The Computational Power of Sets of Random Strings

Rod Downey
Victoria University
Wellington
Joint with Mingzhong Cai, Rachel Epstein, Steffen Lempp, and Joe Miller
Luminy, June 2016.

Sets of Random strings

- We work mainly with C (plain complexity) and prefix-free K.
- $R_H = \{x \mid H(x) > |x|\}$. The H-random strings.
- Related is the *overgraph*. $O_H = \{\langle x, n\rangle \mid H(x) > n\}$. Evidently $R_H \leq_m O_H$.
- Clearly, for $H \in \{C, K\}$, R_H is wtt-complete (strictly, $\overline{R_H}$).

- For the computability-theorist, this can be seen as a game.
- We pick a length $n(e)$ for e, and
- It is within our power to lower the complexity of strings of length $n(e)$ anh the opponent's to lower a proportion.
- The last stage that things stop becoming non-random at this length determines whether $\varphi_e(e)\downarrow$

What about stronger reducibilities?

- And does it depend on choice of universal machine?

Theorem (Kummer, 1996)

R^C_{tt} is always tt-complete.

- The proof was the first evidence of the complexity of the situation.
- It was *nonuniform*.
- It broke the potential random strings into blocks, enumerated them (as a fraction for each size) and argued that infinitely often they would be "true" and for these we would have a conjunctive *tt*-reduction. (i.e. $x \in \emptyset'$ iff all of a certain block are random.)

ALONG THESE LINES

- The following gives the idea and is easier.

THEOREM (AN. A. MUCHNIK)

The conditional overgraph $M = \{(x, y, n) : C(x|y) < n\}$ *is creative*

- It does not matter if K or anything else is used for C.

- The proof. We need $\emptyset' \leq_m M$.
- Parameter d known in advance.
- Construct possible g_x for $x \in [1, 2^d]$.
- Either we know $z \in \emptyset'$, or there is a unique y such that $g_x(z) = (x, y, d)$ and $x \in \emptyset'$ iff $g_x(z) \in M$.
- For some maximal x which enumerates elements infinitely often, g_x works.

- **Construction, stage** $s+1$ For each active $y \leq s$, find the least $q \in [1, 2^p]$ with

$$(q, y, d) \notin M_s.$$

(Notice that such an x needs to exist since $\{q : (q, y, d) \in M\} < 2^d$.)
If q is new, ie $(q', y, d) \in M_s$ for all $q' < q$, find the least z with $z \notin \emptyset'[s+1]$ and define

$$g_q(z) = (q, y, d).$$

- Now for any v, if v enters $\emptyset'[s+1]$, find the largest r, if any, with $g_r(v)$ defined. If one exists Find $\hat{y}$ with $g_r(v) = (r, \hat{y}, d)$. Declare that $\hat{y}$ is no longer active.

- Note that there must a largest $x \leq 2^d$ such that $\exists^\infty v(g_x(v) \in M)$. Call this x. We claim that g_x is the required m-reduction. Work in stages after which g_{x+1} enumerates nothing into M.
- Given z, since g_x is defined on infinitely many arguments and they are assigned in order,we can go to a stage s where either z has entered $\emptyset'[s]$, or $g_x(z)$ becomes defined, and $g_x(z) = (x, y, d)$ for some active y. $g_x(z)$ will be put into M should z enter $\emptyset'$ after s.

The *tt*-case

- The *tt*-case is tricky.
- For each $i < 2^d$ we will this time attempt an infinite sequence $S_{i,x}$ of strings and
- Enforce that for the largest i, $x \in \emptyset'$ iff $S_{i,x} \subseteq \overline{R_C}$.

Muchnik's Theorem

- The situation for R_{tt}^K and O_{tt}^K is more complex.

Theorem (An. A. Muchnik)
There exist universal prefix-free machines U_1 and U_2 where

1. $O_{tt}^{K_{U_1}}$ *is tt-complete.*
2. $O_{tt}^{K_{U_2}}$ *is not tt-complete.*

Theorem (Allender, Buhrman and Koucký)
There is a universal prefix-free V such that $R_{tt}^{K_V}$ is tt-complete.

- Muchnik (1) is kind of easy as we get to control coding locations, and can easily code $\emptyset'$ into universal machine "off to the side".
- Allender, Buhrman and Kouký is significantly more complex
- The proof involves first building a machine *V* which encodes "symmetrically" meaning that if σ has a description of length *n* so does $\overline{\sigma}$. (when τ, σ enters the "normal" *U* put $(0\tau, \sigma)$ and $1\tau, \overline{\sigma})$ into *V*.
- Then build a new universal machine *M* which "breaks the symmetry" at sparse coding locations to encode $\emptyset'$.

DAY'S THEOREMS

- Not covered in detail here is related work of Day.
- Recall M is a process machine if $\sigma \prec \tau$ and $M(\sigma)\downarrow, M(\tau)\downarrow$ implies $M(\sigma) \preceq M(\tau)$. This is *strict* if for all $\sigma' \prec \sigma$ if $M(\sigma)\downarrow$, then $M(\sigma')\downarrow$.
- Day observed that the Allender et. al. Theorem works also for monotone, strict process and process machines.

THEOREM (DAY)

1. *O_{Km} is always m-complete for optimal monotone machines.*
2. *O_{KM} is always tt-complete for optimal monotone machine.*
3. *For optimal (strict) process machines the overgraphs are tt-complete.*
4. *There is an optimal strict process machine where R_{Km_s} is not tt-complete.*

QUESTION (DAY)

Can any of the R_{H_U} not be tt-complete for universal U and $H \in \{Km, KM, KM_D\}$?

Proof of Muchnik's Theorem

- The proof was remarkable in the new ideas it brought to the area to use the determinacy of finite games.
- We want to build a universal prefix-free U to make O_K^U not *tt*-complete.
- We build part of the univeral U, via H and know which is max $K(\sigma) + 2, F(\sigma\}$ and built by KC requests.
- It is within our power to use F to lower the complexity of a string of length and within the opponent's power to lower H using K, but this costs him more.

- We build the rest of the machine and a c.e. set A and ensure that

$$\mathcal{R}_e : \Gamma^{K_U} \neq A.$$

- We pick a follower x, and wait till $\Gamma^{K_U}(x) \downarrow= 0[s]$.
- Now should we put x into A?
- We can view the situation like a directed graph. We can use F to lower complexity and the opponent can too.
- We have more entropy.
- There will be a winning strategy to force a value for $\Gamma^{K_U}(x)$.
- Of course in the real proof this can be construed as a game $G(\epsilon, \delta)$ where these quantities represent the measure each player has to work with.

The Allender et. al. Results

- Allender and his co-authors began a new program based on efficient reductions to theses sets.
- The intuition is that random elements should not help much except by luck, and this should be washed away by machine independence.

Theorem (Buhrman, Fortnow, Koucký and Loff; Allender, Buhrman, Koucký, van Melkebeek and Ronneburger 2006; Allender, Buhrman and Koucký 2006)

Let R be the set of all random strings for either plain or prefix-free complexity.

- $\mathrm{BPP} \subseteq \mathrm{P}^R_{tt}$.
- $\mathrm{PSPACE} \subseteq \mathrm{P}^R$.
- $\mathrm{NEXP} \subseteq \mathrm{NP}^R$.

- In some sense the levels are natural as strategies for games live in *PSPACE*.

Upper Boounds

Theorem (Allender, Friedman and Gasarch)

- $\Delta^0_1 \cap \bigcap_U \mathrm{P}_{tt}^{R_{K_U}} \subseteq \mathrm{PSPACE}$.
- $\Delta^0_1 \cap \bigcap_U \mathrm{NP}^{R_{K_U}} \subseteq \mathrm{EXPSPACE}$.

Here U ranges over universal prefix-free machines, K_U is prefix-free complexity as determined by U, and R_{K_U} is the corresponding set of random strings.

Conjecture (Allender, Friedman and Gasarch)

If $A \in \bigcap_U \mathrm{NP}^{R_{K_U}}$, then A is computable. (Therefore, $\Delta^0_1 \cap$ can be removed from both parts of the Theorem above.)

No minimal pairs

Theorem (CDELM)

For any prefix-free universal machines U_1 and U_2, there is a noncomputable c.e. set A such that $A \leq_{tt} R_{K_{U_1}}$ and $A \leq_{tt} R_{K_{U_2}}$.

Proof

- Let $K_j = K_{U_j}$ and $R_j = R_{K_{U_j}}$ for $j = 1, 2$.
- And $K(\sigma) = K^{U_1}(\sigma)$.
- Let g be a (computable) Solovay function, so that $g(n) \geq K(n)$ for all n, and $g(n) = K(n)$ infinitely often.
- We may assume for some b, $K(x) \leq g(x) \leq$b2 $\log x$ for all x.
- We construct $A \leq_{tt} R_1, R_2$, such that if A is computable, then there exists an infinite c.e. set with $g(n) = K(n)$.
- A contradiction (Solovay), in fact the "hitting set" of a Solovay function is both hyperimmune and Turing complete (Bienvenu, Downey, Merkle, Nies).

- We construct M_1, M_2 prefix-free machines with coding const d.
- The number of non-random strings of length n is $< 2^{n-g(n)-c}$ and we can divide the set of numbers below this into 2^{c+d} many regions of size $2^{n-g(n)-d}$.
- we know there is some maximal such region such that the size of the set of non-random strings lies in this region, and for infinitely many n with $g(n) = K(n)$.
- when we compress we will code information about which n have $K(n) < g(n)$.
- The above resembles the idea used by Kummer in his proof that R_C is *tt*-complete, where a maximum "block" gives the *tt*-information.
- Here we instead construct infinitely many candidates $A_{e,i}$. The *tt*-reduction is more or less the same as Kummer's in that for the correct (dynamically determined) block $\langle n, s\rangle \in A$ iff the block (like the $S_{e,i}$ in Kummer) has empty intersection with R_j.

Stronger Possibility

- Is it possible that there are no *wtt*-complete *tt*-minimal pairs.
- The easiest way would be to use minimal degrees. But...

Theorem (Downey and Shore, 1995)

- *If A has minimal tt-degree and is c.e. then A is low*$_2$.
- *Also the low*$_2$ *c.e. tt-degrees are exactly those with minimal covers.*

Question

- *Which c.e. degrees can contain minimal tt-degrees?*
- *which Turing degrees contain minimal pairs of tt-degrees?*

Question

Is are there wtt-complete c.e. A_1, A_2 *forming a tt-minimal pair? (in the c.e. tt-degrees also open.)*

A Partial Solution

Theorem (CDELM)

There exist Turing complete $A_1 \equiv_T A_2$ such that the tt-degrees form a minimal pair (in the tt-degrees).

Theorem (Downey and Ng, 2014)

There exist complete c.e. A_1, A_2 with A_1 wtt-complete, forming a tt-minimal pair in the tt-degrees.

- It might look like the second is a mild variation of the first, but the strategies are much more complex.
- Of course Selwyn and I were trying to solve the main question, and can do one mininal pair requirement with *wtt*-reductions to $\emptyset'$. The T-comes from combining requirements.
- I will sketch the easier proof of CDELM.
- We manke $A_i \geq_T \emptyset'$ via marker coding $\Gamma_i^{A_i} = Q$, Q complete, with use $\gamma_i(x, s)$.
- The position of $\gamma_i(x + 1)$ will determine $\emptyset'(x)$. Note the "$+1$".
- We use a dump construction for Q a complete set so that if x enters $Q_{s+1} - Q_s$ the so do x' for $x \leq x' \leq s$. This means only certain configurations are possible for the A_i.

- $R_e : \Delta^{A_1} = \Delta^{A_2} = f \to f$ computable.
- The main idea is we can use *tt*-reductions to examine the effect of coding.
- We look to see what happens when $\ell(e, s) > x$ for the first time.
- Let's suppose that $e = 0$ and this has highest priority, so can move all markers.
- We would begin with $x = 0$. When $\ell(e, s) > 0$ for the first time, what our plan is to move $\gamma_i(y, s)$ for $y > 0$ and $i = 1, 2$ and $\gamma_2(0, s)$ to fresh positions above $\delta(0)$.
- Now we would like to enumerate a definition of $\Delta^{A_i}(0)$, and notice that if we cannot, then we can force a disagreement.
- With argument 1 we will ensure that only $\gamma_1(0, s)$ is below the $\delta(1)$-use on the A_1-side and $\gamma_2(0, s)$ on the A_2-side.
- If 0 entering Q_s later can cause a disagreement, if we use $\gamma_i(0, s)$ to code this, then we can use $\gamma_1(1, s)$ on the A_1-side and $\gamma_2(0, s)$ to force a disagreement.

- Now consider arbitrary n.
- We'd like to define $f(n)$, but it could be that small numbers entering can cause a change in the current value even after we do the "kicking" manoeuvre.
- there will be a least such position and this can be exploited to make a disagreement.
- We won't define $f(n)$ but try for this. If later a Q-change causes this disagreement to go away, then since that was the least position then we can define $f(n)$ with with confidence.
- Note that the dump property means that no markers will be sent to infinity.

- The one with Ng requires more complex game analysis.
- Joe Miller has suggested that it might be possible three sets.
- Note you cannot do this with wtt-reducibility as

THEOREM (AMBOS-SPIES, 1985)

Computably enumerable A is wtt-cappable iff it is T-cappable.

THEOREM (CDELM)

For any universal U there is a noncomputable set $X \not\leq_{tt} R_K^U$.
Hence if $X \leq_{tt}$ all R_K^V's it is computable.

- This is the most complex proof in the paper. In fact the strategy is to construct three universal U_i and argue that not $X \leq_{tt} R_i$ for all i. We can assume that given X is Δ^0_2.
- $\mathcal{R}_i : \neg(\Psi_i^{R_j} = X \text{ for } j = 1, 2, 3)$.
- Force one of $\Psi^{R_j} \neq \Psi^{R_k}$ or $\Psi^{R_j} \neq X$ some j. (or prove that X is computable.)
- We make the machine universal as all will code V, via $U_j(000\sigma) = V(\sigma)$ and hence the opponent controls $\frac{1}{8}$ of the total measure.

- This time we modify Muchnik's other proof.
- Again the game works with come measure and $G(\epsilon, \delta)$ is the one where opponent (coder) ϵ to play and we have δ.
- We can force $\Psi^{R_j}(0)$ to be $i \in \{0, 1\}$.
- The possibilities are that for the game for a starting measure of ϵ_0 we can either force a disagreement for some i, j, force him to use too much measure (and then play again) or there is no such strategy and hence we are working towards X being computable.
- That is, we begin with $G(\epsilon_0, \epsilon_0)$ on the R_i's and see if we can force a disagreement; or the opponent uses too much measure.
- In this latter case, reset and start again. He only has $\frac{1}{8}$.

- If we can't with with $G(\epsilon_0, \epsilon_0)$, we call $G(\frac{\epsilon_0}{2}, \frac{\epsilon_0}{2})$.
- We compare the values with the original game.
- If we have $G(\epsilon_0, \epsilon_0)$ giving a R_k value and $G(\frac{\epsilon_0}{2}, \frac{\epsilon_0}{2})$ a different value R_ji, and the opponent plays honestly, we win.
- Continue with a stack of games, with ever smaller measure, and eventually it is the “real” 0-game. The value of the tt-functional on argument 0 is determined.
- There is a complex and technical modification when the opponent cheats at some level but not ϵ_0 we will start a modified game which is no longer symmetric, the asymmetry being determined by the amount the aoponent has spent.

"we can think of R_1 and R_2 as knights who have gone off to fight a battle. Their opponent has cheated and they return home. The bishop, R_3, is waiting for them and restores their faith when they return. If the three new games $G(\epsilon_0, \epsilon_0)$ all force the same value, it will be the same value as before. We will use this in the verification to show that if there is no disagreement between the three tt-reductions, then the set they are computing must be computable and so not X."

Details in the paper.

Multiple bits are even more complicated, as are interactions.

- Many Thanks