Strong Jump Traceability and Variations

Rod Downey
Victoria University
Wellington
New Zealand

REFERENCES

- Calibrating Randomness, (Downey, Hirschfeldt, Nies, Terwijn) Bulletin Symbolic Logic, 2006
- Five Lectures on Algorithmic Randomness (to appear Computational prospects of Infinity)
- The sixth lecture on algorithmic randomness, to appear proceedings Logic Colloquium 2006
- Algorithmic Randomness and Complexity, (Downey and Hirschfeldt) to appear Springer Verlag
- Randomness and Computability (Nies) to appear OUP

References

- Strong Jump-Traceability I : the Computably Enumerable Case, (Cholak, Downey, Greenberg) to appear Advances in Math.
- Strong jump-traceability II: the general case, (Downey and Greenberg) in prep
- Lowness properties and approximations of the jump. (Figueira, Nies, Stephan), to appear APAL
- Beyond strong jump traceability, (Ng Keng Meng) in prep
- On strongly jump traceable reals. (Ng Keng Meng) submitted
- On very high degrees, (Ng Keng Meng) to appear JSL.

Notation

- Real is a member of Cantor space 2^ω with topology with basic clopen sets $[\sigma] = \{\sigma\alpha : \alpha \in 2^\omega\}$ whose measure is $\mu([\sigma]) = 2^{-|\sigma|}$.
- Strings = members of $2^{<\omega} = \{0, 1\}^*$.
- There are theories for more general spaces, notably by Gács, (see his web site), but this is still under development. Certainly no lowness work.

Notation

- Real is a member of Cantor space 2^ω with topology with basic clopen sets $[\sigma] = \{\sigma\alpha : \alpha \in 2^\omega\}$ whose measure is $\mu([\sigma]) = 2^{-|\sigma|}$.
- Strings = members of $2^{<\omega} = \{0, 1\}^*$.
- There are theories for more general spaces, notably by Gács, (see his web site), but this is still under development. Certainly no lowness work.

Plain Kolmogorov Complexity

- Capture the incompressibility paradigm. Random means hard to describe, incompressible: e.g. 1010101010.... (10000 times) would have a short program.
- A string σ is random iff the only way to describe it is by hardwiring it. (Formalizing the Berry paradox)
- For a fixed machine N, we can define
- The Kolmogorov complexity $C(\sigma)$ of $\sigma \in \{0,1\}^*$ with respect to N, is $|\tau|$ for the shortest τ s.t. $N(\tau)\downarrow = \sigma$. (Kolmogorov)

Plain Kolmogorov Complexity

- Capture the incompressibility paradigm. Random means hard to describe, incompressible: e.g. 1010101010.... (10000 times) would have a short program.
- A string σ is random iff the only way to describe it is by hardwiring it. (Formalizing the Berry paradox)
- For a fixed machine N, we can define
- The Kolmogorov complexity $C(\sigma)$ of $\sigma \in \{0,1\}^*$ with respect to N, is $|\tau|$ for the shortest τ s.t. $N(\tau)\downarrow = \sigma$. (Kolmogorov)

- A string σ is N-random iff $C_N(\sigma) \geq |\sigma|$.
- A machine U is called weakly universal iff for all N, there is a d such that for all σ, $C_U(\sigma) \leq C_N(\sigma) + d$.
- Actually we will always use universal machines where the e-th machine is coded in a computable way.
- They exist (Kolmogorov). Hence there is a notion of Kolmogorov randomness for strings up to a constant. Define
$$U(1^e 0\sigma) = M_e(\sigma).$$
This particular coding gives $C(\tau) \leq M_e(\tau) + e + 1$.
- We will often write $=^+$, where we mean $\pm O(1)$.

- A string σ is N-random iff $C_N(\sigma) \geq |\sigma|$.
- A machine U is called weakly universal iff for all N, there is a d such that for all σ, $C_U(\sigma) \leq C_N(\sigma) + d$.
- Actually we will always use universal machines where the e-th machine is coded in a computable way.
- They exist (Kolmogorov). Hence there is a notion of Kolmogorov randomness for strings up to a constant. Define
$$U(1^e 0\sigma) = M_e(\sigma).$$
This particular coding gives $C(\tau) \leq M_e(\tau) + e + 1$.
- We will often write $=^+$, where we mean $\pm O(1)$.

- A string σ is N-random iff $C_N(\sigma) \geq |\sigma|$.
- A machine U is called weakly universal iff for all N, there is a d such that for all σ, $C_U(\sigma) \leq C_N(\sigma) + d$.
- Actually we will always use universal machines where the e-th machine is coded in a computable way.
- They exist (Kolmogorov). Hence there is a notion of Kolmogorov randomness for strings up to a constant. Define

$$U(1^e 0\sigma) = M_e(\sigma).$$

  This particular coding gives $C(\tau) \leq M_e(\tau) + e + 1$.
- We will often write $=^+$, where we mean $\pm O(1)$.

- A string σ is N-random iff $C_N(\sigma) \geq |\sigma|$.
- A machine U is called weakly universal iff for all N, there is a d such that for all σ, $C_U(\sigma) \leq C_N(\sigma) + d$.
- Actually we will always use universal machines where the e-th machine is coded in a computable way.
- They exist (Kolmogorov). Hence there is a notion of Kolmogorov randomness for strings up to a constant. Define

$$U(1^e 0\sigma) = M_e(\sigma).$$

  This particular coding gives $C(\tau) \leq M_e(\tau) + e + 1$.
- We will often write $=^+$, where we mean $\pm O(1)$.

- A string σ is N-random iff $C_N(\sigma) \geq |\sigma|$.
- A machine U is called weakly universal iff for all N, there is a d such that for all σ, $C_U(\sigma) \leq C_N(\sigma) + d$.
- Actually we will always use universal machines where the e-th machine is coded in a computable way.
- They exist (Kolmogorov). Hence there is a notion of Kolmogorov randomness for strings up to a constant. Define

  $$U(1^e 0\sigma) = M_e(\sigma).$$

  This particular coding gives $C(\tau) \leq M_e(\tau) + e + 1$.
- We will often write $=^+$, where we mean $\pm O(1)$.

Definition

Thus we can define the plain Kolmogorov complexity of a string σ as $C(\sigma)$ for a fixed universal machine U.

- We can similarly do an oracle version of this and can define $C(x|y)$ as the Kolmogorov complexity of x given y. (And $C^A(x)$ for a set A)

Plain Counting Theorem

- The following is the basic fact that makes the theory work.

Theorem (Plain Counting Theorem-Kolmogorov)

$|\{\tau : C(\tau) \leq |\tau| - d\}| \leq O(1)2^{|\tau|-d}$.

- Proof: pigeonhole principle.

Definition (Kolmogorov)

We say that σ is *C-random* iff $C(\sigma) \geq |\sigma|$.

Complexity Oscillations

- Tempting but false $C(xy) \leq C(x) + C(y) + O(1)$. The false argument says : concatenate the machines
- The problem is where does x^* stop and y^* begin.
- Martin-Löf showed that the formula always fails for long enough srings and hence reals.

Complexity Oscillations

- Tempting but false $C(xy) \leq C(x) + C(y) + O(1)$. The false argument says : concatenate the machines
- The problem is where does x^* stop and y^* begin.
- Martin-Löf showed that the formula always fails for long enough srings and hence reals.

Complexity Oscillations

- Tempting but false $C(xy) \leq C(x) + C(y) + O(1)$. The false argument says : concatenate the machines
- The problem is where does x^* stop and y^* begin.
- Martin-Löf showed that the formula always fails for long enough srings and hence reals.

- Why? Take any α. Then, as a string $\alpha \upharpoonright n$ corresponds to some number which we can interpret as a string using llex ordering: $\alpha \upharpoonright n$ is the m-th string.
- Now consider the program that does the following. It takes a strings ν, interprets its length $m_\nu = |\nu|$ as a string, $\sigma = \sigma_m$ and outputs $\sigma\nu$.
- Apply this to the string τ whose length is m th code of $\alpha \upharpoonright n$.
- The output would be much longer, and would be $\alpha \upharpoonright m + n$, with input having length m. Thus $C(\alpha \upharpoonright m + n) < m + n - O(1)$.

- Why? Take any α. Then, as a string $\alpha \upharpoonright n$ corresponds to some number which we can interpret as a string using llex ordering: $\alpha \upharpoonright n$ is the m-th string.
- Now consider the program that does the following. It takes a strings ν, interprets its length $m_\nu = |\nu|$ as a string, $\sigma = \sigma_m$ and outputs $\sigma\nu$.
- Apply this to the string τ whose length is m th code of $\alpha \upharpoonright n$.
- The output would be much longer, and would be $\alpha \upharpoonright m + n$, with input having length m. Thus $C(\alpha \upharpoonright m + n) < m + n - O(1)$.

- Why? Take any α. Then, as a string $\alpha \upharpoonright n$ corresponds to some number which we can interpret as a string using llex ordering: $\alpha \upharpoonright n$ is the m-th string.
- Now consider the program that does the following. It takes a strings ν, interprets its length $m_\nu = |\nu|$ as a string, $\sigma = \sigma_m$ and outputs $\sigma\nu$.
- Apply this to the string τ whose length is m th code of $\alpha \upharpoonright n$.
- The output would be much longer, and would be $\alpha \upharpoonright m + n$, with input having length m. Thus $C(\alpha \upharpoonright m + n) < m + n - O(1)$.

- The reason this kills the $C(xy) \leq^{+} C(x) + C(y)$ is to apply this to a sufficiently long random $z = xy$ where
- $C(z) = p$, and $x = z \upharpoonright m + n$ (as above) and $y = z \upharpoonright [m + n + 1, |z|]$.
- Then $p > n + (p - (m + n)) - O(1) = p - m + O(1)$.

- This phenomenom is fundamental in our understanding of Kolmogorov complexity and is called complexity oscillations.
- There are several known ways to get round this problem to cause only to get the information provided by the bits of the strings.
- Telephone numbers!

UNIVERSAL COMPUTERS

- Levin, Gaćs, Chaitin, Schnorr.
- Telephone numbers!!!!
- A computer M is prefix-free if

$$(M(\sigma)\downarrow \ \wedge \ \sigma' \supsetneq \sigma) \Rightarrow M(\sigma')\uparrow .$$

- A prefix-free machine is universal if every other one is coded in it.
- They exist, same proof.
- Now we have the *bits* of σ producing τ.

PREFIX-FREE RANDOMESS

- Prefix freeness gets rid of the use of length as extra information: Machines concatenate!
- The prefix-free complexity $K(\sigma)$ of $\sigma \in \{0,1\}^*$ is $|\tau|$ for the shortest τ s.t. $M(\tau)\downarrow = \sigma$.
- Note now $K(\sigma) \leq |\sigma| + K(|\sigma|) + d$, about $n + 2\log n$, for $\sigma| = n$.
- Build M, $M(z\sigma) = \sigma$ if $U(z) = |\sigma|$.

K-Counting Theorem

Theorem (Counting Theorem-Chaitin)

$|\{\sigma : |\sigma| = n \wedge K(\sigma) \leq n + K(n) - c\}| \leq O(1)2^{n+K(n)-c}$.

- As with life, relationships here are complex (Solovay)

$$K(x) = C(x) + C^{(2)}(x) + \mathcal{O}(C^{(3)}(x)).$$

and

$$C(x) = K(x) - K^{(2)}(x) + \mathcal{O}(K^{(3)}(x)).$$

- These 3's are sharp (Solovay) That is, for example, $K = C + C^2 + C^3 + O(C^4)$ is NOT true.

- As with life, relationships here are complex (Solovay)

$$K(x) = C(x) + C^{(2)}(x) + \mathcal{O}(C^{(3)}(x)).$$

and

$$C(x) = K(x) - K^{(2)}(x) + \mathcal{O}(K^{(3)}(x)).$$

- These 3's are sharp (Solovay) That is, for example, $K = C + C^2 + C^3 + O(C^4)$ is NOT true.

LOWNESS

- I would like to discuss the remarkable story of lowness.
- I will try to explain the little boxes method, which is new and poorly understood.
- Theme: to what extent do computational lowness (the extent to which sets resemble computable ones) and being far from random align themselves?

LOWNESS

- I would like to discuss the remarkable story of lowness.
- I will try to explain the little boxes method, which is new and poorly understood.
- Theme: to what extent do computational lowness (the extent to which sets resemble computable ones) and being far from random align themselves?

Lowness

- I would like to discuss the remarkable story of lowness.
- I will try to explain the little boxes method, which is new and poorly understood.
- Theme: to what extent do computational lowness (the extent to which sets resemble computable ones) and being far from random align themselves?

KEY FACTS

- THEOREM (CHAITIN)

  There is a constant d such that for all c and all σ,

$$|\{\nu : U(\nu) = \sigma \wedge |\nu| \leq C(\sigma) + c\}| \leq d2^c.$$

THEOREM (LEVIN, CHAITIN)

There is a constant d such that for all c and all σ,

$$|\{\nu : U(\nu) = \sigma \wedge |\nu| \leq K(\sigma) + c\}| \leq d2^c.$$

- The point here is that d is independent of $|\nu|$ and depends onl y on the Recursion Theorem, and c

Key facts

- **Theorem (Chaitin)**

  There is a constant d such that for all c and all σ,

  $$|\{\nu : U(\nu) = \sigma \wedge |\nu| \leq C(\sigma) + c\}| \leq d2^c.$$

  Theorem (Levin, Chaitin)

  There is a constant d such that for all c and all σ,

  $$|\{\nu : U(\nu) = \sigma \wedge |\nu| \leq K(\sigma) + c\}| \leq d2^c.$$

  - The point here is that d is independent of $|\nu|$ and depends onl y on the Recursion Theorem, and c

Information characterization of computability

- Chaitin proved that a real A is computable iff for all n, $C(A \upharpoonright n) \leq^{+} \log n$, iff $C(A \upharpoonright n) \leq^{+} C(n)$.
- This is proven using the fact that a Π^0_1 class with a finite number of paths has computable paths, combined with the Counting Theorem $\{\sigma : C(\sigma) \leq C(n) + d \wedge |\sigma| = n\} \leq A2^d$. (The Loveland Technique)
- Loveland had earlier shown A is computable iff $C(A \upharpoonright n|n) \leq c$ for some c and all n.

- If $C(\alpha \upharpoonright n|n) \leq 5$ then there are only 5 programmes possibly computing initial segments of α.
- This computes a tree of *strings* of maximal width 5.
- Therefore only at most 5 paths. Say 4.
- Imagine the situation that there is only *one* path in a tree of maximal width 2.

K-TRIVIALITY

- What is $K(A \upharpoonright n) \leq^+ K(n)$ for all n? We call such reals K-trivial. Does A K-trivial imply A computable?
- Write $A \in KT(d)$ iff for all n, $K(A \upharpoonright n) \leq K(n) + d$.

K-TRIVIALITY

- What is $K(A \upharpoonright n) \leq^{+} K(n)$ for all n? We call such reals K-trivial. Does A K-trivial imply A computable?
- Write $A \in KT(d)$ iff for all n, $K(A \upharpoonright n) \leq K(n) + d$.

The argument fails

- It is still true that $\{\sigma : K(\sigma) \leq K(|\sigma|) + d\}$ is $O(2^d)$, so it would appear that we could run the Π^0_1 class argument used for C. But no...
- The problem is that we don't know $K(n)$ in any computable interval, therefore the tree of K-trivials we would construct would be a Π^0_1 class relative to $\emptyset'$.

Theorem (Chaitin, Zambella)

There are only $O(2^d)$ members of $KT(d)$. They are all Δ^0_2.

Theorem (Solovay)

There are noncomputable K-trivial reals.

Theorem (Zambella)

Such reals can be c.e. sets.

A remarkable class

- K-trivials form a remarkable class as we will see.
- First they solve Post's problem.
- Theorem: (DHNS) If A is K-trivial then $A <_T \emptyset'$.
- See "The Sixth lecture" for details of the proof.

Similar methods allow for us to show the following

THEOREM (NIES)

All K-trivials are superlow $A' \equiv_{tt} \emptyset'$, and are tt-bounded by c.e. K-trivials. In fact they are Jump Traceable as we see below.

Thus triviality is essentially an "enumerable" phenomenom.

There are other antirandomness notions.

Definition (Kučera and Terwijn)

We say A is low for randomness iff the reals Martin-Löf random relative to A are exactly the Martin-Löf random reals.

Definition (Hirschfeldt, Nies, Stephan)

A is a a base of a cone of randomness iff $A \leq_T B$ with B A-random.

THEOREM

The following are equivalent to being K-trivial.

(I) *(Nies) A is low for randomness.*

(II) *(Hirschfeldt and Nies) A is K-low in that $K^A =^+ K$.*

(III) *(Hirschfeldt, Nies, Stephan) A is a base of a cone of randomness.*

(IV) *(Downey, Nies, Weber, Yu+Nies, Miller) A is low for weak-2-randomness.*

Questions and a proper subclass

It is open if this is the same as a number of other “cost function” classes such as the reals which are Martin-Löf cuppable to $\emptyset'$. (Nies)
It is known there is a proper subclass defined by cost function.

Definition (Nies)

Let h be an order. We say that A is jump traceable for the order h iff there is a computable collection of c.e. sets $W_{g(e)}$ with $|W_{g(e)}| < h(e)$ and $J^A(e) \in W_{g(e)}$. A is strongly jump traceable iff it is jump traceable for every computable order.

THEOREM (NIES)

A is K-triv implies that there is an order h (roughly $n \log n$) relative to which A is jump traceable.

THEOREM (FIGUEIRA, NIES, STEPHAN)

Noncomputable sjt c.e. sets exist.

THEOREM (CHOLAK, DOWNEY, GREENBERG)

The c.e. sjt's are a proper subclass of the K-trivials. They form an ideal.

THEOREM (DOWNEY, GREENBERG)

If A is sjt then A is Δ^0_2

- Roughly need orders $\sqrt{\log n}, \log\log n$. Is there a combinatorial characterization?

Theorem (Nies)

A is K-triv implies that there is an order h (roughly $n \log n$) relative to which A is jump traceable.

Theorem (Figueira, Nies, Stephan)

Noncomputable sjt c.e. sets exist.

Theorem (Cholak, Downey, Greenberg)

The c.e. sjt's are a proper subclass of the K-trivials. They form an ideal.

Theorem (Downey, Greenberg)

If A is sjt then A is Δ^0_2

- Roughly need orders $\sqrt{\log n}$, $\log \log n$. Is there a combinatorial characterization?

THEOREM (NIES)

A is K-triv implies that there is an order h (roughly $n \log n$) relative to which A is jump traceable.

THEOREM (FIGUEIRA, NIES, STEPHAN)

Noncomputable sjt c.e. sets exist.

THEOREM (CHOLAK, DOWNEY, GREENBERG)

The c.e. sjt's are a proper subclass of the K-trivials. They form an ideal.

THEOREM (DOWNEY, GREENBERG)

If A is sjt then A is Δ_2^0

- Roughly need orders $\sqrt{\log n}, \log\log n$. Is there a combinatorial characterization?

- Conjecture : A is K-trivial iff A is jump traceable for all computable orders h with $\sum_{n\geq 1}\frac{1}{h(n)} < \infty$.

A hint of the proof techniques

- To show that if A and B are c.e. sjt, so is $A \oplus B$.
- We show h we can construct a slower order k such that if A and B are jump traceable via k then $A \oplus B$ is jump traceable via h
- Opponent gives: $W_{p(x)}$ jump tracing A and $W_{q(x)}$ jump tracing B, such that $|W_{p(x)}|, |W_{p(x)}| < k(x)$.
- We: V_z tracing $J^{A \oplus B}(z)$ with $|V_z| < h(z)$.

Two Obstacles

- We see an apparent jump computation $J^{A\oplus B}(x)\downarrow[s]$.
- Should we believe? We only have $h(x)$ many slots in the trace V_x to put possible values.
- Opponent can change A or B after stage s on the use.
- We build parts of jump (recursion thm) testing A and B
- Basic idea: For some $a = a(x)$ and $b = b(x)$ we will define

$$J^B[s](b) = j_B(x, s) \text{ and } J^A[s](a) = j_A(x, s),$$

where $j_C(x, s)$ denotes the C-use of the $J^{A\oplus B}(x)[s]$ computation.

- Ignore *noncompletion*: that is the $A \oplus B$ computation changes before these procedures return.
- Simplest case: $W_{p(a)}$ and $W_{q(b)}$ were of size 1 (1-boxes)
- Then if return: $A \oplus B$ is correct
- Now 2-boxes. If the $A \oplus B$ computation is wrong, at least one of the A or B ones are too.

- If we are lucky and there is are false jump computations in both of the $W_{p(a)}$ and $W_{q(b)}$.
- The are now, in effect 1-boxes. (Very good)
- Can't allow to only point at one side. Use up all the 2-boxes.
- For example if always the A sides was the wrong part, and there were k 2-boxes then after k attacks, all the 2-boxes would be useless and the information in the B-side is correct, hence the box is used.

MULTIPLE BOXES

- Idea: use multiple 2-boxes. E.g. at the beginning use two 2-boxes for the same computation.
- A side was wrong. Then now we have *two promoted* 1-boxes.
- Since the A-computation now must be correct, if the believed computation is wrong, it must be the B side which wrong the next time, now creating a new B-1-box. Finally the third time we test, we would have two 1-boxes.

Non-return

- Now we face the ignored problem. We test and before the computation retruns, the Jump computation is changed by an A or B change, but possibly one of the A or B uses is correct. Now nothing is promoted. This seems very bad.
- Even with 1-boxes.
- Use descending sequences of boxes, and big metaboxes.
- Complicated, combinatorial.

- The idea is that for a computation whose target is, say, 2-boxes, begin ever further out. Begin by testing at, say, s-boxes.
- Monster boxes called metaboxes.
- If both A and B return at the s-box, go to $s-1$ etc. Only believe if you get back to the 2-boxes. The idea that a failure at k promotes $k+1,\ldots,s$-boxes, at least on one side.
- A combinatorial argument if used to show that cannot favour one side forever.

- How to make a properly ω-c.e. K-trivial?
- Use descending costs....
- If the trace grows slowly enough then can make K-trivial and not jt at that order. Much the same idea, the key point being the to change the trace and use a a box location, the use is very big, and the opponent needs more tailweight.

THE GENERAL CASE

- We conjecture that all sjt's are bounded by c.e. sjts.
- They are all Δ_2^0 (Downey and Greenberg). This is a very difficult result.
- We can prove an apparently smaller class are all good in this sense, strongly well-approximable. (being written)

Ng Keng Meng's Theorems

- The c.e. sjt's are Π^0_4 complete.
- This solves a problem of Nies: there is no minimal order.

Beyond jump traceability

- Say A is C-sjt iff for all orders h^B, for $B \in C$, A is h^B-jt.
- (Ng) No real is C-sjt where $C = \Delta_2$.
- (Ng) There are c.e. reals sjt for all c.e. sets.
- (Ng) They cannot be promptly simple, the first such class.
- (Ng) No real is K^B-trivial for all B, or c.e. B.

Why can't the be prompt?

- The construction is really a $\mathbf{0}'''$ argument since we need to guess whether φ_e^W is really a trace. Like the proof that the sjt's are Π_4^0 complete this need the full apparatus of $\mathbf{0}'''$ method.
- (Ng) There are sjt A and B such that B is not A-sjt.

Lots of ignored work

- Use pseudo-jump inversion to talk about “ultrahighness”
- E.g. a proper subclass of the “almost everywhere dominating”
- cappables etc. (Ng)