# Outline of Lecture 3

## The Computational Power of Randomness

- Randomness and Recursive Enumerability.
- Randomness and Completeness.
- Extracting Information from Random Sequences.
- Relative Randomness.
- Van Lambalgen's Theorem.
- Higher Randomness and Lowness.

# Randomness and Recursive Enumerability

## Theorem [Barzdins]

Let $Z$ be recursively enumerable. Then for all $n$,

$$C(Z\restriction_n) \leq^+ 2\log n.$$

Proof

- To output $Z\restriction_n$, the following information suffices:
  - The length $n$ ($\log n$ bits),
  - the number $k_n$ of elements in $Z$ that are $\leq n$ ($\log n$ bits), and
  - the index $e$ such that $Z = W_e$ (constant number of bits).
- Given this, run $W_e$ till exactly $k_n$ elements $\leq n$ have been enumerated.
- Since we know descriptions of $n$ and $k_n$ have exactly $\log n$ bits, we can just concatenate them.

# Randomness and Recursive Enumerability

Since $K(x) \leq C(x) + 2 \log |x|$, it follows that for $Z$ recursively enumerable,

$$(\forall n)\ [K(Z \upharpoonright_n) \leq^+ 4 \log n].$$

Hence by Schnorr's Theorem, recursively enumerable sequences cannot be random.

# Randomness and Recursive Enumerability

Question: Are random sequences always computationally very difficult?

Consider $\Omega$. It was defined as

$$\Omega = \sum_{\sigma \in \mathrm{dom}(S)} 2^{-|\sigma|}.$$

$\Omega$ is left-computable, since every new $\sigma$ that shows up in $\mathrm{dom}(S)$ will increase the current approximation to $\Omega$.

It turns out $\Omega$ is computationally as difficult as the halting problem $\emptyset'$, where

$$\emptyset' = \{n : \varphi_n(n) \downarrow\}.$$

# Randomness and Completeness

**Theorem**

$\Omega$ is Turing-equivalent to the halting problem $\emptyset'$, $\Omega \equiv_T \emptyset'$.

Proof

- If we know $\emptyset'$, we can compute $\Omega \upharpoonright_n$ uniformly in $n$, by enumerating the left cut of $\Omega$ and using $\emptyset'$ to ask whether the current approximation is close enough.
- Hence $\Omega \leq_T \emptyset'$.

# Randomness and Completeness

- Now show $\emptyset' \leq_T \Omega$.
- Define a prefix-free machine $M$ by letting $M(0^n1) = s$ if $n$ enters $\emptyset'$ at stage $s$.
- Let $d$ be the coding constant for $M$ with respect to $S$.
- Using an oracle for $\Omega$, we can decide whether $n \in \emptyset'$ as follows.
  - Compute a stage $t$ such that $\Omega \upharpoonright_{n+d+1} = \Omega_t \upharpoonright_{n+d+1}$.
  - $n$ cannot enter $\emptyset'$ after stage $t$, since this would add an element of length $\leq |n + d + 1|$ to the domain of $S$, and hence change $\Omega$ below $n + d + 1$.
  - Hence $n \in \emptyset'$ iff it has been enumerated into $\emptyset'$ by stage $t$.

# Randomness and Completeness

It turns out every other left-computable random real has the same computational difficulty as $\Omega$.

**Theorem**

If $\alpha$ is a left-computable random real, then $\alpha \equiv_{\mathrm{T}} \Omega \equiv_{\mathrm{T}} \emptyset'$.

Question: What about non-left-computable reals?

- We know there are measure 1 (and hence uncountably) many random sequences, but every Turing degree has only countably many members, hence there must be random sequences that are not T-equivalent to the halting problem.
- Do they have to be more complicated, i.e. does it hold that $\emptyset' <_{\mathrm{T}} X$?

# Randomness and Lowness

Recall that the Turing jump of a set $X \subseteq \mathbb{N}$ is defined as

$$X' = \{n : \varphi_n^X(n) \downarrow\}.$$

The jump of $X$ is the halting problem relative to $X$.

**Definition**

A set $X$ is low if $X' \leq_T \emptyset'$

A low set $X$ can be thought of as close to computable since their jump has the same computational power as computational sets.

- In particular, no low set can compute the halting problem.

# Randomness and Lowness

**Theorem**

There exists a low random sequence.

The result follows from a central result of computability theory, the Low Basis Theorem.

A set $\mathcal{A} \subseteq 2^{\mathbb{N}}$ is effectively closed if its complement is effectively open, i.e. if there exists an r.e. set $W \subseteq 2^{<\mathbb{N}}$ such that

$$X \notin \mathcal{A} \quad \Leftrightarrow \quad (\exists \sigma \in W)\ [X \in [\![\sigma]\!]]$$

Alternatively, effectively closed sets can be described as infinite paths through computable trees.

Effectively closed sets are also called $\Pi^0_1$-classes.

# Randomness and Lowness

## Low Basis Theorem [Jockusch and Soare]

Every effectively closed set of sequences contains a low element.

We show that this implies the existence of low random sequences.

Observation: The set of all random sequences is an effective union of effectively closed sets, namely the complements of the effectively open sets given by the levels $\mathcal{U}_n$ of the universal ML-test.

Hence we can take e.g. the first level $\mathcal{U}_1$ and consider the complement of $[\![\mathcal{U}_1]\!]$. This consists of only random sequences.

By the Low Basis Theorem, one of these sequences must be low.

# Extracting Information from Random Sequences

Nevertheless, the set of all random sequences contains enough information to compute any other sequence.

**Theorem [Kucera; Gacs]**

Every sequence is Turing reducible to a random one.

In other words, for any sequence $X$ we can find a random sequence $R$ and an algorithm that can, with access to the sequence $R$, compute $X$.

*We can code information into a ML random sequnce.*

# Extracting Information from Random Sequences

## Lemma

Given rational $\delta > 1$, $k \in \mathbb{N}$, we can compute a length $l(\delta, k)$ such that for any martingale $F$ and any $\sigma$,

$$\left|\{\tau \in \{0,1\}^{l(\delta,k)} : F(\sigma\tau) \leq \delta F(\sigma)\}\right| \geq k.$$

This is an application of Kolmogorov's inequality.

The lemma effectively guarantees a number of paths extending $\sigma$ along which $F$ does not gain much money.

# Extracting Information from Random Sequences

Proof of Kucera-Gacs [due to Merkle and Mihailovic]

- Let $F$ be the left-computable martingale corresponding to the universal ML-test $(U_n)$.
- Hence $X$ is random iff $F$ does not succeed on $X$, in fact iff $F$ does not succeed with $\lim = \infty$.
- Let $r_0 > r_1 > \dots$ be a sequence of rationals such that $\beta_i = \prod_{j \leq i} r_j$ converges.
- Let $l_s = l(r_s, 2)$. Hence for any $\sigma$ there are at least two extensions of length $l_s$ with $F(\sigma\tau) \leq r_s F(\sigma)$.
- Idea: If we choose among these extensions, $F$ will stay bounded and hence we get a random sequence.
- Now code $X$ into one of these paths, $R$.

# Extracting Information from Random Sequences

- Partition $\mathbb{N}$ into intervals $\{I_s\}$, $|I_s| = l_s$.
- Construct random $R$ in stages. At stage $s$, specify $R$ on $I_s$. Start with the empty string $\epsilon$.
- Suppose $\sigma_s = R\restriction_{m_s}$, where $m_s = \sum_{i<s} l_i$, is given, with $F(\sigma_s) \leq \beta_{s-1}$.
- Call $\tau$ of length $l_s$ $s$-admissible if

$$F(\sigma_s\tau) \leq \beta_s = r_s\beta_{s-1}.$$

  By choice of $l_s$, there are at least two $s$-admissible strings.
- Let $\tau_0, \tau_1$ be the leftmost and the rightmost, resp., $s$-admissible strings.
- Set $\sigma_{s+1} = \sigma_s\tau_i$, where $i = X(s)$.

# Extracting Information from Random Sequences

By construction, $R$ is random.

We show how to extract $X$ from $R$ effectively.

- We compute $X(s)$ from $\sigma_{s+1} = R\restriction_{m_s}$.
- $\sigma_{s+1}$ is either the leftmost or the rightmost $s$-admissible extension of $\sigma_s$.
- Observation: Being $s$-admissible is co-r.e., i.e. we can enumerate the extensions which are not $s$-admissible.
- Hence we can wait till either to the left or to the right of $\sigma_{s+1}$ there are no more $s$-adm. extensions left.
- If to the left: $X(s) = 0$.
  If to the right: $X(s) = 1$.

# Extracting Information from Random Sequences

The Kucera-Gacs Theorem says that it is possible to effectively code information into random sequences.

One can analyze the proof and deduce that if a sequence computes the halting problem $\emptyset'$, then it is Turing equivalent to a random sequence.

Hence random sequences can be computationally very powerful.

One may object that this does not agree with our intuition of "true randomness".

We will see that these phenomena can be avoided by making tests algorithmically stronger.

# Relative Randomness

Test can be made stronger if we give them access to an additional oracle.

## Definition

Given a sequence $Z$, we say $(W_n)$ is a ML-test relative to $Z$ (or ML-$Z$-test), if $(W_n)$ is uniformly r.e. in $Z$ and for all $n$, $\sum_{W_n} 2^{-|\sigma|} \leq 2^{-n}$.

$X$ is random relative to $Z$, or simply $Z$-random, if it is not covered by any ML-$Z$-test.

# Relative Randomness

Of particular interest are randomness tests relative to instances of the Turing jump $\emptyset^{(n)}$.

$X$ is called $n$-random if it is random relative to $\emptyset^{(n-1)}$. Hence 1-randomness coincides with ML-randomness.

Obviously, if $X$ is $n$-random, then $X$ is also $k$-random for all $k < n$.

In general, if $X$ is $Z$-random and $Y \leq_T Z$, then $X$ is $Y$-random, too.

# Randomness of Joins

Given two sequences $X, Y$, we can form a new sequence $Z = X \oplus Y$, the join of $X$ and $Y$ by letting

$$Z = X(0)\, Y(0)\, X(1)\, Y(1)\, X(2)\, \ldots$$

What happens if we join two random sequences? Will the join be random, too?

- Certainly not if we join a random sequence with itself, $X \oplus X$.

  There is an easy betting strategy succeeding on all sequences of the form $X \oplus X$.

# Van Lambalgen's Theorem

**Theorem [Van Lambalgen]**

$X \oplus Y$ is ML-random iff $X$ is random and $Y$ is $X$-random.

The proof is an application of Fubini's Theorem from measure theory.

Note that the theorem also says that if we split a random sequence into two halves, we get two sequences that are random relative two each other.

- If $X$ is $Y$-random then $X$ is not computable in $Y$.
- Hence if we split a random sequence, the halves are computationally incomparable: neither half computes the other.

# Higher Randomness and Lowness

We want to show that higher random sequences cannot be computationally powerful anymore.

**Definition**

A sequence $X$ is generalized low, $GL_1$, if $X' \equiv_T X \oplus \emptyset'$.

Being $GL_1$ means that the jump is as low as possible (since always $X' \geq_T X \oplus \emptyset'$).

- Note: If $X \leq_T \emptyset'$ and $X$ $GL_1$, then $X$ is low.

# Higher Randomness and Lowness

**Lemma**

*If $X \leq_T \emptyset'$ and $X$ is $Z$-random, then $Z$ is $GL_1$.*

Proof

- By the Limit Lemma, there exists a computable approximation $X_s \to X$. Let $f$ be the settling function, i.e. $f(n)$ is the minimal $s$ such that $X_s \upharpoonright_n = X \upharpoonright_n$.
- Define $U_e = \{X_{s_e} \upharpoonright_{e+1}\}$ if $\varphi_e^Z(e) \downarrow$ in $s_e$ steps ($\emptyset$ otherwise).
- Then $(V_n)$ with $V_n = \bigcup_{e>n} U_e$ is a ML-$Z$-test.
- Since $X$ is $Z$-random, it is covered by only finitely many $V_n$.
- Hence $X_s \upharpoonright_n$ changes after $s_e$ for almost all $e$ (if exists).
- This implies $f(e) \geq s_e$ for all but finitely many $e$.
- Hence we can decide $Z'$ from $f \leq_T \emptyset'$ and $Z$, by checking $\varphi_e^Z(e)$ up to time $f(e)$.

# Higher Randomness and Lowness

**Corollary**

*If $\Omega$ is $Z$-random, then $Z$ is $GL_1$.*

We can combine this with van Lambalgen's Theorem to prove an alternative characterization of 2-randomness:

$X$ is 2-random iff $X$ is random and $\Omega$ is $X$-random.

Proof: Since $\Omega \equiv_T \emptyset'$, $X$ is 2-random iff $X$ is random relative to $\Omega$. By van Lambalgen, this is equivalent to $X \oplus \Omega$ being random. Again by van Lambalgen, this is equivalent to $X$ random and $\Omega$ $X$-random.

# Higher Randomness and Lowness

Now we can prove the desired result.

**Theorem**

*If $X$ is 2-random then it is $\mathrm{GL}_1$.*

Proof

- If $X$ is 2-random, then by the previous result, $\Omega$ is $X$-random.
- Since $\Omega \leq_T \emptyset'$, it follows from the lemma that $X$ is $\mathrm{GL}_1$.

Note that no $\mathrm{GL}_1$ set can compute the halting problem.