Lowness properties and cost functions

André Nies
The University of Auckland

ASL Annual Meeting, May 2009

Part 1 :

The interaction of computability and randomness

The complexity aspect of a set

We study sets of natural numbers $A \subseteq \mathbb{N}$ (simply called *sets*). We want to understand their computational complexity.

- Absolute complexity: we introduce classes such as

$$\text{computable} \subset \text{low} \subset \Delta^0_2 \ldots$$

  and locate the set A in one of the classes.
- Other classes of shared complexity might be incompatible with them. An example is being computably dominated: every function f computable relative to A is dominated by a computable function.
- Relative complexity: we compare sets A and B using a reducibility such as Turing $\leq_T$.

The randomness aspect of a set

(a) 00000000 00000000 00000000 00000000 0000...
(b) 10100100 01000010 00001000 00010000 0001...
(c) 00100100 00111111 01101010 10001000 1000 ...
(d) 10010100 00010001 11110100 00101101 1111 ...
(e) 11101101 01111010 10101111 11001110 1110 ...

(a) Only zeros
(b) $\prod_i 0^i 1$
(c) $\pi - 3$ in binary
(d) Coin tossing
(e) Coin tossing

Randomness theory

- For the absolute randomness aspect of a set, one introduces a hierarchy of randomness notions.
- The central notion is Martin-Löf-randomness, based on a computably enumerable test concept.
- Others notions can often be viewed as variants of Martin-Löf-randomness. For instance, we have

  weakly 2-random $\Rightarrow$ ML-random $\Rightarrow$ Schnorr random.

- The relative randomness aspect of sets has been studied to a lesser extent. One asks: when is a set B “more random” than a set A?

Applying computability to randomness I

- The formal definition of randomness notions relies on computability theoretic tools.
- We study them with computability theoretic methods.

For instance, consider the definition of Martin-Löf-randomness.
Sets are elements of Cantor space $2^{\mathbb{N}}$.
Let λ denote the uniform (product) measure on $2^{\mathbb{N}}$.

- A ML-test is a uniformly computably enumerable sequence $(G_m)_{m \in \mathbb{N}}$ of open sets such that $\lambda G_m \leq 2^{-m}$ for each m.
- A set Z is ML-random if Z passes each ML-test, in the sense that $Z \notin \bigcap_m G_m$.

A ML-random set Z can be low ($Z' \equiv_T \emptyset'$), but it can also be Turing complete ($Z \equiv_T \emptyset'$).

Applying computability to randomness II

- A ML-test is a uniformly computably enumerable sequence $(G_m)_{m\in\mathbb{N}}$ of open sets such that $\lambda G_m \leq 2^{-m}$ for each m.
- $(G_m)_{m\in\mathbb{N}}$ is a generalized ML-test if the condition "$\lambda G_m \leq 2^{-m}$ for each m" is weakened to $\lim_m \lambda G_m = 0$.
  Such tests are equivalent to null Π^0_2 classes.
- We say that Z is weakly 2-random if Z is in no null Π^0_2 class.

Theorem (Hirschfeldt, Miller 06)

Let Z be ML-random. Then
Z is weakly 2-random $\Leftrightarrow$ each computably enumerable set Turing below Z is computable $\Leftrightarrow$
Z and $\emptyset'$ form a minimal pair.

Applying randomness to computability

Randomness-related concepts enrich computability theory.

- New examples:
  - Chaitin's halting probability Ω, a left-c.e. real.
  - the class of K-trivial sets, a natural Σ^0_3 ideal in the Δ^0_2 Turing degrees.
- New methods: cost functions as a way to understand injury-free solutions to Post's problem.
- New results: purely computability-theoretic classes can be characterized via randomness.

Part 2 : Lowness properties of Δ^0_2 sets

Three ways to be almost computable

We will use randomness to study lowness properties of Δ^0_2 sets. There are three ways in which a Δ^0_2 set A can be almost computable:

- Weak as an oracle:
  A does not provide much computational power as an oracle set. For instance, A is low, namely $A' \leq_T \emptyset'$.
- Easy to compute:
  in some sense, the class of sets computing A is large.
- Approximable with few mind changes:
  $A(x) = \lim_s A_s(x)$ for a computable approximation $(A_s)_{s \in \mathbb{N}}$ such that the total amount of changes is small. (We will introduce cost functions to measure this.)

New lowness properties

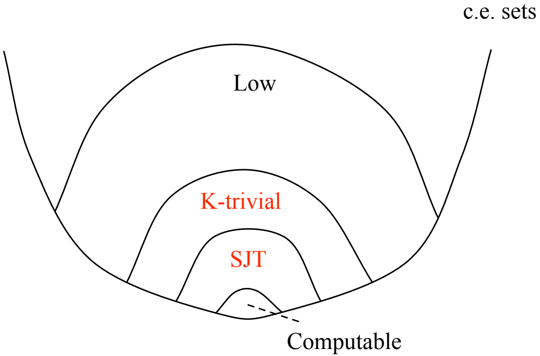
- Till about 2000, the usual lowness $A' \leq_T \emptyset'$ was the most restrictive property studied that says “almost computable”.
- Recently, two interesting classes inside the low sets have emerged: K trivial sets, and strongly jump traceable sets.
- The classes have many characterizations, of all three types: weak as an oracle/ easy to compute/ few mind changes.
- The classes have nice properties:
  - they induce ideals in the Turing degrees (in the computably enumerable degrees, at least);
  - there is a natural, injury-free construction of a c.e. incomputable (even promptly simple) member.

Two classes inside Low

The two classes are:

- The *K*-trivial sets.
  Equivalently, the sets that are low for ML-randomness.
- The strongly jump traceable sets.

Within the c.e. sets we have this picture:



Part 3: K-triviality

Machines and K

Let $\{0,1\}^*$ be the strings over $\{0,1\}$. A machine is a partial recursive function $M : \{0,1\}^* \mapsto \{0,1\}^*$.

M is prefix free if its domain is an antichain under inclusion of strings.

Let $(M_d)_{d \geq 0}$ be an effective listing of all prefix free machines. The standard universal prefix free machine $\mathbb{U}$ is given by

$$\mathbb{U}(0^d 1\sigma) = M_d(\sigma).$$

The prefix free version $K(y)$ of descriptive string complexity (aka Kolmogorov complexity) is the length of a shortest prefix free description of y:

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

K-triviality

- A set A is K-trivial (Chaitin, 1975) if each initial segment has minimal prefix free complexity, namely, it is no greater than the one of its length.
- More precisely, there is $c \in \mathbb{N}$ such that

$$\forall n \, K(A \upharpoonright_n) \leq K(n) + c.$$

- Chaitin showed: computable $\Rightarrow$ K-trivial $\Rightarrow$ Δ^0_2.
- Solovay built an incomputable K-trivial.
- Schnorr's Theorem:

$$Z \text{ is ML-random iff } \forall n \, K(Z \upharpoonright_n) \geq n - c \text{ for some } c.$$

  So being K-trivial says that A is far from random.
- It is not clear why this should be a lowness property at all.

Weak as an oracle: low for ML-randomness

- *A* is low for ML-randomness if each ML-random set is already ML-random relative to *A* (Zambella, 1990).
- This says that *A* is weak as an oracle: *A* cannot find new "regularities" in any ML-random set.

Theorem (Nies 05, Hirschfeldt)

A is K-trivial $\Leftrightarrow$ *A is low for ML-randomness.*

"$\Rightarrow$" uses the golden run method.

Bases for ML-randomness

We say that A is a base for ML-randomness(Kučera, 1993) if

$$A \leq_T Z \text{ for some } Z \in \mathrm{MLR}^A.$$

That is, A can be computed from a set that is random relative to it. This says that the class of sets computing A is large (in a sense relative to A itself).
Kučera proved that some (promptly) simple set is a base for ML-randomness.

Coincidence of “base for ML” with K-trivialilty

- The Kučera-Gács Theorem says that for each set A, there is a ML-random Z such that $A \leq_T Z$.
- So, if A is low for ML-randomness then A is a base for ML-randomness.
- We already know that K-trivial $\Rightarrow$ low for ML-randomness $\Rightarrow$ base for ML-randomness.

  The following then shows that all three classes coincide.

Theorem (Hirschfeldt,Nies, Stephan 07)

Each base for ML-randomness is K-trivial.

Part 4 : Cost functions

We head for a characterization of K-triviality saying that the set A can be computably approximated with a small total amount of mind changes.

Definition of cost functions

Definition

A cost function is a computable function

$$c : \mathbb{N} \times \mathbb{N} \to \{x \in \mathbb{Q} \colon\ x \geq 0\}.$$

We view $c(x, s)$ as the cost of changing $A(x)$ at stage s.

Definition

We say that a computable approximation $(A_s)_{s \in \mathbb{N}}$ obeys a cost function c if

$$\infty > \sum_{x,s} c(x, s)\, [\![x < s \;\&\; x \text{ is least s.t. } A_{s-1}(x) \neq A_s(x)]\!].$$

Basic existence theorem

For a cost function $c : \mathbb{N} \times \mathbb{N} \to \mathbb{Q}$, let $c(x) = \sup_s c(x, s)$.
We say that c has the limit condition if $\lim_x c(x) = 0$.

Theorem (Various)

If a cost function c has the limit condition, then some (promptly) simple set A obeys c.

Proof. Let W_e be the e-th c.e. set. If W_e is infinite we want some $x \in W_e$ to enter A. We define a computable enumeration $(A_s)_{s \in \mathbb{N}}$ as follows. $A_0 = \emptyset$. For $s > 0$,
$A_s = A_{s-1} \cup \{x : \exists e$

$W_{e,s} \cap A_{s-1} = \emptyset$	We haven't met e-th simplicity requirement.
$x \in W_{e,s}$	We can meet it via x.
$x \geq 2e$	This makes A co-infinite.
$c(x, s) \leq 2^{-e}\}$.	This ensures that A obeys c.

□

The K-Mart analogy

- We want to buy a shirt of each color e at K-Mart, provided that there is a sufficient number of shipments from China.
- For the shirt of color e we can spend at most 2^{-e}.
- Eventually, a sufficiently cheap shirt of color e will arrive, unless that color is discontinued.
- We can buy all shirts that are not discontinued.
- We spend at most 2 dollars in total.

Cost function characterization of the K-trivials

The standard cost function $c_{\mathcal{K}}$ is given by

$$c_{\mathcal{K}}(x, s) = \sum_{x < w \leq s} 2^{-K_s(w)}.$$

We could also use $c(x, s) = \text{Prob}[\{\sigma \colon \mathbb{U}_s(\sigma) \geq x\}]$, the chance that the universal machine prints a string $\geq x$ within s steps.

Theorem (Nies 05)

A is K-trivial $\Leftrightarrow$
some computable approximation of A obeys $c_{\mathcal{K}}$.

Corollary

For each K-trivial A there is a c.e. K-trivial set $D \geq_T A$.

D is the change set $\{\langle x, i\rangle \colon A(x)$ changes at least i times$\}$.
One verifies that D obeys $c_{\mathcal{K}}$ as well.

Analogy with model theory

- We think of a cost function as a description of a class of Δ^0_2 sets: those sets with an approximation obeying the cost function.
- For instance, the standard cost function describes the K-trivial sets.
- This is somewhat similar to a sentence in some formal language describing a class of structures.
- "A obeys c" is like $A \models c$.
- The limit condition is consistency. We disregard computable sets.
- If c has a model it must satisfy the limit condition.
- The basic existence theorem shows that each "consistent" cost function has a (promptly simple) model.

Analogy with model theory

- We think of a cost function as a description of a class of Δ^0_2 sets: those sets with an approximation obeying the cost function.
- For instance, the standard cost function describes the K-trivial sets.
- This is somewhat similar to a sentence in some formal language describing a class of structures.
- "A obeys c" is like $A \models c$.
- The limit condition is consistency. We disregard computable sets.
- If c has a model it must satisfy the limit condition.
- The basic existence theorem shows that each "consistent" cost function has a (promptly simple) model.

Part 5 : Strong jump-traceability

- During 2002-2005 researchers thought of the K-trivials as the “strongest” lowness property on the Δ^0_2 sets.
- Recently a proper subclass has emerged (at least on the c.e. sets).
- It is defined in a purely computability-theoretic way, but can be characterized via randomness, using the “computed by many” paradigm.

Tracing

- The idea of tracing: the set A is weak as an oracle because for certain functions ψ computed relative to A, the possible values $\psi(x)$ lie in a finite set T_x of small size.
- The sets T_x are obtained effectively from x (not using A as an oracle).

Strongly jump traceable sets

- An order function is a function $h : \mathbb{N} \to \mathbb{N}$ that is computable, nondecreasing, and unbounded.
- A c.e. trace with bound h is a uniformly c.e. sequence $(T_x)_{x \in \mathbb{N}}$ such that $|T_x| \leq h(x)$ for each x.
- Let $J^A(e)$ be the value of the A-jump at e, namely, $J^A(e) \simeq \Phi_e^A(e)$.
- The set A is called strongly jump traceable if for each order function h, there is a c.e. trace $(T_x)_{x \in \mathbb{N}}$ with bound h such that, whenever $J^A(x)$ it is defined, we have

$$J^A(x) \in T_x$$

  (Figueira, Nies, Stephan, 2004).
- For jump-traceability, one merely requires that this works for some order function h.

A proper subclass of the c.e. K-trivial sets

Theorem (Figueira, Nies, Stephan 2004)

There is a c.e. incomputable strongly jump traceable set.

We also prove that A is strongly jump traceable $\Leftrightarrow$ A is "lowly" for the plain Kolmogorov complexity C, namely, for every order function h and almost every x, $C(x) \leq C^A(x) + h(C^A(x))$.
The hope was that strong jump traceability is a computability-theoretic characterization of K-triviality.
But, in fact:

Theorem (Cholak, Downey, Greenberg 2006)

The c.e. strongly jump traceable sets form a proper subideal of the K-trivial sets.

It is open whether this also holds within the Δ^0_2 sets.

Building a promptly simple strongly jump traceable set

We meet the prompt simplicity requirements

$$PS_e\colon \ \#W_e = \infty \ \Rightarrow \ \exists s\, \exists x\, [x \in W_{e,\mathrm{at}\ s} \ \&\ x \in A_s].$$

The function $\overline{K}(x) := \min\{K(y)\colon \ y \geq x\}$ is dominated by each order function g.

Construction of A. Let $A_0 = \emptyset$.

Stage $s > 0$. For each $e < s$, if PS_e is not satisfied and there is $x \geq 2e$ such that $x \in W_{e,\mathrm{at}\ s}$ and

$$\forall i\, \big[(e \geq \overline{K}_s(i)\ \&\ J^A(k)[s-1]\!\downarrow) \ \rightarrow \ x > \text{use } J^A(i)[s-1]\,\big]$$

then put x into A_s and declare PS_e satisfied.

Benign cost functions

The result of Cholak e.a. that SJT implies K-trivial for c.e. sets was reproved and extended using the language of cost functions.

Definition

We say that a cost function c is benign if

- $c(x+1,s) \leq c(x,s) \leq c(x,s+1)$ for each $x < s$ (monotonicity), and
- there is a computable function g such that $x_0 < x_1 < \ldots < x_k$ and $\forall i < k\,[c(x_i, x_{i+1}) \geq 2^{-n}]$ $\Rightarrow\ k \leq g(n)$.

Intuitively, for at most $g(n)$ times the cost of the current candidate x can grow to exceed 2^{-n}.
The standard cost function $c_{\mathcal{K}}$ is benign via $g(n) = 2^n$.

Characterizing SJT via cost functions

Theorem (Greenberg, Nies, ta)

Let A be c.e. Then
A is strongly jump traceable $\Leftrightarrow$
A obeys each benign cost function.

- In particular, A is K-trivial.
- We also prove that each benign cost function is obeyed by some c.e. set that is not strongly jump traceable.
- Hence we have another proof that SJT is a proper subclass of $\mathcal{K}$.

For "$\Leftarrow$" we have to define the right benign cost function to ensure tracing of J^A at order h.
The harder direction is "$\Rightarrow$". It uses the "box promotion method" of Cholak, Downey and Greenberg.

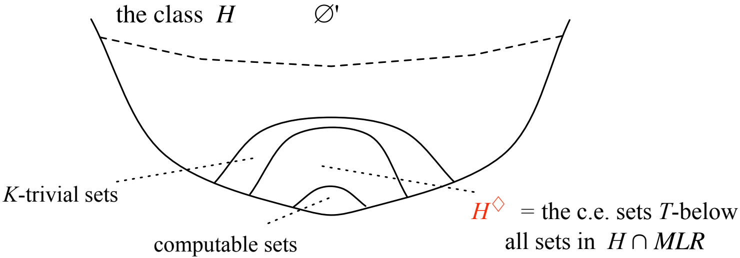
Part 6 : SJT sets are computed by many oracles

- We will give several characterizations of *SJT* as the c.e. sets that are easy to compute (in the sense that the class of oracles computing the set is large).
- For instance, A is strongly jump traceable $\Leftrightarrow$ A is Turing below each ω-c.e. ML-random set. (We say Y is ω-c.e. if $Y \leq_T \emptyset'$ with computably bounded use.)
- Thus, the computability-theoretic notion *SJT* can be characterized via randomness.
- For the K-trivials, the "easy to compute" property is "base for ML-randomness": $A \leq_T Y$ for some Y that is ML-random in A. In contrast, to characterize *SJT* we don't need to relativize ML-randomness.

Diamond Classes

For a null class $\mathcal{H} \subseteq 2^{\mathbb{N}}$, we define

$\mathcal{H}^{\Diamond}$ = the c.e. sets *A* Turing below each ML-random set in $\mathcal{H}$.



- The larger $\mathcal{H}$ is, the smaller is $\mathcal{H}^{\Diamond}$.
- $\mathcal{H}^{\Diamond}$ induces an ideal in the c.e. Turing degrees.
- If some ML-random set $Z \not\geq_T \emptyset'$ is in $\mathcal{H}$, then $\mathcal{H}^{\Diamond} \subseteq K$-trivial.

Existence Theorem

Theorem (Hirschfeldt/Miller)

For each null Σ^0_3 class $\mathcal{H}$, there is a promptly simple set in $\mathcal{H}^\Diamond$.

For instance, there is a promptly simple set in $(\omega\text{-c.e.})^\Diamond$.

- The theorem is proved by defining an appropriate cost function $c_\mathcal{H}$ with the limit condition.
- Whenever a c.e. set A obeys $c_\mathcal{H}$, then A is in $\mathcal{H}^\Diamond$.
- Now recall that some promptly set obeys A.

This implies that a ML-random set Y that is not weakly 2-random bounds an incomputable c.e. set: for $\mathcal{H}$ choose a null Π^0_2 class containing Y.

In the proof we implicitly build a Turing functional Γ. If $A = \Gamma^Z$ becomes wrong because A changes, we put Z into a Solovay test. So this Z cannot be random. The fact that A obeys c is used to show that it is indeed a Solovay test, i.e., we don't have to "correct" Γ on too many sets.

A lowness property and its dual highness property

- Recall that Z is low if $Z' \leq_T \emptyset'$, and Z is high if $\emptyset'' \leq_T Z'$.
- These classes are "too big": we have

$$(\text{low})^{\diamondsuit} = (\text{high})^{\diamondsuit} = \text{computable}.$$

  (For instance, $(\text{high})^{\diamondsuit}$= computable because there is a minimal pair of high ML-random sets.)
- So we will try somewhat smaller classes, replacing $\leq_T$ by the stronger truth-table reducibility $\leq_{tt}$.

Definition

A set Z is superlow if $Z' \leq_{tt} \emptyset'$. Z is superhigh if $\emptyset'' \leq_{tt} \emptyset'$.

A random set can be superlow (low basis theorem). It can also be superhigh but Turing incomplete (Kučera coding).

SJT is contained in the diamond classes

- Superlow is a countable Σ^0_3 class. Superhigh is contained in a null Σ^0_3 class (Simpson).
- So by the Hirschfeldt/Miller cost function we already know there is a promptly set in each of the corresponding diamond classes.
- Now we make such a cost function benign.

Theorem (Greenberg,Nies)

Let $\mathcal{H}$ be either superlowness or superhighness.

- *Then there is a benign cost function c such that each c.e. set obeying c is in $\mathcal{H}^\Diamond$.*
- *Thus SJT $\subseteq \mathcal{H}^\Diamond$.*

Conversely, the diamond classes are contained in SJT

- Greenberg, Hirschfeldt and Nies showed the converse inclusion, thereby giving two characterizations of the c.e. strongly jump traceable sets via randomness.
- We use a “golden run” construction with infinitely many levels.

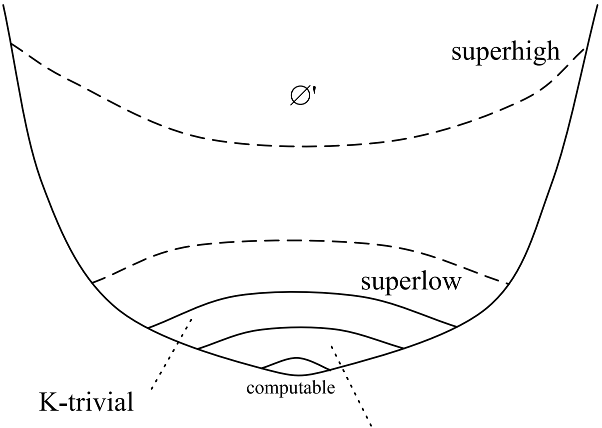
To summarize, we have:

Theorem

SJT $= (\omega - \text{c.e.})^{\Diamond}$ $=$*superlow*$^{\Diamond}$$=$ *superhigh*$^{\Diamond}$.

The proof that *SJT* $\supseteq$ superlow$^{\Diamond}$ is very general.

- We don’t need the hypothesis that the set is c.e.
- We can replace the ML-random sets by any non-empty Π^0_1 class.



SJT= $(\textit{superlow})^{\Diamond}$ $=(\textit{superhigh})^{\Diamond}$

Corollaries to the characterization of SJT

Often new characterizations give new views of the class. We obtain

- A new proof of the Cholak e.a. result that *SJT* induces an ideal in the c.e. Turing degrees (because every diamond class does that).
- a cost function construction (hence, injury-free) of a promptly simple set in *SJT* via the Hirschfeldt/MIller cost function $c_{\mathcal{H}}$ where $\mathcal{H} = (\omega)$-c.e. (say).

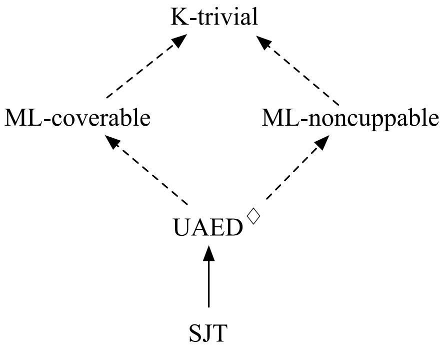
Open questions on classes between *SJT* and *K*-trivial

- No natural classes are currently known to lie properly between *SJT* and *K*-trivial
- A good candidate is $(\text{UAED})^\Diamond$. Here UAED is the class of uniformly almost everywhere dominating sets Z of Dobrinen and Simpson. (Equivalently, each random in Z is random in $\emptyset'$.) For the highness properties, there are proper implications

$$\text{Turing-complete} \Rightarrow \text{UAED} \Rightarrow \text{superhigh}.$$

- For the corresponding diamond classes, Greenberg and Nies proved that *SJT* is properly contained in $(\text{UAED})^\Diamond$.
- However, $(\text{UAED})^\Diamond$ may coincide with *K*-trivial.
- This would imply that the classes ML-coverable and ML-noncuppable also coincide with *K*-trivial.

Classes of c.e. sets between *SJT* and *K*-trivial



(The dashed arrows may be coincidences.)

- A is ML-coverable if $A \leq_T Y$ for some ML-random $Y \not\geq_T \emptyset'$.
- A is ML-noncuppable if
  $\emptyset' \leq_T A \oplus Y$ for ML-random Y implies $\emptyset' \leq_T Y$.