The interplay of randomness and genericity

Laurent Bienvenu (CNRS & Université de Bordeaux)
Christopher P. Porter (Drake University)

Computabilty Theory and Applications
Online seminar
November 10, 2020

Randomness and genericity in computability theory

(Algorithmic) randomness and genericity are central concepts of computability theory.

Randomness and genericity in computability theory

(Algorithmic) randomness and genericity are central concepts of computability theory.

A real (or infinite binary sequence) is “generic” if it is “typical” from the point of view of Baire category theory.

Randomness and genericity in computability theory

(Algorithmic) randomness and genericity are central concepts of computability theory.

A real (or infinite binary sequence) is “generic” if it is “typical” from the point of view of Baire category theory.

A real is “random” if it is “typical” from the point of view of measure theory.

A quick reminder

Definition

A real $X \in 2^\omega$ is **(Cohen) weakly n-generic** if X belongs to every dense $\emptyset^{(n-1)}$-effectively open set.

A quick reminder

Definition

A real $X \in 2^\omega$ is **(Cohen) weakly** ***n*****-generic** if X belongs to every dense $\emptyset^{(n-1)}$-effectively open set.

Definition

A real $X \in 2^\omega$ is **(Cohen)** ***n*****-generic** if for every $\emptyset^{(n-1)}$-effectively open set $\mathcal{U}$, either X belongs either to $\mathcal{U}$ or X belongs to the interior of $\mathcal{U}^c$ (equivalently, for every $\emptyset^{(n-1)}$-c.e. set of strings S, there is an n such that $X \upharpoonright n \in S$ or $X \upharpoonright n$ has no extension in S).

A quick reminder

Definition

A real $X \in 2^\omega$ is **(Cohen) weakly *n*-generic** if X belongs to every dense $\emptyset^{(n-1)}$-effectively open set.

Definition

A real $X \in 2^\omega$ is **(Cohen) *n*-generic** if for every $\emptyset^{(n-1)}$-effectively open set $\mathcal{U}$, either X belongs either to $\mathcal{U}$ or X belongs to the interior of $\mathcal{U}^c$ (equivalently, for every $\emptyset^{(n-1)}$-c.e. set of strings S, there is an n such that $X \upharpoonright n \in S$ or $X \upharpoonright n$ has no extension in S).

Strict hierarchy: weak-1-generic $\Leftarrow$ 1-generic $\Leftarrow$ weak-2-generic $\Leftarrow$ 2-generic $\ldots$

A quick reminder

Definition

For $n \geq 2$ a real $X \in 2^\omega$ is **weakly n-random** if for every sequence of uniformly $\emptyset^{(n-2)}$-effectively open sets $(\mathcal{U}_n)$ with $\mu(\mathcal{U}_n) \to 0$, we have $X \notin \bigcap_n \mathcal{U}_n$.

A quick reminder

Definition

For $n \geq 2$ a real $X \in 2^\omega$ is **weakly** ***n*-random** if for every sequence of uniformly $\emptyset^{(n-2)}$-effectively open sets $(\mathcal{U}_n)$ with $\mu(\mathcal{U}_n) \to 0$, we have $X \notin \bigcap_n \mathcal{U}_n$.

Definition

A real $X \in 2^\omega$ is ***n*-random** if for every sequence of uniformly $\emptyset^{(n-1)}$-effectively open sets $(\mathcal{U}_n)$ with $\mu(\mathcal{U}_n) \leq 2^{-n}$, $X \notin \bigcap_n \mathcal{U}_n$.

A quick reminder

Definition

For $n \geq 2$ a real $X \in 2^\omega$ is **weakly** ***n*-random** if for every sequence of uniformly $\emptyset^{(n-2)}$-effectively open sets $(\mathcal{U}_n)$ with $\mu(\mathcal{U}_n) \to 0$, we have $X \notin \bigcap_n \mathcal{U}_n$.

Definition

A real $X \in 2^\omega$ is ***n*-random** if for every sequence of uniformly $\emptyset^{(n-1)}$-effectively open sets $(\mathcal{U}_n)$ with $\mu(\mathcal{U}_n) \leq 2^{-n}$, $X \notin \bigcap_n \mathcal{U}_n$.

Strict hierarchy: 1-random $\Leftarrow$ weak-2-random $\Leftarrow$ 2-random ...

Randomness vs genericity

Random reals and generic real “look” very different. A random real looks... random (satisfies the law of large numbers in every base and in every subsequence), whereas a generic looks nothing like this (for example, the frequency of zeroes on initial segments oscillates between 0 and 1).

Randomness vs genericity

In fact, for sufficiently high levels of randomness and genericity, the two notions are completely orthogonal.

Theorem (Nies, Stephan, Terwijn)

If X is 2-random and Y is 2-generic, then (X, Y) form a minimal pair (for Turing reducibility).

Randomness vs genericity

However, this orthogonality no longer holds at lower levels of randomness. While generics are always bad at computing randoms (folklore result: no 1-generic can compute a 1-random), the opposite is not true.

Randomness vs genericity

However, this orthogonality no longer holds at lower levels of randomness. While generics are always bad at computing randoms (folklore result: no 1-generic can compute a 1-random), the opposite is not true.

- For any n-generic Y, there is a 1-random X such that $X \geq_T Y$ (Kučera-Gács).

Randomness vs genericity

However, this orthogonality no longer holds at lower levels of randomness. While generics are always bad at computing randoms (folklore result: no 1-generic can compute a 1-random), the opposite is not true.

- For any n-generic Y, there is a 1-random X such that $X \geq_T Y$ (Kučera-Gács).
- For any 2-random X, there exists a 1-generic Y such that $X \geq_T Y$ (Kautz).

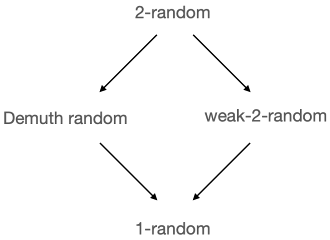
Between 1- and 2-

This raises the following question: can we get a more complete picture of the interplay between randomness and genericity when "randomness" is somewhere between 1-randomness and 2-randomness and/or genericity between 1-genericity and 2-genericity?

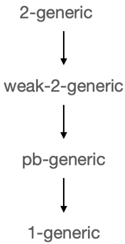
Between 1- and 2-

We will look at:

RANDOMNESS



GENERICITY



Demuth randomness

An ω-c.a. function $g : \mathbb{N} \to \mathbb{N}$ is a Δ^0_2 function with a computable approximation such that for each n, the number of mind changes for $g(n)$ is bounded by $h(n)$ for some computable bound h.

Definition

Let $(\mathcal{V}_e)$ be an enumeration of all c.e. open sets. A Demuth test is a sequence $(\mathcal{V}_{g(n)})$ where g is an ω-c.a. function and for all n, $\mu(\mathcal{V}_{g(n)}) \leq 2^{-n}$. A real $X \in 2^\omega$ is Demuth random if for every Demuth test $(\mathcal{V}_{g(n)})$, X only belongs to finitely many $\mathcal{V}_{g(n)}$'s.

A closer look at Kautz's result

Recall Kautz's theorem: every 2-random computes a 1-generic.
Originally, proof framed as a "measure-risking" strategy.

A closer look at Kautz's result

Recall Kautz's theorem: every 2-random computes a 1-generic.
Originally, proof framed as a "measure-risking" strategy.

However, it is more informative to frame it via a so-called fireworks argument (Shen).

A closer look at Kautz's result

Suppose we walk into a fireworks shop.

- The fireworks sold there are very cheap so we are suspicious that some of them are defective.

A closer look at Kautz's result

Suppose we walk into a fireworks shop.

- The fireworks sold there are very cheap so we are suspicious that some of them are defective.
- Since they are cheap we can ask the owner to test a few of them before buying one.

A closer look at Kautz's result

Suppose we walk into a fireworks shop.

- The fireworks sold there are very cheap so we are suspicious that some of them are defective.
- Since they are cheap we can ask the owner to test a few of them before buying one.
- **Our goal: either buy a good one (untested) and take it home OR get the owner to fail a test, and then sue him.**

A closer look at Kautz's result

Clearly there is no deterministic strategy which works in all cases. There is however, for any $\delta > 0$, a probabilistic strategy which wins with probability $> 1 - \delta$.

A closer look at Kautz's result

Clearly there is no deterministic strategy which works in all cases. There is however, for any $\delta > 0$, a probabilistic strategy which wins with probability $> 1 - \delta$.

- Fix n such that $1/n < \delta$.
- Pick a number k at random between 0 and n.
- Test the k first fireworks (stop if you get a bad one!).
- Buy the $(k+1)$-th box.

A closer look at Kautz's result

Clearly there is no deterministic strategy which works in all cases. There is however, for any $\delta > 0$, a probabilistic strategy which wins with probability $> 1 - \delta$.

- Fix n such that $1/n < \delta$.
- Pick a number k at random between 0 and n.
- Test the k first fireworks (stop if you get a bad one!).
- Buy the $(k + 1)$-th box.

This works because the only bad case is when $k + 1$ is the position of the first bad box.

Back to our construction of Y. Let (S_e) be an enumeration of all c.e. sets of strings. We want to satisfy for all e:

Back to our construction of Y. Let (S_e) be an enumeration of all c.e. sets of strings. We want to satisfy for all e:

$(\mathcal{R}_e)$: either for some n we have $Y \upharpoonright n$ is in S_e
or for some n, no extension of $Y \upharpoonright n$ is in S_e

Back to our construction of Y. Let (S_e) be an enumeration of all c.e. sets of strings. We want to satisfy for all e:

$(\mathcal{R}_e)$: either for some n we have $Y \upharpoonright n$ is in S_e
or for some n, no extension of $Y \upharpoonright n$ is in S_e

We build Y by finite extension, starting initially with the empty string.

The algorithm for a requirement *e* and global error probability δ:

The algorithm for a requirement e and global error probability δ:

Step 1 Pick a number k_e between 1 and some $q(e, \delta)$ at random, with $\sum_e 1/q(e, \delta) < \delta$. Set the 'error counter' to 0

The algorithm for a requirement e and global error probability δ:

Step 1 Pick a number k_e between 1 and some $q(e, \delta)$ at random, with $\sum_e 1/q(e, \delta) < \delta$. Set the 'error counter' to 0

Step 2
(a) Suppose we have already built some initial segment σ of X. Make the passive guess that there is no extension of σ in S_e
(b) Start handling other requirements. If we discover that our guess was wrong, increase error counter by 1 and go back to Step 2.a.
(c) If the error counter is $< k_e$, go back to the beginning of Step 2; if it is $= k_e$, go to Step 3.

The algorithm for a requirement e and global error probability δ:

Step 1 Pick a number k_e between 1 and some $q(e, \delta)$ at random, with $\sum_e 1/q(e, \delta) < \delta$. Set the 'error counter' to 0

Step 2
(a) Suppose we have already built some initial segment σ of X. Make the passive guess that there is no extension of σ in S_e
(b) Start handling other requirements. If we discover that our guess was wrong, increase error counter by 1 and go back to Step 2.a.
(c) If the error counter is $< k_e$, go back to the beginning of Step 2; if it is $= k_e$, go to Step 3.

Step 3 Stop everything else we were doing for other requirements. Let σ be the initial segment built so far; wait for some extension τ of σ to appear in S_e, and if so, let τ be our new initial segment of X and declare the requirement satisfied (otherwise, stay stuck in this loop forever).

Analysis of the algorithm

The algorithm works because of our discussion of the fireworks problem: the probability to get stuck at Step 3 for requirement $(\mathcal{R}_e)$ is $\leq 1/q(e, \delta)$.

Analysis of the algorithm

The algorithm works because of our discussion of the fireworks problem: the probability to get stuck at Step 3 for requirement $(\mathcal{R}_e)$ is $\leq 1/q(e, \delta)$.

Hence a global probability of failure bounded by $\sum_e 1/q(e, \delta) < \delta$.

Analysis of the algorithm

Suppose now that we are building Y by using the bits of an oracle $X \in 2^{\omega}$ as randomness generator. What does the failure set of our algorithm look like?

Analysis of the algorithm

Suppose now that we are building Y by using the bits of an oracle $X \in 2^\omega$ as randomness generator. What does the failure set of our algorithm look like?

Answer: for a given requirement $(\mathcal{R}_e)$, the set of X's that make the algorithm fail **because of** $(\mathcal{R}_e)$ form a difference of two effectively open sets.

Analysis of the algorithm

Suppose now that we are building Y by using the bits of an oracle $X \in 2^\omega$ as randomness generator. What does the failure set of our algorithm look like?

Answer: for a given requirement $(\mathcal{R}_e)$, the set of X's that make the algorithm fail **because of** $(\mathcal{R}_e)$ form a difference of two effectively open sets. Indeed, it is the difference of:

$\mathcal{U}_e^\delta$, the set of X's that make us enter Step 3 for $(\mathcal{R}_e)$,

Analysis of the algorithm

Suppose now that we are building Y by using the bits of an oracle $X \in 2^\omega$ as randomness generator. What does the failure set of our algorithm look like?

Answer: for a given requirement $(\mathcal{R}_e)$, the set of X's that make the algorithm fail **because of** $(\mathcal{R}_e)$ form a difference of two effectively open sets. Indeed, it is the difference of:

$\mathcal{U}_e^\delta$, the set of X's that make us enter Step 3 for $(\mathcal{R}_e)$,

minus $\mathcal{V}_e^\delta$, the set of X's that make us enter Step 3 for $(\mathcal{R}_e)$ and succeed at satisfying $(\mathcal{R}_e)$.

Strong difference randomness(?)

Now choose the bound function q such that for all $k = \langle e, n \rangle$, the failure set F_k of the algorithm for requirement $(\mathcal{R}_e)$ and error bound 2^{-n} has measure at most 2^{-k}.

Strong difference randomness(?)

Now choose the bound function q such that for all $k = \langle e, n \rangle$, the failure set F_k of the algorithm for requirement $(\mathcal{R}_e)$ and error bound 2^{-n} has measure at most 2^{-k}.

Now consider the test (F_k). If X passes the test (F_k) in the strong sense that X belongs to only finitely many F_k's, then this means that for some n, X is not in any of the the failure sets $F_{\langle e,n \rangle}$, i.e., the probabilistic algorithm with error bound 2^{-n} succeeds when using X as random source.

Strong difference randomness(?)

Now choose the bound function q such that for all $k = \langle e, n\rangle$, the failure set F_k of the algorithm for requirement $(\mathcal{R}_e)$ and error bound 2^{-n} has measure at most 2^{-k}.

Now consider the test (F_k). If X passes the test (F_k) in the strong sense that X belongs to only finitely many F_k's, then this means that for some n, X is not in any of the the failure sets $F_{\langle e,n\rangle}$, i.e., the probabilistic algorithm with error bound 2^{-n} succeeds when using X as random source.

Thus X computes a 1-generic via this algorithm (which is just a Turing reduction!).

Strong difference randomness(?)

The shape of the test X has to pass, a family (F_k) of differences of effectively open sets with $\mu(F_k) \leq 2^{-k}$ is exactly the same as the tests used to define **difference randomness** (Franklin and Ng), but the passing condition is harder (be in finitely many instead of not being in all F_k's).

Strong difference randomness(?)

The shape of the test X has to pass, a family (F_k) of differences of effectively open sets with $\mu(F_k) \leq 2^{-k}$ is exactly the same as the tests used to define **difference randomness** (Franklin and Ng), but the passing condition is harder (be in finitely many instead of not being in all F_k's).

In earlier presentation of this work, we defined strong difference randoms to be the set of X's such that for any family (F_k) of differences of effectively open sets with $\mu(F_k) \leq 2^{-k}$, X belongs to finitely many F_k's.

Strong difference randomness(?)

The shape of the test X has to pass, a family (F_k) of differences of effectively open sets with $\mu(F_k) \leq 2^{-k}$ is exactly the same as the tests used to define **difference randomness** (Franklin and Ng), but the passing condition is harder (be in finitely many instead of not being in all F_k's).

In earlier presentation of this work, we defined strong difference randoms to be the set of X's such that for any family (F_k) of differences of effectively open sets with $\mu(F_k) \leq 2^{-k}$, X belongs to finitely many F_k's.

What we missed (thanks to Hoyrup for pointing this out!) is that this is not a robust notion, i.e., it is not independent of the bound 2^{-n} (unlike Demuth randomness which is: we can replace 2^{-n} by $1/n^2$ or any computable sequence of bounds whose sum is a computable real).

Strong difference randomness(?)

Two options:

- Option 1: Quantify over all possible bounds, defining a strong difference test to be a sequence (F_k) of differences of effectively open sets with $\mu(F_k)$ uniformly computable in k and $\sum_k \mu(F_k)$ a computable real.
- Option 2: Keep the bound 2^{-n} but allow the F_k to be finite unions of differences of effectively open sets (this time the notion does not depend on the bound).

Strong difference randomness(?)

Two options:

- Option 1: Quantify over all possible bounds, defining a strong difference test to be a sequence (F_k) of differences of effectively open sets with $\mu(F_k)$ uniformly computable in k and $\sum_k \mu(F_k)$ a computable real.
- Option 2: Keep the bound 2^{-n} but allow the F_k to be finite unions of differences of effectively open sets (this time the notion does not depend on the bound).

The first option is what we should probably call strong difference randomness, but has not been studied in depth yet (there is recent work by McCarthy, but used the “old” definition).

Strong difference randomness(?)

An interesting turn of events:

Strong difference randomness(?)

An interesting turn of events:

Theorem

Option 2 is equivalent to Demuth randomness.

Strong difference randomness(?)

An interesting turn of events:

Theorem
Option 2 is equivalent to Demuth randomness.

and thus, as a corollary, we answer a question of Barmpalias, Day and Lewis-Pye:

Strong difference randomness(?)

An interesting turn of events:

Theorem
Option 2 is equivalent to Demuth randomness.

and thus, as a corollary, we answer a question of Barmpalias, Day and Lewis-Pye:

Theorem
Every Demuth random real computes a 1*-generic.*

Demuth randomness vs genericity

However, one cannot do better than 1-genericity in the previous theorem, at least for existing notions of genericity.

Theorem

If X is Demuth random and Y is pb-generic, then (X, Y) *form a minimal pair.*

Weak-2-randomness vs genericity

We now turn to weak-2-randomness. How does it interact with genericity?

Weak-2-randomness vs genericity

We now turn to weak-2-randomness. How does it interact with genericity?

In a nutshell: not all weak-2-random agree on the answer to this question!

Weak-2-randomness vs genericity

At one end of the spectrum, there are weak-2-randoms which are of hyperimmune-free degrees (folklore).

Weak-2-randomness vs genericity

At one end of the spectrum, there are weak-2-randoms which are of hyperimmune-free degrees (folklore).

... but a given X computes a weak-1-generic if and only if it has hyperimmune degree. So some weak-2-randoms cannot compute a single weak-1-generic.

Weak-2-randomness vs genericity

At the other end of the spectrum, it follows from earlier work that *some* weak-2-randoms can compute a 2-generic. The proof has two parts.

Weak-2-randomness vs genericity

At the other end of the spectrum, it follows from earlier work that *some* weak-2-randoms can compute a 2-generic. The proof has two parts.

Part 1. There is an interesting correspondance between the ability to compute generics and the ability to compute a function that is hard to bound. Let $\mathcal{F}$ be a family of functions from $\mathbb{N}$ to $\mathbb{N}$. We say that *X* has $\mathcal{F}$-escaping degree if *X* computes a function *g* which is not bounded by any $f \in \mathcal{F}$. For example, Δ^0_1-escaping = hyperimmune degree.

Weak-2-randomness vs genericity

The correspondance is as follows:

Theorem

- *X computes a weakly 1-generic iff X has* Δ^0_1*-escaping degree (Kurtz)*
- *X computes a pb-generic iff it has* $(\omega$*-c.a.)-escaping degree (Downey-Jockusch)*
- *X computes a weakly 2-generic iff it has* Δ^0_2*-escaping degree (Andrews-Gerdes-Miller)*
- *If X has* Δ^0_3*-escaping degree, it computes a 2-generic (Andrews-Gerdes-Miller)*

Weak-2-randomness vs genericity

Part 2. The second part is the following surprising theorem of Barmpalias, Downey and Ng.

Weak-2-randomness vs genericity

Part 2. The second part is the following surprising theorem of Barmpalias, Downey and Ng.

Theorem

For any countable family $\mathcal{F}$ of functions, there exists a weak-2-random X which has $\mathcal{F}$-escaping degree.

Weak-2-randomness vs genericity

Part 2. The second part is the following surprising theorem of Barmpalias, Downey and Ng.

Theorem

For any countable family $\mathcal{F}$ of functions, there exists a weak-2-random X which has $\mathcal{F}$-escaping degree.

Putting the two parts together:

Weak-2-randomness vs genericity

Part 2. The second part is the following surprising theorem of Barmpalias, Downey and Ng.

Theorem

For any countable family $\mathcal{F}$ of functions, there exists a weak-2-random X which has $\mathcal{F}$-escaping degree.

Putting the two parts together:

Theorem

There exists a weak-2-random X which computes a 2-generic.

Weak-2-randomness vs genericity

Can we do better than 2-generic? Perhaps, but not as a consequence of Barmpalias, Downey and Ng's theorem.

Weak-2-randomness vs genericity

Can we do better than 2-generic? Perhaps, but not as a consequence of Barmpalias, Downey and Ng's theorem.

Indeed, the correspondance between computing a generic and computing an escaping function abruptly ceases at the next level:

Weak-2-randomness vs genericity

Can we do better than 2-generic? Perhaps, but not as a consequence of Barmpalias, Downey and Ng's theorem.

Indeed, the correspondance between computing a generic and computing an escaping function abruptly ceases at the next level:

Theorem (Andrews, Gerdes, Miller)

There is no countable family $\mathcal{F}$ such that computing an $\mathcal{F}$-escaping function implies computing a weak-3-generic.

Weak-2-randomness vs genericity

However, one can strengthen Barmpalias, Downey and Ng's theorem and get:

Weak-2-randomness vs genericity

However, one can strengthen Barmpalias, Downey and Ng's theorem and get:

Theorem

For any comeager set $\mathcal{G}$, there is a weak-2-random which computes a member of $\mathcal{G}$ (in particular, for any n there is a weak-2-random which computes an n-generic).

A pretty complete picture

	n-gen. ($n \geq 2$)	weakly 2-gen.	pb-gen.	1-gen.
n-random ($n \geq 2$)	min. pair	min. pair	min. pair	computes
weakly 2-random	may compute	may compute	may compute	may compute
Demuth random	min. pair	min. pair	min. pair	computes
1-random	may compute	may compute	may compute	may compute

A pretty complete picture

	n-gen. ($n \geq 2$)	weakly 2-gen.	pb-gen.	1-gen.
n-random ($n \geq 2$)	min. pair	min. pair	min. pair	computes
weakly 2-random	may compute	may compute	may compute	may compute
Demuth random	min. pair	min. pair	min. pair	computes
1-random	may compute	may compute	may compute	may compute

A related open question:

If X is 1-random and of hyperimmune degree, does it compute a 1-generic?

Thank you