

RANDOMNESS II

CMU MATH LOGIC SEMINAR

JASON RUTE

TUESDAY, NOVEMBER 17, 2009

JOINT SERIES WITH ED DEAN

Review

We are working in Cantor set $2^{\mathbb{N}}$ and are using the "fair-coin" Borel measure μ

We often identify $X \in 2^{\mathbb{N}}$ with the set that it's a characteristic function of.

Computable (Recursive)

A partial function f is computable (recursive) if (intuitively) it can be described by a deterministic algorithm that runs the same way each time, (Church-Turing Thesis)

A set X of natural numbers is computable (recursive) if there is a computable program that "sorts" them, i.e. there exists a total computable function f such that $f(n) = 1$ if $n \in X$ and $f(n) = 0$ otherwise.

The key is that the function is total. It must halt and give an answer either way. Example: the prime numbers are computable. So are the codes for every partial computable function. (see if the code represents a recursive function).

Fact! If a set is computable, then there is a computable f such that $f(n) =$ the n th element of X . (Just test each element in order.) in order

A sequence $X \in 2^{\mathbb{N}}$ and an ^{arithmetical} predicate P are computable (recursive) if the corresponding set is computable.

A set X is computably enumerable (recursively enumerable)

if there is a computable partial function f which halts on and only on elements of X .
The main idea is that one can test that $n \in X$ but not that $n \notin X$. Examples:

the set of (Gödel #'s) of formulas ψ s.t., ψ is provable in Peano arithmetic, is computably enumerable but not computable. X is also referred to as semi-decidable.

Fact: Computable enumerable sets X have a total computable function f s.t. $\text{range } f = X$.
However, the elements may not appear in order.

A set $B \subseteq 2^{\mathbb{N}}$ is computable if there is a computable program that tests whether $X \in B$ or $X \notin B$. By looking at initial segments. Such a set is also be called decidable or Δ_1^0 .

The predicate of B is computable if B is.

↑
where X is its characteristic function in the predicate.

e.g. predicate for sequences with at least two 1's in the first 100 digits.

$$X \in P \iff \exists n_1, n_2 < 100 \dots X(n_1) = 1 \wedge X(n_2) = 1$$

Testing For Effective Openness

Tarski / Kuratowski: Computation

$X \in B \iff P(X)$ where X is characteristic function

B is computable iff P is computable

B is effectively open (Σ_1^0) iff P is Σ_1^0

Note: computable $\subseteq \Sigma_1^0$

Assuming P is computable!

$$X \in B \iff \underbrace{\forall n < 10 \exists m \in \mathbb{Z}^0}_{\text{bounded computation}} \underbrace{P(X, m, n)}_{\text{computable}}$$

computable

$$X \in U \iff \underbrace{\forall n < 10}_{\text{bounded quantification}} \underbrace{\exists m \exists l}_{\Sigma_1^0} \underbrace{P(X, n, m, l)}_{\text{computable}}$$

Σ_1^0

Why it works!

finite object. can be coded.

$$X \in U \iff \exists \sigma, n \text{ } \underbrace{f(n) = \sigma}_{\text{computable}} \text{ and } \sigma \text{ is initial segment of } X$$

$$\iff \exists \sigma, n \text{ } \underbrace{f(n) = \sigma}_{\text{computable}} \wedge \underbrace{\forall k < n}_{\text{bounded quantification}} \underbrace{\sigma(k) = X(k)}_{\text{computable}}$$

Σ_1^0

A set $U \subseteq 2^{\mathbb{N}}$ is effectively open, also called Σ_1^0 , or semi-decidable, if there is a computer program that tests whether $X \in U$ (but may not be able to test whether $X \notin U$) by checking initial segments of X .

More formal definition:

U is effectively open if $U = \bigcup_n [\sigma_n]$

where σ_n are a computably enumerable sequence of finite strings of 0's and 1's and $[\sigma_n] = \{X \in 2^{\mathbb{N}} : \sigma_n \text{ is an initial segment of } X\}$

Also we may assume σ_n are disjoint. (This is because we can skip (or break apart) any σ_n that has had a segment of it come before in our computably enumerable listing.)

A sequence of computable or computably enumerable objects A_n are uniform if there is a single program that generates the codes for each object using n as a parameter.

Example: The sequence of functions

$$f_n \text{ s.t. } f_n = \begin{cases} 1 & \text{if } n = \text{Code\# of } \psi \\ & \text{and } \vdash \psi \\ 0 & \text{otherwise} \end{cases}$$

exists since there are function $g \equiv 1, h \equiv 0$ but the sequence f_n is not uniform.

The sequence $U_n = \{X \in 2^{\mathbb{N}} \mid X \text{ starts with } n \text{ 1's}\}$ is uniform.

Randomness

A Martin-Löf Test is a sequence of uniform effectively open sets V_n s.t.
 $\mu(V_n) < 2^{-n}$.

Note: The 2^{-n} is just some computable rate of convergence. Any computable rate will give the same result.

The intersection $\bigcap V_n$ is called Martin-Löf null or effectively null.

$X \in 2^{\mathbb{N}}$ is Martin-Löf random (ML-Random) or just random if $X \notin \bigcap V_n$ for any Martin-Löf Test V_n .

Fact (From Last Week)

There exists a universal ML Test V_n s.t. $X \notin V_n \iff X$ is ML-Random.

Solovay's Lemma

Thm Let $X \in 2^{\mathbb{N}}$ be random.

Let V_n be uniformly Σ_1^0 subsets of $2^{\mathbb{N}}$ such that $\sum_{n \geq 0} \mu(V_n) < \infty$.

Then $X \in V_n$ for only finitely many n .

Solovay's Lemma vs. ML Test

ML Tests

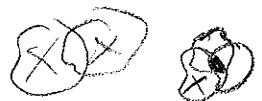
- X is not random if in intersection $\bigcap V_n$
- Rate of convergence (of measure) is 2^{-n} .



Solovay Lemma

- X is not random if in the limit of the union:

$$\lim_k \bigcup_{n \geq k} V_n$$



- Rate of convergence isn't fixed
But it shinks significantly fast.

So we are able to extract a ML Test.

Also note that both specify G_δ sets
(more specifically Π_2^0 sets)

Proof

There is some c s.t. $\sum_{n \geq 0} \mu(V_n) \leq 2^c < \infty$

(Note: There is no way to find this c unless one knows a bound on $\sum_{n \geq 0} \mu(V_n)$)

Let $W_k = \{X \in 2^{\mathbb{N}} : X \in V_n \text{ for at least } k\text{-many } n\}$

Claim:
 W_k is uniformly effectively open (uniformly Σ_1^0)

PF of Claim

$$X \in W_k \Leftrightarrow \exists n_1 < \dots < n_k \quad \underbrace{\forall i \leq k}_{\text{bounded quantification}} \quad \underbrace{(X \in V_{n_i})}_{\Sigma_1^0}$$

$\underbrace{\hspace{15em}}_{\Sigma_1^0}$

Uniform since given the code for $\{V_n\}$ we can, with one program, calculate the code for W_k using k as a parameter.

Claim:

$$\mu(W_k) \leq 2^{-c}/k \text{ for all } k.$$

Assuming this claim, we have

$$\mu(W_{2^{c+k}}) \leq 2^{-c}/2^{c+k} = 1/2^k$$

This forms a ML-test since $W_{2^{c+k}}$ are uniform.

So if X is random, $X \notin \bigcap_k W_{2^{c+k}}$

So $X \notin W_{2^{c+k}}$ for some k .

Proof is complete (mod Claim).

Proof of Claim in Solovay's Lemma

We have $W_k = \{x \in 2^{\mathbb{N}} \mid (\exists \geq k_n) (x \in V_n)\}$

For all s , let

$$W_{k,s} := \{x \in 2^{\mathbb{N}} \mid (\exists \geq k_n \leq s) (x \in V_n)\}$$

Now we have

$$\begin{aligned} 2^c &\geq \sum_{n=0}^{\infty} \mu(V_n) \geq \sum_{n=0}^s \mu(V_n) \\ &= \sum_{n=0}^s \int_{2^{\mathbb{N}}} \mathbb{1}_{V_n}(x) \, dX \end{aligned}$$

$$\begin{aligned} &\left(\begin{array}{l} \text{note: } \sum_{n=0}^s \mathbb{1}_{V_n}(x) \geq k \\ \text{if } x \in W_{k,s} \end{array} \right) &= \int_{2^{\mathbb{N}}} \sum_{n=0}^s \mathbb{1}_{V_n}(x) \, dX \\ &\quad \searrow &\geq \int_{2^{\mathbb{N}}} k \mathbb{1}_{W_{k,s}}(x) \, dX \\ & &= k \mu(W_{k,s}) \end{aligned}$$

But $W_k = \bigcup_{s=0}^{\infty} W_{k,s}$, so

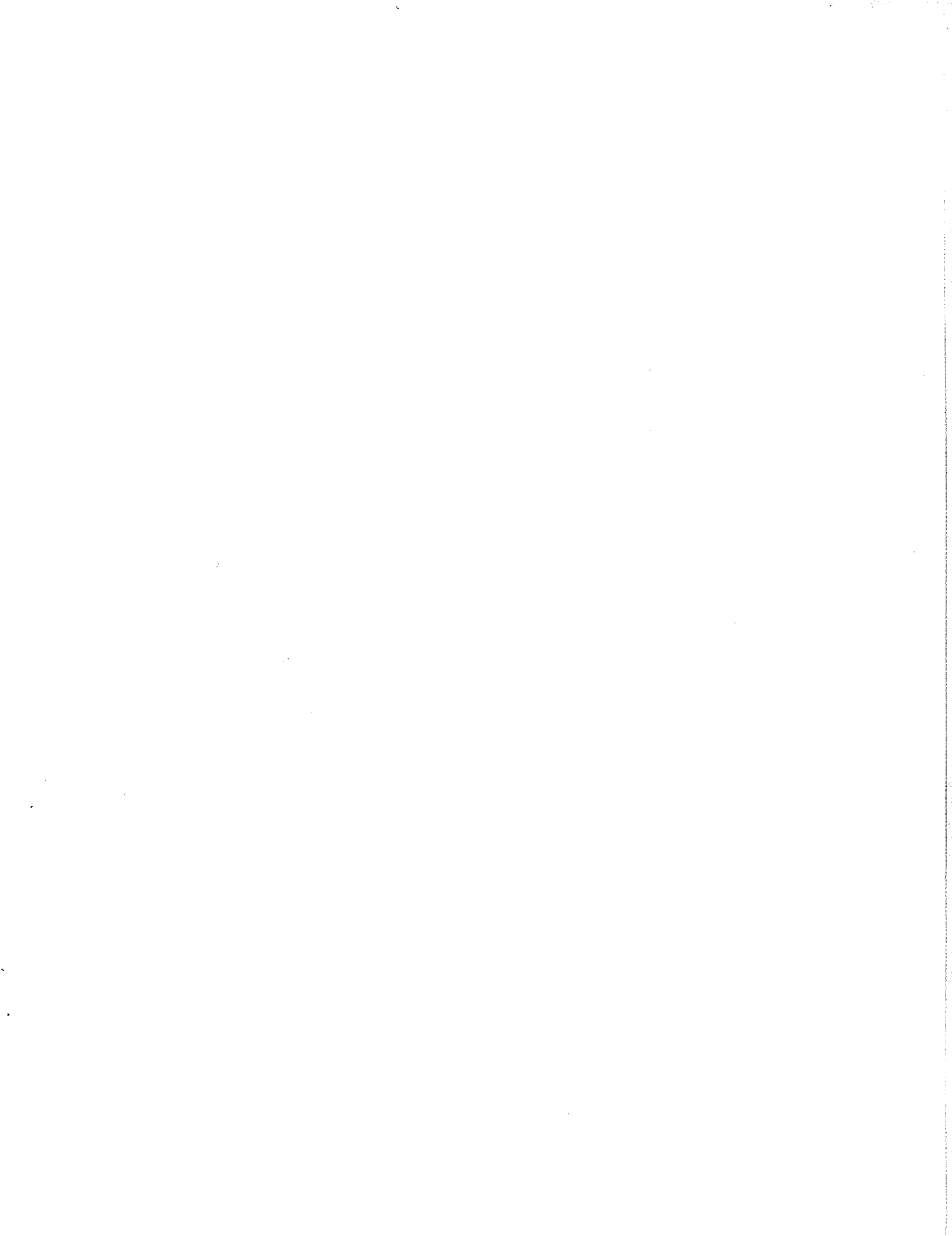
$$\mu(W_k) = \lim_{s \rightarrow \infty} \mu(W_{k,s})$$

Hence $2^c \geq k \mu(W_k)$

□ claim

□ Solovay's
Lemma

Note! This is a recursion theoretic refinement of the Borel/Cantelli lemma in probability theory.



Strong Law of Large Numbers

Thm (Classical Version)

For μ -a.e. $X \in 2^{\mathbb{N}}$

$$\lim_{n \rightarrow \infty} \frac{\sum_{i=0}^{n-1} X(i)}{n} = \frac{1}{2}$$

Thm (Recursion Theoretic)

Replace " μ -a.e. $X \in 2^{\mathbb{N}}$ " with
"every ML-Random $X \in 2^{\mathbb{N}}$ "

Note

Converse obviously doesn't hold

0101010...

pf

Fact (Hoeffding's Inequality)

$$\text{Prob} \left(\left| \frac{\sum_{i=0}^{n-1} X(i)}{n} - \frac{1}{2} \right| > \varepsilon \right) < \frac{2}{e^{2n\varepsilon^2}}$$

under the fair coin probability measure on $2^{\mathbb{N}}$

Let

$$V_n := \left\{ X \in 2^{\mathbb{N}} : \left| \frac{\sum_{i=0}^{n-1} X(i)}{n} - \frac{1}{2} \right| > \sqrt{\frac{\log n}{n}} \right\}$$

Observations about V_n :

(a) It's effectively open (Σ_1^0)

only need to know 1st n digits of X

$$X \in V_n \Leftrightarrow \exists k \left| \frac{\sum_{i=0}^{n-1} X(i)}{n} - \frac{1}{2} \right| > \sqrt{\frac{\log n}{n}}$$

rounded up at the k -th decimal

computable

Σ_1^0 (actually computable since LHS is rational and RHS is irrational so they differ at some decimal)

(b) It's uniform

Testing whether $X \in V_n$ can be done with one program that has n as a parameter

(c) $\sqrt{\frac{\log n}{n}} \rightarrow 0$ so if $\lim_{n \rightarrow \infty} \frac{\sum_{i=0}^{n-1} X(i)}{n} \neq \frac{1}{2}$

then $X \in V_n$ for infinitely-many V_n .

(d) $\sum_{n=0}^{\infty} \mu(V_n) \leq \sum_{n=0}^{\infty} \frac{2}{n^2} < \infty$

↑
Hoeffding's Inequality

So by (a), (b), (c) and Solovay's Lemma if X is ML-random, then X satisfies Strong Law of Large Numbers.

Joins and Randomness of Substrings

Def

If $A, B \in 2^{\mathbb{N}}$ and $A = A_1 A_2 A_3 \dots$ and $B = B_1 B_2 B_3 \dots$
← third digit of A

then $A \oplus B = A_1 B_1 A_2 B_2 A_3 B_3 \dots$

(This is a specific way to join the information contained in A with the information contained in B.)

Thm

If X is ML-random and $X = A \oplus B$, then A and B are both ML-Random

Pf

Assume for contradiction A is not ML-Random. Then $A \in \bigcap V_n$ for some ML-test V_n .

Let $U_n = \{Y \in 2^{\mathbb{N}} : Y = C \oplus D \text{ for } C \in V_n\}$.

Observations on U_n :

Ⓐ U_n is uniformly effectively open.

$$Y \in U_n \iff \underbrace{\exists \sigma, \tau : |\sigma| = |\tau| \wedge [\sigma] \subseteq V_n}_{\Sigma_1^0} \wedge \underbrace{\sigma \oplus \tau \prec Y}_{\text{computable}}$$

Σ_1^0

$\sigma_0 \tau_0 \sigma_1 \tau_1 \dots$
is initial segment of Y

Ⓑ $\mu(U_n) = \mu(V_n) \leq 2^{-n}$ ← Exercise for reader. (Consider how each $[\sigma] \subseteq V_n$ changes)

So \mathcal{U}_n is a ML-test and
 assuming $A \in \bigcap V_n$ then $X = A \oplus B \in \bigcap \mathcal{U}_n$
 A contradiction. So A is ML-Random.

The same argument goes for B . \square

Note

The converse does not hold.

Take X random.

Consider $X \oplus X$ and

the ML-test $V_n = \left\{ Y \in 2^{\mathbb{N}} : Y(2i) = Y(2i+1) \text{ for } i < 2^n \right\}$

It is ML-test

$$\bullet Y \in V_n \iff \underbrace{\forall i < 2^n}_{\text{bounded quantification}} \underbrace{Y(2i) = Y(2i+1)}_{\text{computable}}$$

$\underbrace{\hspace{15em}}_{\text{computable (so } \Sigma_1^0)}$

$$\bullet \mu(V_n) = \frac{1}{2^n}$$

So we need a notion of "independence"
 for ML-Randoms.

Relativization and Oracles

Each $X \in 2^{\mathbb{N}}$ contains information.

For example there is some $K \in 2^{\mathbb{N}}$

such that $K(e) = 1 \iff \varphi_e(e)$

(where φ_e is some enumeration of partial computable functions)

Also there is some $G \in 2^{\mathbb{N}}$ where

$G(\ulcorner \psi \urcorner) = 1$ iff Peano arithmetic $\vdash \psi$

(where $\ulcorner \psi \urcorner$ is ψ 's Gödel number).

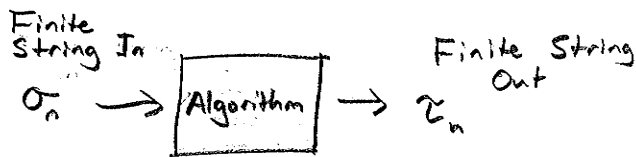
Unlike, for example, the binary digits of π , the above examples are not computable.

However, we can ask, "What if I know the string K ? Then what can I compute?"

For example, if I know enough of the values of K I could compute $G(\ulcorner \psi \urcorner)$ for any sentence ψ and if I know enough of the values of G , I could compute $K(e)$ for any e .

This is called computing relative to an oracle. If we can compute Y using X we say $Y \leq_T X$, read "Y is Turing reducible to X".

Finite way to think about it

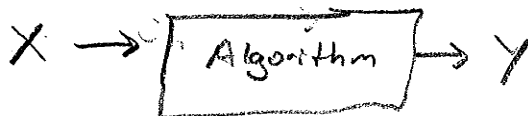


Must have $\sigma_n \subseteq \sigma_m \Rightarrow \tau_n \subseteq \tau_m$

Infinite way to think about it

(Finite approx of) Infinite String In

(Finite approx of) Infinite String Out



Must be able to approximate Y with approximations of X

What does fit this pattern

$X \mapsto X \oplus X$

$X \mapsto$ compliment of X (change 0 to 1, 1 to 0)

$X \mapsto$ binary digits of π (just ignore X)

$X \mapsto \begin{cases} 11111\dots & \text{if } X \in U \\ \text{doesn't halt} & \text{otherwise} \end{cases}$ (U is a specific effectively open set)

What doesn't fit this pattern

$X \mapsto \begin{cases} 000\dots & \text{if } X \notin U \\ 111\dots & \text{if } X \in U \end{cases}$

$X \mapsto \begin{cases} 000\dots & \text{if } X \text{ has } \infty\text{-many } 1\text{'s} \\ \text{doesn't halt} & \text{otherwise} \end{cases}$

Randomness Relative to an Oracle

Every concept so far can be done relative to an oracle A

- Effectively open (Σ_1^0) \rightarrow Effectively Open Relative to A $(\Sigma_1^{0,A})$
- Uniform \rightarrow Uniform relative to A
- ML-Test \rightarrow ML-Test relative to A
- Universal ML-Test \rightarrow Universal ML-Test Relative to A
- (ML-)Random \rightarrow A -Random

This gives us a way to express independence, namely B is A -random and A is B -random,

Thm (Van Lambalgen's Theorem)

TFAE

- ① $A \oplus B$ is random
- ② A is random and B is A -random
- ③ B is random and A is B -random

PR Omitted

Thm (Miller / Yu 2004)

If A is random and $A \leq_T B$ where B is C -random. Then A is C -random.

PR Omitted (Difficult)

