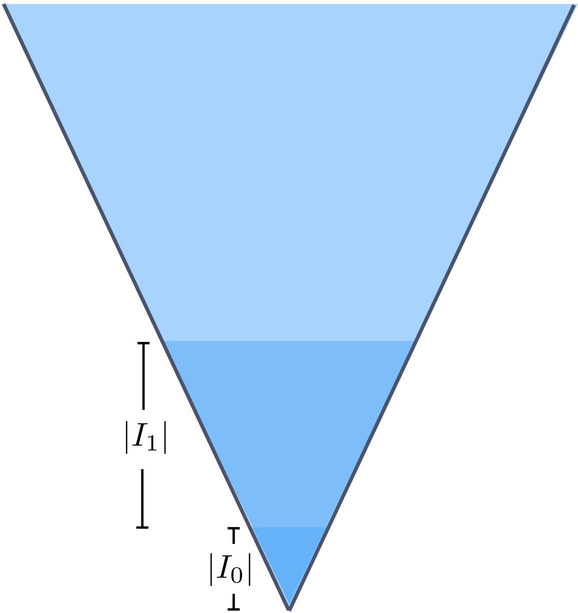
Completions of PA proof sketch, 2

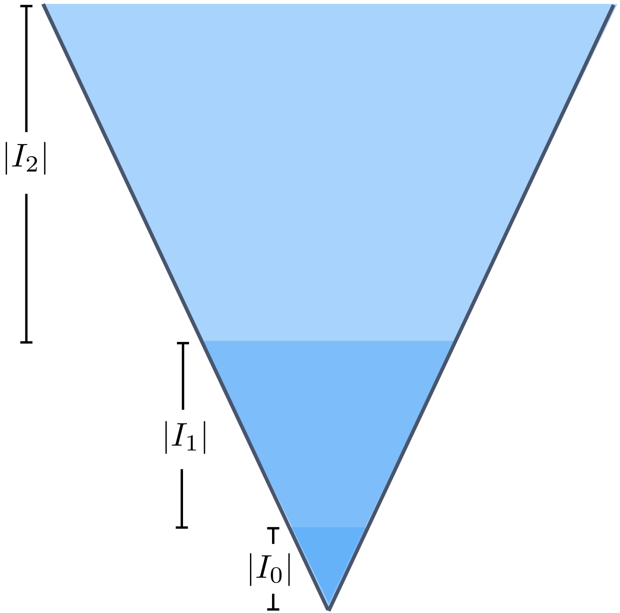
Since we are defining ϕ_e, we have control of the values $u(\langle e, x\rangle)$ for every $x \in \omega$.

Let $(I_k)_{k\in\omega}$ be an effective collection of intervals forming a partition of ω, where we have control of 2^{k+1} values of u inside of I_k for each $k \in \omega$.




$|I_0|$

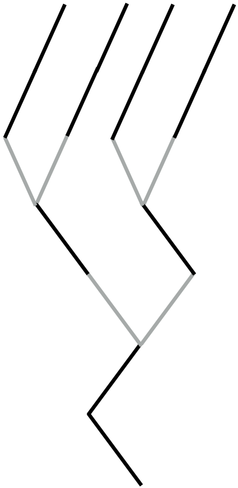

$|I_1|$
$|I_0|$


$|I_2|$
$|I_1|$
$|I_0|$

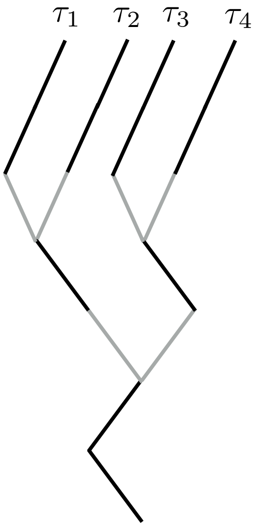
$$u_s = 01 * 0 * 11$$

$$u_s = 01 * 0 * 11$$

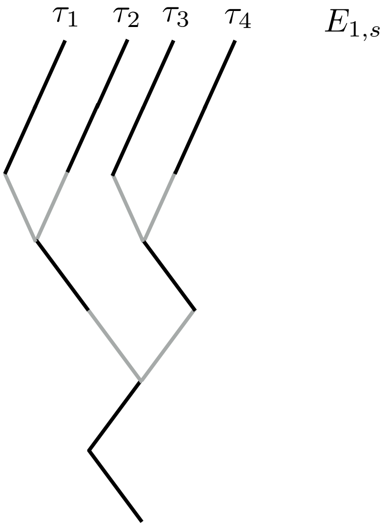
$u_s = 01 * 0 * 11$



$u_s = 01 * 0 * 11$

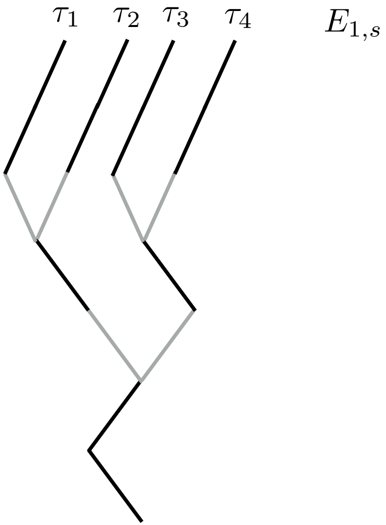


$u_s = 01 * 0 * 11$



$u_s = 01 * 0 * 11$

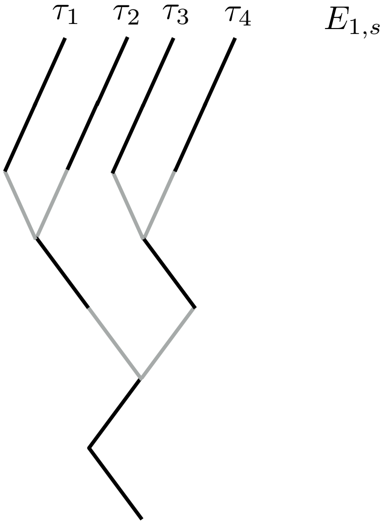
$M_s(E_{1,s}) \geq 1/2$



$$u_s = 01 * 0 * 11$$

$$M_s(E_{1,s}) \geq 1/2$$

$$M_s(\tau_1) + M_s(\tau_2) = 3/8$$

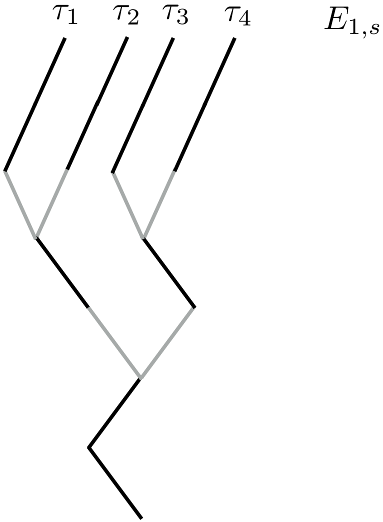


$u_s = 01 * 0 * 11$

$M_s(E_{1,s}) \geq 1/2$

$M_s(\tau_1) + M_s(\tau_2) = 3/8$

$M_s(\tau_3) + M_s(\tau_4) = 1/4$



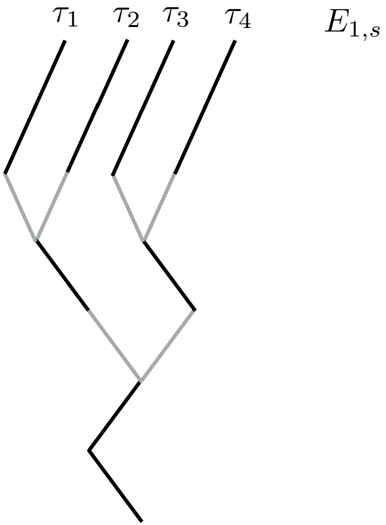
$u_s = 01 * 0 * 11$

$M_s(E_{1,s}) \geq 1/2$

$M_s(\tau_1) + M_s(\tau_2) = 3/8$

$M_s(\tau_3) + M_s(\tau_4) = 1/4$

We want to kill off τ_1 *and* τ_2.



$$u_s = 01 * 0 * 11$$

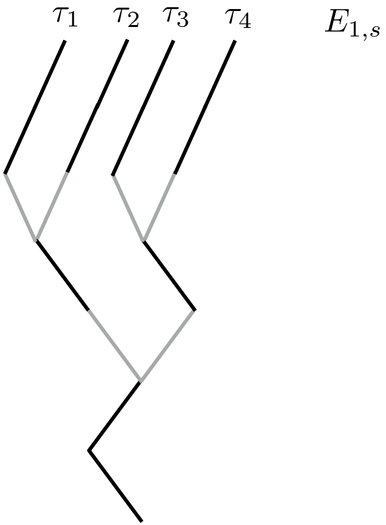
$$M_s(E_{1,s}) \geq 1/2$$

$$M_s(\tau_1) + M_s(\tau_2) = 3/8$$

$$M_s(\tau_3) + M_s(\tau_4) = 1/4$$

We want to kill off τ_1 *and* τ_2.

We set $u_{s+1} = 01\textcolor{red}{1}0 * 11$.



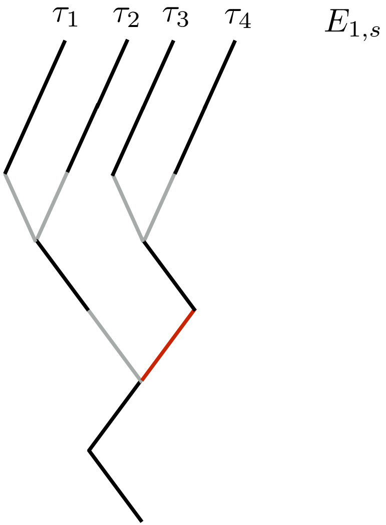
$$u_s = 01 * 0 * 11$$

$$M_s(E_{1,s}) \geq 1/2$$

$$M_s(\tau_1) + M_s(\tau_2) = 3/8$$

$$M_s(\tau_3) + M_s(\tau_4) = 1/4$$

We want to kill off τ_1 *and* τ_2.
We set $u_{s+1} = 0110 * 11$.



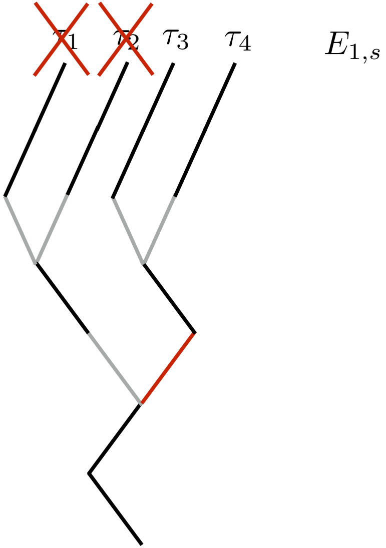
$$u_s = 01 * 0 * 11$$

$$M_s(E_{1,s}) \geq 1/2$$

$$M_s(\tau_1) + M_s(\tau_2) = 3/8$$

$$M_s(\tau_3) + M_s(\tau_4) = 1/4$$

We want to kill off τ_1 *and* τ_2.
We set $u_{s+1} = 01{\color{red}1}0 * 11$.



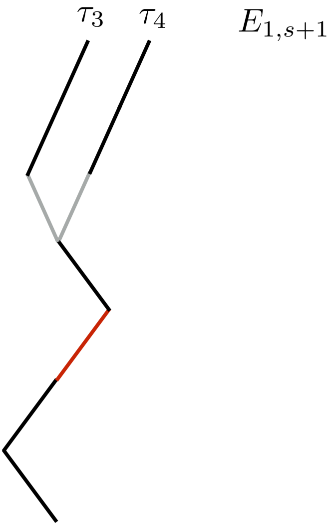
$$u_s = 01 * 0 * 11$$

$$M_s(E_{1,s}) \geq 1/2$$

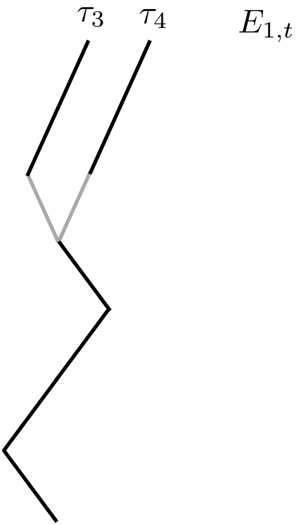
$$M_s(\tau_1) + M_s(\tau_2) = 3/8$$

$$M_s(\tau_3) + M_s(\tau_4) = 1/4$$

We want to kill off τ_1 *and* τ_2.
We set $u_{s+1} = 0110 * 11$.

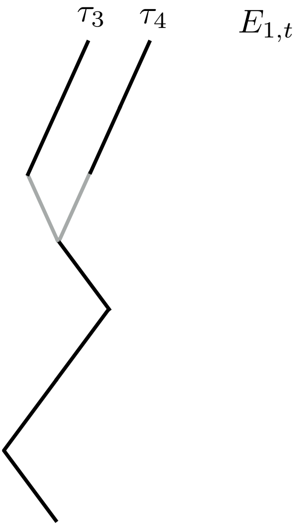


$u_t = 0110 * 11$



$u_t = 0110 * 11$

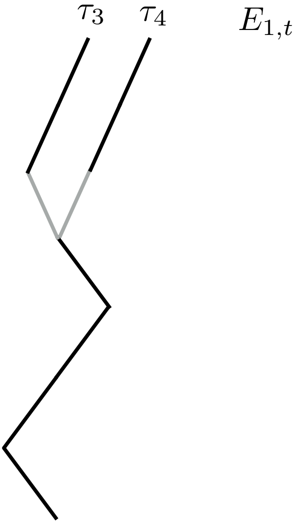
$M_t(E_{1,t}) \geq 1/2$



$$u_t = 0110 * 11$$

$$M_t(E_{1,t}) \geq 1/2$$

$$M_t(\tau_3) = 3/16$$

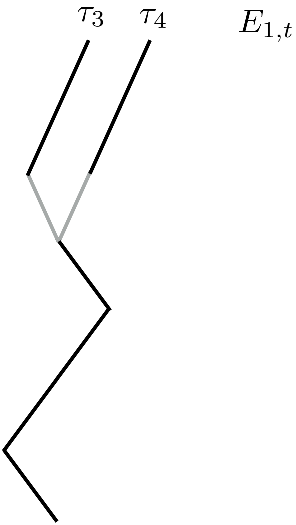


$$u_t = 0110 * 11$$

$$M_t(E_{1,t}) \geq 1/2$$

$$M_t(\tau_3) = 3/16$$

$$M_t(\tau_4) = 5/16$$



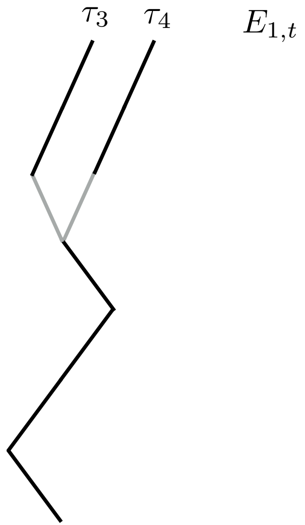
$u_t = 0110 * 11$

$M_t(E_{1,t}) \geq 1/2$

$M_t(\tau_3) = 3/16$

$M_t(\tau_4) = 5/16$

We want to kill off τ_4.



$u_t = 0110 * 11$

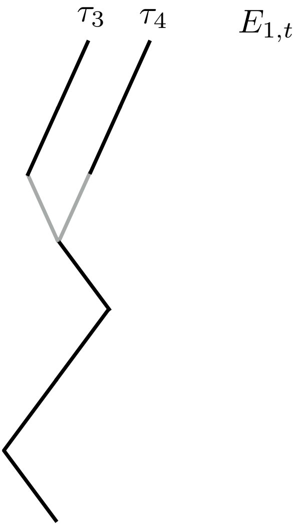
$M_t(E_{1,t}) \geq 1/2$

$M_t(\tau_3) = 3/16$

$M_t(\tau_4) = 5/16$

We want to kill off τ_4.

We set $u_{t+1} = 0110\textcolor{red}{0}11$.



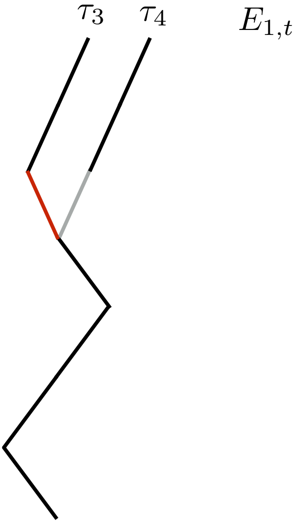
$u_t = 0110 * 11$

$M_t(E_{1,t}) \geq 1/2$

$M_t(\tau_3) = 3/16$

$M_t(\tau_4) = 5/16$

We want to kill off τ_4.
We set $u_{t+1} = 0110\textcolor{red}{0}11$.



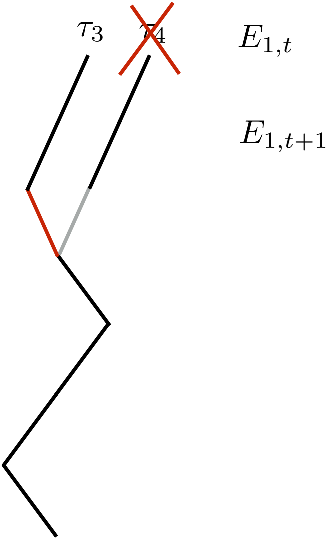
$$u_t = 0110 * 11$$

$$M_t(E_{1,t}) \geq 1/2$$

$$M_t(\tau_3) = 3/16$$

$$M_t(\tau_4) = 5/16$$

We want to kill off τ_4.
We set $u_{t+1} = 0110{\color{red}0}11$.



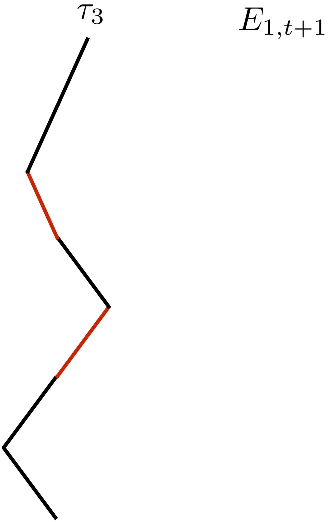
$u_t = 0110 * 11$

$M_t(E_{1,t}) \geq 1/2$

$M_t(\tau_3) = 3/16$

$M_t(\tau_4) = 5/16$

We want to kill off τ_4.
We set $u_{t+1} = 0110011$.



Completions of PA proof sketch, 3

Step 1: For each k, we consider the sets

$$E_{k,s} = \{\sigma \in 2^{<\omega} : \sigma{\upharpoonright}I_k \text{ extends } u_s{\upharpoonright}I_k\},$$

and wait for a stage s such that

$$M(E_{k,s}) \geq 2^{-k}.$$

Completions of PA proof sketch, 4

Step 2: Pick some $y \in I_k$ on which we have yet to define u.

Consider the sets

$$E^0_{k,s}(y) = \{\sigma \in E_{k,s} : \sigma(y) = 0\}$$

and

$$E^1_{k,s}(y) = \{\sigma \in E_{k,s} : \sigma(y) = 1\}.$$

Then $M(E^i_{k,s}(y)) \geq 2^{-(k+1)}$ for $i = 0$ or 1 (or both).

If this holds for $i = 0$, we set $u(y) = 1$; otherwise we set $u(y) = 0$.

Completions of PA proof sketch, 5

We repeat the process, going back to Step 1.

We can repeat the process at most 2^{k+1} times (since we have enough values to work with in I_k).

Eventually, we will get stuck at Step 1.

Setting $f(k) = \max(I_k)$, we will have

$$M(\{\sigma : \sigma \restriction f(k) \text{ extends } u\}) \leq 2^{-k}.$$

That is,

$$M(T_{f(k)}) \leq 2^{-k}.$$

Establishing the depth of a given Π_1^0 class

The technique for showing that the class of consistent completions of PA is deep is what we refer to as a *wait and kill* argument.

We need to work with some object that we have control over in some way.

For example, in the previous proof we define a partial computable $\{0, 1\}$-valued function ϕ using the recursion theorem.

We *wait* to see a sufficiently large collection of oracles compute some possible extension of ϕ (at some place at which ϕ is currently undefined).

We then define ϕ at this place in such a way as to *kill* off each of these oracles.

Shift-complex sequences: the idea

A Martin-Löf random sequence X has high initial segment complexity, satisfying

$$K(X\upharpoonright n) \geq n - O(1).$$

Nonetheless, X will still contain arbitrarily long runs of 0s (since all Martin-Löf random sequences are normal).

That is, certain subwords of X can have fairly low initial segment complexity.

By contrast, a shift-complex sequence is a sequence with the property that every subword has high initial segment complexity.

Shift-complex sequences: the formal definition

For $\delta \in (0,1)$ and $c \in \omega$, we say that $X \in 2^\omega$ is *(δ, c)-shift complex* if

$$K(\tau) \geq \delta|\tau| - c$$

for every subword τ of X.

The following draws upon work of Rumyantsev.

Theorem (Bienvenu, Porter)

For every $\delta \in (0,1)$ and $c \in \omega$, the (δ, c)-shift complex sequences form a deep class.

Diagonally non-computable sequences and randomness

Recall that a sequence X is diagonally non-computable if there is some total function $f \leq_T X$ such that $f(e) \neq \phi_e(e)$ for every e.

Every Martin-Löf random sequence X is diagonally non-computable:

Let $f(e) = X \restriction e$ (coded as a natural number).

Note that $f(e) < 2^{e+1}$.

DNC_h functions

Let h be a computable, non-decreasing, unbounded function.

f is a DNC_h function if

- f is total,
- $f(e) \neq \phi_e(e)$ for every e, and
- $f(e) < h(e)$ for every e.

Theorem (Bienvenu, Porter)

DNC_h is a deep class if and only if $\sum_{n=0}^{\infty} \frac{1}{h(n)} = \infty$.

Moreover, if $\sum_{n=0}^{\infty} \frac{1}{h(n)} < \infty$, then every Martin-Löf random computes a DNC_h function.

Thank you for your attention!