Yet more on strongly jump-traceable reals

Noam Greenberg

2nd June 2009

Definitions

A TRACE for a partial function $p\colon \omega \to \omega$ is a uniformly c.e. sequence of finite sets $\langle S_x \rangle$ such that for all $x \in \operatorname{dom} p$, $p(x) \in S_x$.

An ORDER is a non-decreasing and unbounded recursive function.

A trace $\langle S_x \rangle$ is BOUNDED by an order h if for all x, $|S_x| \leqslant h(x)$.

Strong jump-traceability

Definition (Figueira, Nies, Stephan)

A Turing degree $\mathbf{a}$ is *strongly jump-traceable* if for every order function h, every $\mathbf{a}$-partial computable function has a trace which is bounded by h.

Theorem (Figueira, Nies, Stephan)

There is a promptly simple c.e. degree which is strongly jump-traceable.

c.e. SJTs – structure

Theorem (Cholak, Downey, G)

The c.e. strongly jump-traceable degrees form an ideal, which is strictly contained in the K-trivial degrees.

Robustness of SJT

Theorem (G, Hirschfeldt, Nies)

The following are equivalent for a c.e. set A.

1. *A is computable from every superlow random set.*
2. *A is computable from every superhigh random set.*
3. $\deg_T(A)$ *is strongly jump-traceable.*

What about non-c.e. SJTs?

Theorem

Every strongly jump-traceable degree is K-trivial.

THE PROOF

Let A be a set whose Turing degree is strongly jump-traceable.

FACT (ZAMBELLA)

If A is K-trivial, then A is a path on a Δ^0_2 tree which has finitely many paths:

$$\{\sigma \,:\, K(\sigma) \leqslant K(|\sigma|) + d\}\,.$$

We will find such a tree T.

A simplification

Without loss of generality, the function $n \mapsto K(n)$ is computable:

Theorem (Bienvenu, Downey)

There is a computable function g such that for all X, if for all n,

$$K(X \upharpoonright n) \leqslant^{+} g(n),$$

then X is K-trivial.

So the tree above is actually Σ^0_1.

The general plan

So we want to enumerate a tree T such that A is a path on T, and such that

$$\sum_{\sigma \in T} 2^{-K(|\sigma|)}$$

is finite. (We then use the KC theorem.)

The price for enumerating σ on T is

$$c(|\sigma|) = \sum_{m \leqslant |\sigma|} 2^{-K(m)}.$$

REQUIREMENTS

To make sure that the total price of T is finite, we consider infinitely many "requirements".

For $q \in \mathbb{Q}$, $q < c(\omega) = \lim_n c(n) = \Omega$, let n_q be the least n such that $c(n) \geq q$.

For $k \geq 1$, we let

$$T_k = \{\sigma \in T \ : \ |\sigma| = n_{2^{-k}}, n_{2 \cdot 2^{-k}}, n_{3 \cdot 2^{-k}}, \dots\} \, .$$

GOAL: enumerate T so that T_k has at most k leaves.

THE GOAL IS SUFFICIENT

If σ is a leaf of T_k, $|\sigma| = n_{m \cdot 2^{-k}}$, and $l = n_{(m-1)2^{-k}}$, we charge to σ the enumeration of all strings

$$\sigma \upharpoonright (l+1), \sigma \upharpoonright (l+2), \ldots, \sigma$$

into T. The cost is at most

$$m2^{-k} - (m-1)2^{-k} = 2^{-k}.$$

Hence the total charges are bounded by

$$\sum_k \sum_{\sigma \text{ a leaf of } T_k} 2^{-k} \leqslant \sum_k k 2^{-k}$$

which is finite.

The goal is sufficient

Every string on T is accounted for: let $\sigma \in T$, let $q = c(|\sigma|)$, and let k such that $q = m2^{-k}$ for some m.

Ask: is σ a leaf of T_k? If not, then either:

- $\sigma \in T_{k-1}$; or
- σ has an immediate successor τ on T_k which is in T_{k-1}.

In the second case, charge σ to τ, and then pass on the charge if necessary.

BUT HOW DO WE GET THE TREES T_k?

Tracing gives us a mechanism of TESTING strings, with a prescribed degree of certainty.

Formally, we define a functional Ψ and a slow-growing order function h. By the recursion theorem, we get a trace $\langle S_x \rangle$ for Ψ^A which is bounded by h (ignore overheads).

We say that x is a k-BOX if $h(x) \leqslant k$. Defining h allows us to specify, for each k, how many k-boxes we need for the argument. This number must be computable.

TESTING a string σ in a box x means defining $\Psi^\sigma(x)\downarrow = \sigma$. The test is successful if σ shows up in S_x. Note that when we issue an instruction to test σ on x, we need to make sure that we did not previously test on x any string comparable with σ.

First approximation to T_k: pre-approval

For every $q \in \{2^{-k}, 2 \cdot 2^{-k}, 3 \cdot 2^{-k}, \ldots\}$, we test all strings of length n_q on a k box. This gives us, for every such q, at most k possibilities for $A \upharpoonright n_q$.

THE MAIN STEP

Let B_1 the set of (at most k) strings which are pre-approved for the first level of T_k.

To enumerate strings into the second level of T_k, we need to guess which strings on the first level of T_k are leaves of T_k. Thus for every subset D of B_1 we test the strings in D on a dedicated k-box.

One can think of these boxes as forming a k-dimensional vector space over $\mathbb{F}_2$. Each string in B_1 corresponds to a hyperplane of this space.

The main step, repeated

Now for every string σ in B_2, we repeat this process, localised to every box whose guess about the first-level leaves is consistent with σ being on T_k. Note that this keeps the testing requirements consistent.

Hence we need, for every $D \subseteq B_1$, to split the box dedicated for D (to test all subsets of B_2 consistent with D). Hence instead of a single box for each D, we have a conglomeration of boxes (a "meta box").

What next?

Question

Let A be strongly jump-traceable. Is A computable from a c.e., strongly jump-traceable set?

If so, we get some nice results:

- The strongly jump-traceable degrees form an ideal.
- Strong superlowness and strong jump-traceability coincide.