Classical results and new developments on logical depth of infinite binary sequences

Valentino Delle Rose

Università degli Studi di Siena

30.05.2022

$\emptyset'$ vs Ω

The same piece of information may be encoded in different ways, making it more or less useful for certain computational purposes: a paradigmatic example of this situation is given by the comparison between the halting set and Chaitin's Omega.

Let $(\varphi_e)_{e\in\mathbb{N}}$ be a fixed effective enumeration of all p.c. functions:

$$\emptyset' = \{\langle e, x\rangle : \ \varphi_e(x)\downarrow\},$$

is a c.e. set, hence its prefixes have *low* information content: $K(\emptyset' \upharpoonright n) \sim \log n$.

Let $\mathcal{U}$ be a fixed universal prefix-free machine:

$$\Omega = \sum_{\tau\in \mathrm{dom}\,\mathcal{U}} 2^{-|\tau|},$$

is a Martin-Löf random set, hence its prefixes have *maximal* information content: $K(\Omega \upharpoonright n) \sim n$.

$\emptyset'$ vs Ω

It is well-known that $\emptyset' \equiv_T \Omega$: in other words, they encode the same information.

More specifically, Ω can be seen as a *compressed version* of $\emptyset'$.

Fact.

For any string τ, one can decide whether $\mathcal{U}(\tau)\downarrow$ using $\Omega \upharpoonright |\tau|$ as oracle.

Proof. On input τ, look for the first step s such that $\Omega \upharpoonright |\tau| = \Omega_s \upharpoonright |\tau|$: then $\mathcal{U}(\tau)\downarrow$ if and only if it halts within s steps. Indeed, if $\mathcal{U}(\tau)\downarrow$ at some stage $s' > s$, we would have $\Omega_{s'} \geq \Omega_s + 2^{-|\tau|}$, contrary to our choice of s. □

In particular, one only needs the first n bits of Ω to compute the first $\sim 2^n$ bits of $\emptyset'$.

$\emptyset'$ vs Ω

However, this reduction is necessarily slow: indeed, **slower than any computable time-bound**.

In fact, to decide the first 2^n cases of the halting problem, we must enumerate enough halting programs until $\Omega_s \geq \Omega - 2^{-n}$: so, s must be at least as large as the running time of the slowest program of length n!

Actually, much more is true: recall that truth-table reductions are equivalent to Turing reductions running within some computable time bound.

Theorem 1 (Calude, Nies - 1997).

If $\emptyset' \leq_{tt} X$, then X is not Kurtz random (in particular, not ML-random).

In conclusion: Ω contains the same information as $\emptyset'$, but in such a compressed way to make it "computationally useless".

Logical depth: motivations

How can we distinguish between *useful* information and *useless* one?

The mere information content does not help in this task (see $\emptyset'$ vs Ω).

Bennett's approach: we have to look at how a given piece of information is internally organized. By using his own words:
"The value of a message is the amount of mathematical or other work plausibly done by its originator, which its receiver is saved from having to repeat". [1]

[1] C. Bennett. Logical Depth and Physical Complexity. *A half-century survey on The Universal Turing Machine*, pp. 227-257, New York, 1988.

Depth of finite strings

Bennett originally defined depth for finite strings: for any $c \in \mathbb{N}$ the *depth at significance level* c of a string σ is

$$\mathrm{depth}_c(\sigma) = \min_{s \in \mathbb{N}} (\exists \tau)[\mathcal{U}(\tau)[s] \downarrow = \sigma \text{ and } |\tau| - K(\tau) < c],$$

the minimum time required to compute σ from a string that cannot be compressed by c or more many bits.

In other words, at least c bits of "redundancy" are needed in order to compute σ in less than $\mathrm{depth}_c(\sigma)$ many steps.

Deep sets

Bennett considered a set to be *deep* if, no matter how much redundancy we allow, the depth of its prefixes dominates every computable time bound.

Definition 2.

$X \in \mathbf{2}^{\mathbb{N}}$ is *deep* if, for every computable time bound t and $c \in \mathbb{N}$,

$$\left(\overset{\infty}{\forall} n\right)\left[\mathrm{depth}_c(X \upharpoonright n) > t(n)\right].$$

Otherwise, X is said to be *shallow*.

A characterization of depth

Theorem 3 (Juedes, Lathrop and Lutz - 1994).

Let t be a computable time bound.

1. *If* $\mathrm{depth}_c(\sigma) > t(|\sigma|)$*, then* $K^t(\sigma) - K(\sigma) \geq c - O(1)$.
2. *There are a computable time bound* t' *and a constant* c'*, depending only on* t*, such that, if* $K^{t'}(\sigma) - K(\sigma) \geq c$*, then* $\mathrm{depth}_{c'}(\sigma) > t(|\sigma|)$.

Corollary.

X is deep if and only if, for every computable time bound t,

$$\lim_{n\to\infty} K^t(X \restriction n) - K(X \restriction n) = \infty.$$

Main features of depth

The comparison $\emptyset'$ vs Ω illustrates well the key properties of depth.

- $\emptyset'$ is the paradigmatic example of a deep set.
- Both sufficiently trivial (e.g. computable) and sufficiently random (e.g. ML-random) sets are shallow.
- **Slow Growth Lemma.** No deep set can be computed within any computable time bound from (i.e. is *tt*-reducible to) a shallow set.

$\emptyset'$ is deep

Theorem 4 (Bennett - 1988).

$\emptyset'$ is deep.

Proof. Given any computable time bound t, we construct a TM M: by the Recursion Theorem, we know an index e for M in advance.

Let $I_0, I_1, \dots$ and $J_0, J_1, \dots$ be partitions of $\mathbb{N}$ into consecutive intervals, with $\max I_k = 2^{k+1}$ (hence, $|I_k| = 2^k$ for $k > 0$) and $\max J_k = \langle e, \max I_k \rangle$.
NB. $x \mapsto \langle e, x \rangle$ maps elements in I_k to elements in J_k.

Claim. $n \in J_{k+1} \;\Rightarrow\; K^t(\emptyset' \upharpoonright n) \geq 2^k$.

If the claim is true, we are done: since $\emptyset'$ is c.e., for $n \in J_{k+1}$ we have

$$K(\emptyset' \upharpoonright n) \leq^+ \log n \leq^+ \log(\max J_{k+1}) \leq^+ \log(e + 2^{k+2})^2$$

and hence

$$K^t(\emptyset' \upharpoonright n) - K(\emptyset' \upharpoonright n) \geq^+ 2^k - \log(e + 2^{k+2})^2,$$

which is eventually larger than any constant, witnessing the depth of $\emptyset'$.

$\emptyset'$ is deep

Claim. $n \in J_{k+1} \Rightarrow K^t(\emptyset' \upharpoonright n) \geq 2^k$.

Proof of claim. Let

$$P_k = \{p : |p| < 2^k \text{ and } \mathcal{U}(p)[t(\max J_{k+1})] = \sigma \text{ with } |\sigma| \in J_{k+1}\}.$$

We must ensure: $\mathcal{U}(p) \neq \emptyset' \upharpoonright |\mathcal{U}(p)|$, whenever $p \in P_k$ and $|\mathcal{U}(p)| \in J_{k+1}$.

We use the 2^k inputs for M in I_k to diagonalize against the at most 2^{2^k} programs in P_k, as follows. On input x:

- compute k such that $x \in I_k$;
- compute the set of programs which havent't yet been "killed", i.e.

$$P_x = \{p \in P_k : (\forall y \in I_k)[y < x \Rightarrow \mathcal{U}(p)(\langle e, y\rangle) = \emptyset'(\langle e, y\rangle)]\};$$

- let $M(x)\downarrow \Leftrightarrow \mathcal{U}(p)(\langle e, x\rangle) = 0$ for at least half of the programs $p \in P_x$.

In this way, with each input x we "kill" half of the remaining programs in P_k, so we eventually diagonalize against the whole set P_k. □

Martin-Löf random sets are shallow

Recall that $X \in \mathbf{2}^{\mathbb{N}}$ is *ML-random* if and only if $K(X \upharpoonright n) \geq^{+} n$.

Theorem 5 (Bennett - 1988).

Every ML-random set is shallow.

Proof. Consider the copying function $\mathrm{id}(x) = x$: we can assume that $\mathcal{U}(\langle \mathrm{id}, x \rangle)$ runs in some polynomial time $t(|x|)$.

Let X be a ML-random set and let c be such that $K(X \upharpoonright n) \geq n - c$ for all n. In this case, we have

$$|\langle \mathrm{id}, X \upharpoonright n \rangle| - K(\langle \mathrm{id}, X \upharpoonright n \rangle) \leq^{+} n - K(X \upharpoonright n) < c,$$

meaning that $\mathrm{depth}_{c'}(X \upharpoonright n) \leq t(n)$ for some constant c'. □

The Slow Growth Lemma

Recall that $X \leq_{tt} Y$ if and only if there is a computable $\Phi : \mathbf{2}^{<\mathbb{N}} \to \mathbf{2}^{<\mathbb{N}}$ and a computable, non-decreasing and unbounded $f : \mathbb{N} \to \mathbb{N}$ such that $\Phi(Y \restriction n) = X \restriction f(n)$.

Theorem 6 (SGL: Juedes, Lathrop and Lutz - 1994).

If X is deep and $X \leq_{tt} Y$, then Y is deep.

Proof. Let $X \leq_{tt} Y$ via Φ and f.
Y is shallow $\Rightarrow$ there is a computable time bound t and a constant c with

$$\left(\overset{\infty}{\exists} n \, \exists \tau_n\right) \mathcal{U}(\tau_n)[t(n)] = Y \restriction n \text{ and } |\tau_n| - K(\tau_n) < c.$$

Φ is computable $\Rightarrow$ there is a computable time bound t' (depending only on t and Φ) such that, for all n, $\mathcal{U}(\langle \Phi, \mathcal{U}(\tau_n)\rangle)[t'(f(n))] = X \restriction f(n)$.
Then X is shallow as well, because

$$|\langle \Phi, \mathcal{U}(\tau_n)\rangle| - K(\langle \Phi, \mathcal{U}(\tau_n)\rangle) \leq^{+} |\tau_n| - K(\tau_n) < c,$$

hence, for infinitely many n, $\text{depth}_{c'}(X \restriction n) \leq t'(n)$, for some c'. □

Trivial sets

The fact that **computable sets are shallow** follows immediately by the previous two results.

Actually, a weaker "triviality condition" has been proven sufficient for a set to be shallow. Recall that a set X is *K-trivial* if $K(X \upharpoonright n) \leq^+ K(n)$. Clearly, every computable set is K-trivial, but there are also uncomputable ones (first proven by Solovay).

Theorem 7 (Moser, Stephan - 2017).

Every K-trivial set is shallow.

Depth vs highness

Every high degree contains an order-deep set

Several authors have discovered evidences of the interplay between depth and computational strength: an example concerns the degree of "highly" deep sets.

First, some terminology. Let us call an *order* any non-decreasing and unbounded function $h : \mathbb{N} \to \mathbb{N}$: then we say that a set X is *order-deep* if there is a computable order h such that, for every computable time bound t, $K^t(X \upharpoonright n) - K(X \upharpoonright n) \geq^+ h(n)$.

We can adapt the diagonalization technique used to prove that $\emptyset'$ is deep to show that **every high degree contains an order-deep set**.

Depth vs highness

Every high degree contains an order-deep set: construction

If A is high, the construction of an order-deep set $B \equiv_T A$ goes as follows.

Partition $\mathbb{N}$ into consecutive intervals $I_0, I_1, \dots$ with $I_k = [2^k, 2^{k+1})$ for $k > 0$. Choose $\delta < \frac{1}{2}$ and fix a computable function $h(k) < \delta k$: partition each I_k into $n_k = 2^{h(k)}$ many subintervals $I_k^0, \dots, I_k^{n_k - 1}$ of length $s_k = \frac{|I_k|}{n_k}$. Since A is high, there is a dominating function $T \leq_T A$: using T, we can compute, for each k, the set

$$\mathrm{TOT}_k = \{e \in [0, n_k - 2] : (\forall x \leq \max I_{k+1})\, \varphi_e(x)[T(\max I_{k+1})] \downarrow\}.$$

We define B by specifying its bits at each subinterval I_k^e.

1. The fist subinterval is used to code one bit of A: $B[I_k^0] = A(k)0^{s_k - 1}$.
2. For each $e \in [0, n_k - 2]$, do the following. If $e \notin \mathrm{TOT}_k$, let $B[I_k^{e+1}] = 0^{s_k}$. Otherwise, use the s_k bits in I_k^{e+1} to diagonalize the at most 2^{s_k} possible φ_e-fast codes for $B[I_{k+1}]$ of length $\leq s_k$, similarly to how we did to prove that $\emptyset'$ is deep.

Depth vs highness

Every high degree contains an order-deep set: verification

- Item 1 above ensures that $A \equiv_T B$.
- By item 2 above, for every computable time bound t and every $n \in I_{k+1}$, it holds that

$$K^t(B \upharpoonright n) > s_k = 2^{k-h(k)} \geq^+ 2^{\log n - h(\log n)}.$$

- Finally, we observe that, for every $e > 0$, the bits of $B[I_k^e]$ are uniformly computable, given the information whether $e \in \mathrm{TOT}_k$. Hence, to describe $B \upharpoonright n$ we only need the length n and, for each $k \leq \log n$ and $e \in [0, n_k - 2]$, one bit of information (i.e. whether $e \in \mathrm{TOT}_k$). Hence,

$$K(B \upharpoonright n) \leq^+ 2^{h(\log n)} \cdot \log n.$$

- Thus, by our choice of h and δ, we get

$$K^t(B \upharpoonright n) - K(B \upharpoonright n) > n^{1-\delta} - n^{\delta} \log n > n^{\epsilon},$$

for some small enough ϵ, meaning that B is order-deep. □

Depth vs highness

In fact, highness coincides with "large depth magnitudes" on the Turing degrees.

Theorem 8 (Moser and Stephan, 2017).

$A \in \mathbf{2}^{\mathbb{N}}$ is high if and only if there is $B \equiv_T A$ which is $(1-\epsilon)n$-deep, for every $\epsilon \in (0,1)$.

They also proved that any order-deep set must be either high or *DNC* (i.e. it computes a function D with $D(e) \neq \varphi_e(e)$ for every $e \in \mathbb{N}$). In particular, any c.e. order-deep set must be high (as every DNC c.e. set must be high). Morever, if A is c.e. and high, we can slighty modify the above construction to obtain a c.e. order-deep set $B \equiv_T A$: hence, **a c.e. degree is high if and only if it contains a c.e. order-deep set**.

On the other hand, highness is not necessary for smaller depth magnitudes.

Theorem 9 (Downey, McInerney and Ng - 2017).

There is a superlow c.e. deep set.

Relativized depth

The relativized notion of depth is meant to better understand the power of oracles in organizing information.

Definition 10.

Given an oracle A, we say that a set X is *A-deep* if, for every computable time-bound t,

$$\lim_{n\to\infty} K^{A,t}(X \upharpoonright n) - K^{A}(X \upharpoonright n) = \infty.$$

Otherwise, we say that X is *A-shallow*.

Notice that we stick to the class of computable time bounds, and not merely computable in the oracle A: indeed, we want a notion which is preserved upwards by the same class of "fast" reductions as Bennett's depth.

Main features of relativized depth

Theorem 11.

- *A' is A-deep.*
- **(Relativized SGL)** *If X is A-deep and $X \leq_{tt} Y$, then Y is A-deep.*
- *Every A-ML-random set is A-shallow.*
- *If $X \leq_{tt} A$, then X is A-shallow.*

An unusual scenario

Usually, the relativization of a class $\mathcal{C}$ defines a class $\mathcal{C}^A$ such that either $\mathcal{C}^A \subseteq \mathcal{C}$ (e.g., when $\mathcal{C}$ is the class of ML-random sets) or $\mathcal{C} \subseteq \mathcal{C}^A$ (e.g., when $\mathcal{C}$ is the class of computable sets), for all oracles A.

Being defined in terms of two quantities which decrease mutually independently when an oracle is applied, this is not the case of the class of deep sets. A priori, for an oracle A, we have four possible different cases:

1. DEEP and DEEP^A are incomparable;
2. $\text{DEEP} \subsetneq \text{DEEP}^A$;
3. $\text{DEEP} \supsetneq \text{DEEP}^A$;
4. $\text{DEEP} = \text{DEEP}^A$.

Jointly with Bienvenu and Merkle, we have shown that $A = \emptyset'$ is an example of the first case. Moreover, we observed that any K-trivial oracle fits one of the last two cases, but it is still open which one (we do not even know if it depends on the particular oracle!).

Depth relative to ML-random oracles

We also studied the case of ML-random oracles.

Theorem 12 (Bienvenu, Delle Rose, Merkle).

If R is ML-random, then, for every computable time-bound t, there is a computable time-bound t' with

$$(\forall\sigma)\left[K^{t'}(\sigma) - K(\sigma) \leq^{+} K^{R,t}(\sigma) - K^{R}(\sigma)\right].$$

Corollary.

If R is ML-random, then every deep set is also R-deep.

Moreover, we could prove that every shallow set remains shallow relative to almost every oracle.

Theorem 13 (Bienvenu, Delle Rose, Merkle).

If X is shallow, then $\mu(\{A : X \text{ is } A\text{-deep}\}) = 0$. In fact, X remains shallow relative to every X-2-random oracle.

Depth relative to ML-random oracles

Despite the previous result, it turns out that every ML-random R oracle makes some shallow set deep relative to it, meaning that $\mathrm{DEEP} \subsetneq \mathrm{DEEP}^R$.

Theorem 14 (Bienvenu, Delle Rose, Merkle).

For every ML-random set R, there is a shallow set which is R-deep.

Intuitively, the proof of this fact is similar to the one-time pad protocol in cryptography: we can "mix" together some important piece of information d with some random string r we know, so that the output $d \triangle r$ still looks important for us (as we can distinguish the added random noise r), while looking random to the others.

A shallow set which is R-deep

Lemma 15 (Moser and Stephan - 2017).

There is a non-empty Π^0_1 class containing only deep sets.

Hence, we can use well-known basis theorems to obtain deep sets with some desired properties.

Randomness Basis Theorem.

Let R be ML-random. Every non-empty Π^0_1 class contains a set X such that R is X-ML-random.

Let R be ML-random: combining the above results, there is a deep set D such that R is also D-ML-random. Moreover, D is R-deep, as ML-random oracles preserves depth. Consider

$$D \triangle R = (D \setminus R) \cup (R \setminus D).$$

For every pair of sets A, B, clearly $K^A(A \triangle B \upharpoonright n) =^+ K^A(B \upharpoonright n)$, hence:

- R is D-ML-random $\Rightarrow$ $D \triangle R$ is D-ML-random, hence shallow.
- D is R-deep $\Rightarrow$ $D \triangle R$ is R-deep. □

An application to PA-complete degrees

As a consequence of our previous result, we can give a short proof of the following theorem on PA-complete degrees.

Recall that a degree is PA-*complete* if and only if it computes a DNC_2 function, i.e. a DNC function with range $\{0, 1\}$.

Theorem 16 (Barmpalias, Lewis and Ng, 2010).

Every PA-*complete degree is the join of two ML-random degrees.*

An application to PA-complete degrees

A non-empty Π^0_1 class $\mathcal{C}$ is called *Medvedev-complete* if, for every Π^0_1 class $\mathcal{D}$, there is a total Turing functional (i.e. a *tt*-reduction) Φ such that $\Phi(\mathcal{C}) \subseteq \mathcal{D}$: a well-known example of such class is that of DNC_2 functions.

Notice that **all members of a non-empty Medvedev-complete Π^0_1 class must be deep**. Indeed, by Lemma 15, there is a non-empty Π^0_1 class $\mathcal{D}$ whose members are all deep. If $\mathcal{C}$ is Medvedev-complete, for every $X \in \mathcal{C}$ there is a deep set $D \in \mathcal{D}$ such that $D \leq_{tt} X$: then, by the Slow Growth Lemma, X must be deep as well.

An application to PA-complete degrees

The key point of our proof is the following lemma, whose proof uses techniques due to Kučera and Slaman (2006).

Lemma 17 (Bienvenu, Delle Rose, Merkle).

Let $\mathcal{C}$ be a non-empty Medvedev-complete Π^0_1 class. For every A of PA*-complete Turing degree, there exists $B \in \mathcal{C}$ such that $B \equiv_T A$.*

Since any DNC_2 function is deep, by Theorem 14 we get that the following Π^0_1 class is non-empty (for large enough d):

$$\mathcal{C} = \{\langle A, X, Y\rangle : \ A \in DNC_2, X \in MLR_d, Y \in MLR_d, X \triangle Y = A\},$$

where $MLR_d = \{X : (\forall n)[K(X \upharpoonright n) \geq n - d]\}$.
The first projection witnesses the Medvedev-completeness of $\mathcal{C}$, hence Lemma 17 applies to our class. Hence, for every B with PA-complete degree there is a triple $\langle A, X, Y\rangle \in \mathcal{C}$ such that $B \equiv_T \langle A, X, Y\rangle$.
Moreover, since $A = X \triangle Y$, clearly $B \equiv_T \langle A, X, Y\rangle \equiv_T \langle X, Y\rangle$. □