Lowness in algorithmic randomness

Noam Greenberg

Victoria University of Wellington

13th January 2012

Lowness

Definition

A set (an oracle) A is called low if $A' \equiv_T \emptyset'$.

That is, applying the Turing jump operator erases the difference between A and the computable sets. The general question is:

- How do we extend computability “just a little bit” beyond the computable sets (but not going near the halting problem)? What oracles are close to useless?

Low basis

A basis theorem says that problems in a certain class always have simple solutions. The following is prominent (and useful):

Theorem (Josckusch,Soare)

Every nonempty effectively closed subset of Cantor space 2^ω *contains a low element.*

Equivalently, every infinite, computable, binary branching tree contains a low path. This is useful because some such trees do not contain computable paths.

So for example:

- There is a low completion of Peano Arithmetic.
- There is a low Martin-Löf random set.

Forcing

Basis theorems are usually proved by forcing arguments. In computability, this is just a fancy (but useful) way to say that an object is constructed by a sequence of approximations. Each step specifies an easily describable (and usually closed) subset of Cantor space 2^ω, and the final object will lie in their intersection.

For example, for the low basis theorem, we "force with Π^0_1-classes" (i.e., computable trees). We repeatedly trim the tree, ensuring that all paths have a desired property (deciding the jump). While each stage specifies a computable object (the tree), the final object will be incomputable.

What about randomness?

Let $\mathcal{C}$ be a notion of randomness (such as Martin-Löf's, Schnorr's, Demuth's,...). We can relativise using an oracle A to obtain $\mathcal{C}^A$, the class of A-random sets. An oracle may detect more regularities than computable strategies, and so $\mathcal{C}^A \subseteq \mathcal{C}$, and in general $\mathcal{C}^A \subsetneq \mathcal{C}$.

Definition

An oracle A is low for $\mathcal{C}$ if $\mathcal{C}^A = \mathcal{C}$.

Note a similarity with lowness: a set A is low if and only if $\Delta^0_2(A) = \Delta^0_2$.

A double goal:

- Understand $\mathcal{C}$ by understanding its possible relativisations;
- Understand weak oracles.

A good example: Schnorr

Theorem (Terwijn,Zambella;Kjos-Hanssen,Nies,Stephan)

An oracle is low for Schnorr randomness if and only if it is computably traceable.

Traceability is a uniform version of being computably dominated.

Definition

An oracle A is computably traceable if for all $f \leqslant_T A$ there is a computable sequence $\langle T_n \rangle$ of finite sets such that for all n,

- $|T_n| \leqslant n$; and
- $f(n) \in T_n$.

Note: for the bound on the size of T_n, we can take any order function.

How is this done?

One direction: use tracing to cover an A-Schnorr test by an unrelativised Schnorr test.

Other direction: use forcing.

1. Force with Schnorr closed sets of positive measure. (A closed set is a Schnorr set if its measure is computable.) This gives a Schnorr random set X.
2. Code functions f by a sequence of independent clopen sets $\langle B_{n,f(n)} \rangle$. Show that if (almost) all of these are (almost) disjoint from a fixed Schnorr closed set (of positive measure), then f has a computable trace. (Requires some calculations.)
3. Hence, if A is not computably traceable, witnessed by f, then we can force X to be contained in the A-Schnorr test $\langle B_{n,f(n)} \rangle$.

The same method can be used for Demuth randomness [Bienvenu,Downey,Greenberg,Nies,Turetsky]. Part (2) - obtaining the traces - required some probability theory.

An issue which comes up in this case: what is the correct relativisation of a randomness notion? Perhaps some of the ingredients should remain computable.

So we get a deeper understanding of the notion of randomness itself.

But this is not the same for all notions of randomness

For example,

Theorem (Nies)

An oracle is low for computable randomness if and only if it is computable.

In this way, randomness captures computability.

The curious case of ML randomness

Another deviation from the path is lowness for ML randomness.

Theorem (Nies)

The following are equivalent for an oracle A:

1. $K(A\restriction_n) \leqslant^+ K(n)$;
2. $K^A =^+ K$;
3. *A is low for ML randomness.*

There are only countably many such oracles. They are generated by c.e. sets. So: unlike lowness for Schnorr and Demuth, they cannot be created using forcing.

Capturing K-triviality?

Nonetheless, we seek a combinatorial characterisation of lowness for ML randomness – one that does not mention measure, randomness, or Kolmogorov complexity.

Definition

Let h be an order function (a computable growth rate). An oracle A is h-jump-traceable if every A-partial computable function ψ has a uniformly c.e. trace bounded by h.

Question: does h-jump-traceability for a class of order functions h capture lowness for ML randomness?

Theorem (Hölzl,Kräling,Merkle)

An oracle is low for ML randomness if and only if it is $O(g(n) - K(n))$-jump-traceable for every Solovay function g (think: time bounded complexity).

Question: can we get rid of O? Is every K-trivial set $\log n$-jump-traceable?

Random sets in the Turing degrees

Lowness and traceability play a role in understanding how the random degrees sit in the Turing degrees, especially with respect to c.e. degrees.

Theorem (Kučera;Gács)

Every set is computable from a ML random set.

Theorem (Kučera)

Every Δ^0_2 random set computes a noncomputable c.e. set.

ML covering

Theorem (Hirschfeldt,Nies,Stephan)

1. *Every c.e. set computable from an incomplete ML random set is low for ML randomness.*
2. *An oracle A is low for ML randomness if and only if there is some A-ML random set which computes A.*

The converse of (1) is an interesting open problem.

Random covering

We can vary the notion of randomness and ask analogues of the incomplete ML covering question.

Theorem (Nies,Kučera;Greenberg,Turetsky)

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

1. *A is computable from a Demuth random set.*
2. *A is h-jump-traceable for all order functions h.*

SJT

Those so-called strongly jump-traceable degrees form a very well-behaved proper subclass of the K-trivial (low for ML random) degrees. Like the K-trivials, they form an ideal, and are essentially enumerable. They also have characterisations as being computable from all ML random sets in particular classes, such a superlow and superhigh [Greenberg,Hirschfeldt,Nies]. They are also used to give solutions to problems in c.e. degree theory.

Many other variants

1. Lowness for effective Hausdorff dimension.
2. Lowness for Ω, and weak lowness for K (used by Miller to give a characterisation of 2-randomness using K).
3. Lowness for very weak randomness, and for notions of genericity.
4. Lowness for pairs of randomness notions.

Other issues: lowness vs. lowness for tests. Some unnerving recent results (Diamondstone and Franklin).

Weak reducibilities

Lowness can be interpreted as being the least degree of a weak reducibility [Nies]. Prominent is the weak reducibility corresponding to lowness for ML randomness, denoted by $\leqslant_{\mathrm{LR}}$.

- Gives a fascinating degree structure.
- Uses partial relativisation.

Thank you.