A promptly simple set which is not superlow cuppable

David Diamondstone

5/19/08

Background

Lowness is a ubiquitous concept in computability. Two of the most important ones are:

- low: $A' \leq_T \emptyset'$
- low for random/K-trivial: $\mathrm{MLR}^A = \mathrm{MLR}$

In this talk we'll focus on a slight strengthening of the first condition:

- superlow: $A' \leq_{tt} \emptyset'$

Superlow sets are connected to various other lowness notions concepts in randomness. Here are a few facts:

- all K-trivials are superlow (Nies, 2005)
- for c.e. sets, being superlow is equivalent to being jump traceable
- there is a Martin-Löf random set which is superlow (this follows from the superlow basis theorem)
- there are c.e. superlow sets A_0 and A_1 so that $A_0 \oplus A_1$ is complete

In the last property on the preceding slide, we saw that there are two superlow c.e. sets which cup, in the sense that their join is complete.

- Question: Which sets low cup?

In the last property on the preceding slide, we saw that there are two superlow c.e. sets which cup, in the sense that their join is complete.

- Question: Which sets low cup?
- Answer (Ambos-Spies, Jockusch, Shore, Soare, 1984): A c.e. set B can be cupped by a low c.e. set iff B is of promptly simple degree.

Definition

A c.e. set B is promptly simple if there is an enumeration (B_s) of B and a computable function $p : \omega \to \omega$ so that

$$(\forall e)[W_e \text{ infinite } \Rightarrow (\exists x)(\exists s)[x \in W_{e,\text{ at} s} \cap B_{p(s)}]]$$

Superlow cupping

André Nies asked the same question about superlow sets:

- Question (Nies): Can every promptly simple set be cupped to $\emptyset'$ by some superlow c.e. set?

Superlow cupping

André Nies asked the same question about superlow sets:

- Question (Nies): Can every promptly simple set be cupped to $\emptyset'$ by some superlow c.e. set?
- Answer (D.): No. There is a promptly simple set which is not superlow cuppable.

Think about superlow sets using the following fact:

- $X \leq_{tt} \emptyset'$ iff X is ω-c.e.

In particular, A is superlow iff A' is ω-c.e. This means we have a Δ^0_2 approximation to A', and a computable bound on the number of times the approximation changes its mind.

Prompt sets are low-cuppable

Theorem

(Ambos-Spies, Jockusch, Shore, Soare, 1984) If B is promptly simple, then there is a low c.e. set A for which $A \oplus B$ is complete.

Proofs are games

Let's examine the proof of this theorem.

- Imagine the proof as a two-player game:
- Red wants to build a promptly simple set B which is not superlow cuppable.
- Blue wants to build a low set A which cups with Red's B.
- Blue also builds Θ to (try to) compute $\emptyset'$ from $A \cup B$.

Imagine the use of $\Theta^{A \cup B}(i)$ as a movable marker $\theta(i)$.

- Blue can move $\theta(i)$ by changing A, and has to do so when i enters $\emptyset'$. Blue can also move $\theta(i)$ if some smaller element enters B.

Imagine the use of $\Phi_e^A(e)$ as a movable marker $\phi(e)$.

- We think of Red as controlling these markers to his own advantage. Blue can pick up a marker by changing A, but then Red decides when and where to put it down again.

Blue needs to compute $\emptyset'$ while ensuring that each marker $\phi(e)$ is picked up only finitely often.

There is some interaction between requirements, but it can be ignored:

- Each requirement R_e is trying to ensure $\phi(e)$ is picked up finitely often.
- Any finite amount of injury is acceptable.
- Each marker only produces finite injury.

Can A be made superlow?

To make A superlow, Blue needs to computably bound the number of changes made by some approximation to A'.

- If Blue makes the obvious approximation to A', Red can easily defeat him.

Can A be made superlow?

To make A superlow, Blue needs to computably bound the number of changes made by some approximation to A'.

- If Blue makes the obvious approximation to A', Red can easily defeat him.
- What if Blue make some other approximation?

Can A be made superlow?

To make A superlow, Blue needs to computably bound the number of changes made by some approximation to A'.

- If Blue makes the obvious approximation to A', Red can easily defeat him.
- What if Blue make some other approximation?
- In fact, there is a strategy for Red that defeats all attempts to make A superlow.

Theorem

There is a promptly simple set B which is not superlow cuppable.

Proof. We will construct such a set B.

Theorem

There is a promptly simple set B which is not superlow cuppable.

Proof. We will construct such a set B.
I will explain how to defeat a single A which claims to be superlow via some approximation g to A', and cup with B via some functional Θ.

Our strategy

In building B, we have a difficulty:

- We want to take some e and force it in and out of the jump A' until the approximation g changes its mind more than it said it would.
- Imagine we control all of the functionals Φ_e, as well as $\emptyset'$.
- We can stick e into the jump by defining $\Phi_e^\sigma(e)\downarrow$ on some apparent initial segment of A
- We can pull e back out of the jump by changing $\emptyset'$ and waiting for A to change in response

Our strategy

In building B, we have a difficulty:

- We want to take some e and force it in and out of the jump A' until the approximation g changes its mind more than it said it would.
- Imagine we control all of the functionals Φ_e, as well as $\emptyset'$.
- We can stick e into the jump by defining $\Phi_e^\sigma(e)\downarrow$ on some apparent initial segment of A if the approximation to A' changes in response.
- We can pull e back out of the jump by changing $\emptyset'$ and waiting for A to change in response

Our strategy

In building B, we have a difficulty:

- We want to take some e and force it in and out of the jump A' until the approximation g changes its mind more than it said it would.
- Imagine we control all of the functionals Φ_e, as well as $\emptyset'$.
- We can stick e into the jump by defining $\Phi_e^\sigma(e) \downarrow$ on some apparent initial segment of A if the approximation to A' changes in response.
- We can pull e back out of the jump by changing $\emptyset'$ and waiting for A to change in response if there is some i for which $\theta(i) < \phi(e)$.

Why this doesn't work:

- We have to wait for g's approximation to A' to change, and restrain B while we are waiting.
- Instead of the approximation changing, A might change.
- When it happens again, we impose more and more restraint. B is not prompt.

Three key ideas:

- First idea: don't restrain B while we wait.
- Second idea: test the waters with a marker $\phi(10)$ while we keep $\phi(0)$ in reserve.
- Third idea: do impose restraint on B when some test marker is down, but with low priority.

First idea:

- We want a test marker, *any marker*, $\phi(e)$ bigger than some attackers $\theta(0), \ldots, \theta(k)$. We can just place markers one after another until g agrees that the corresponding e is in A'.
- At some point we place a marker, say $\phi(10)$, and don't see an A change removing it. (If this never happens, one of the use markers $\theta(i)$ moves infinitely often.)
- At some later stage the approximation g agrees that $10 \in A'$. (Otherwise g is permanently mistaken.)

Second idea:

- When g agreed that our test marker $\phi(10)$ was down, it indicated our test was successful. We now place $\phi(0) = \phi(10)$.

Second idea:

- When g agreed that our test marker $\phi(10)$ was down, it indicated our test was successful. We now place $\phi(0) = \phi(10)$.
- What we really want is for g to agree that 0 is in A'. Since $\phi(0) > \theta(0), \ldots, \theta(k)$, we can force g to take it out again by enumerating one of $0, \ldots k$ into $\emptyset'$, restraining B, and waiting for A to change.

Second idea:

- When g agreed that our test marker $\phi(10)$ was down, it indicated our test was successful. We now place $\phi(0) = \phi(10)$.
- What we really want is for g to agree that 0 is in A'. Since $\phi(0) > \theta(0), \ldots, \theta(k)$, we can force g to take it out again by enumerating one of $0, \ldots k$ into $\emptyset'$, restraining B, and waiting for A to change.
- If g doesn't agree, then A has to change below $\phi(0) = \phi(10)$. This gives us two mind-changes by g on 10. If we can repeat this process, we can make g change its mind too often on 10 instead.

Second idea:

- When g agreed that our test marker $\phi(10)$ was down, it indicated our test was successful. We now place $\phi(0) = \phi(10)$.
- What we really want is for g to agree that 0 is in A'. Since $\phi(0) > \theta(0), \ldots, \theta(k)$, we can force g to take it out again by enumerating one of $0, \ldots k$ into $\emptyset'$, restraining B, and waiting for A to change.
- If g doesn't agree, then A has to change below $\phi(0) = \phi(10)$. This gives us two mind-changes by g on 10. If we can repeat this process, we can make g change its mind too often on 10 instead.
- We need a new test marker, say $\phi(20)$. Every time we see agreement, we work back up the chain of test markers. Every time A changes, we add a new link at the end.

Third idea:

- We obtain a long chain of test markers.
- Every time we place a test marker, we don't want B to change below $\max\{\theta(0), \dots, \theta(k)\}$.
- So we restrain B, with priority equal to the position of our test marker in the chain.
- When restraint is violated, we throw out only as much of the chain as we have to, and start building a new branch. Our chain becomes a tree.

There are two possibilities:

1. We keep returning our attention to the same height.

2. We keep building downwards, returning attention to each height finitely often.

There are two possibilities:

1. We keep returning our attention to the same height.
   - Each time we return to that height, we put some e into A', get agreement, and take it back out.
   - Eventually g changes its mind too many times.
   - At this stage we have defeated (A, g, Θ), and don't need to do many work.
2. We keep building downwards, returning attention to each height finitely often.

There are two possibilities:

1. We keep returning our attention to the same height.
   - Each time we return to that height, we put some e into A', get agreement, and take it back out.
   - Eventually g changes its mind too many times.
   - At this stage we have defeated (A, g, Θ), and don't need to do many work.
2. We keep building downwards, returning attention to each height finitely often.
   - In this case, we impose restraint of each priority at most finitely often.
   - A keeps changing below one of $\theta(0), \ldots, \theta(k)$, so Θ is not total. Again, we have defeated (A, g, Θ).

There are two possibilities:

1. We keep returning our attention to the same height.
   - Each time we return to that height, we put some e into A', get agreement, and take it back out.
   - Eventually g changes its mind too many times.
   - At this stage we have defeated (A, g, Θ), and don't need to do many work.
2. We keep building downwards, returning attention to each height finitely often.
   - In this case, we impose restraint of each priority at most finitely often.
   - A keeps changing below one of $\theta(0), \ldots, \theta(k)$, so Θ is not total. Again, we have defeated (A, g, Θ).

□

Thank you

- K. Ambos-Spies, C. G. Jockusch, Jr., R. A. Shore and R. I. Soare, An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees, Trans. Amer. Math. Soc. *281* (1984), 109–128.
- R. Downey, N. Greenberg, J. Miller, and R. Weber, Prompt simplicity, array computability and cupping, Computational Prospects of Infinity, forthcoming.
- A. Nies, Lowness properties and randomness, Adv. Math. *197* (2005), no. 1, 274–305.
- A. Nies, *Computability and Randomness*, Clarendon Press, Oxford, to appear.
- R. I. Soare, *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets*, Springer-Verlag, Heidelberg, 1987.

R. I. Soare, *Computability Theory and Applications*, Springer-Verlag, Heidelberg, to appear.