Drawbacks of plain Kolmogorov complexity

Intuitively, $C(\sigma\tau) \le C(\sigma) + C(\tau) + O(1)$ should hold

namely, the amount of information carried by $\sigma\tau$ should be no more than that of σ and τ

But

Theorem (1) For any $k \in \mathbb{N}$, we can find $\sigma \in 2^{<\omega}$ s.t. $C(\sigma) < |\sigma| - k$

Furthermore, Let $Z \in 2^\omega$ be any infinite 0,1-sequence, we can find such a $\sigma \prec Z$

(2) For any d, there is $\mu = \sigma\tau$ s.t. $C(\mu) > C(\sigma) + C(\tau) + d$. (Furthermore, if $Z \in 2^\omega$ has sufficiently long C-random initial segment, then there is such $\mu \prec Z$)

Proof (1) Consider the following machine M: with input ρ, it outputs

$$\sigma = \nu\rho \quad \text{where } \nu \text{ is the finite string coded by } |\rho|$$

Then $C(\sigma) = C_M(\sigma) + c_M = |\rho| + c_M = |\sigma| - |\nu| + c_M$ where c_M is the coding constant for M

Let $\nu = Z \upharpoonright k + c_M + 1$

note $|\nu| = k + c_M + 1$ and ν is coded by some $n \in [2^{k+c_M}, 2^{k+c_M+1})$
[namely, ν is the nth string in $2^{<\omega}$]

Let $\rho = Z \upharpoonright [k+c_M+1,\ k+c_M+n+1)$ and $\sigma = \nu\rho$.

So $|\rho| = n$ and $\sigma \prec Z$, $|\sigma| = k + c_M + 1 + n < k + c_M + 1 + 2^{k+c_M+1}$

$C(\sigma) = C_M(\sigma) + c_M = |\rho| + c_M = |\sigma| - |\nu| + c_M = |\sigma| - k - 1 < |\sigma| - k$

(2) Fix $d \in \mathbb{N}$. Let $k = d + c_{id}$ where c_{id} is the coding constant for identity

i.e. $C(\tau) \le |\tau| + c_{id}$ for and $\tau \in 2^{<\omega}$

Find σ s.t. $C(\sigma) < |\sigma| - k$.

Find an extension $\mu \succ \sigma$ s.t. $C(\mu) \ge |\mu|$ [it is possible by counting Lemma]
(If Z is sufficiently random, then there is such $\mu \prec Z$)

Let $\mu = \sigma\tau$

Then $C(\mu) \ge |\mu| = |\sigma| + |\tau| > C(\sigma) + k + |\tau| \ge C(\sigma) + k + C(\tau) - c_{id}$
$= C(\sigma) + C(\tau) + d$ □

This is possible, because we "cheated" when generate σ from ρ by using the information of $|\rho|$ while to generate μ, we don't know the information, since we don't know where the description of σ ends

Proposition $C(\sigma\tau) \le C(\sigma, \tau) + O(1) \le C(\sigma) + C(\tau) + 2\log C(\sigma) + O(1)$

Proof $C(\sigma\tau) \le C(\sigma,\tau) + O(1)$ is easy [consider the machine concatenate two input strings]

$C(\sigma,\tau) \le C(\sigma) + C(\tau) + O(1)$ may not hold
Let σ_C^*, τ_C^* be the shortest description of σ, τ respectively
Then $C(\sigma) + C(\tau) = |\sigma_C^* \tau_C^*|$, But we don't know where is the end of σ_C^* in $\sigma_C^* \tau_C^*$

We want to firstly code the length of σ_C^*. Note $|\sigma_C^*| = C(\sigma)$

Let ρ be coded by $C(\sigma)$, then $|\rho| = \log C(\sigma)$

Let $\bar{\rho} = \rho_0\rho_0 \cdots \rho_{|\rho|-1}\rho_{|\rho|-1}01$, then $|\bar{\rho}| = 2\log C(\sigma) + 1$

Consider the machine M: input μ, it try to find a initial segment of μ of the form $\bar{\rho}$, then compute the code n of ρ, pick $\mu \upharpoonright [|\bar{\rho}|, |\bar{\rho}|+n)$ (presume to be σ_C^*)
use the universal machine to compute $\sigma = U(\mu \upharpoonright [|\bar{\rho}|, |\bar{\rho}|+n))$
and use the rest of μ to compute τ

Therefore $C(\sigma,\tau) \le C_M(\sigma,\tau) + O(1) = |\bar{\rho}| + |\sigma_C^*| + |\tau_C^*| + O(1)$
$= C(\sigma) + C(\tau) + 2\log C(\sigma) + O(1)$ □

This technique is also used in the definition of

Conditional Complexity

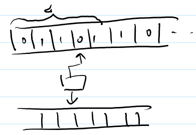
Def For strings σ and τ, define the (plain) Kolmogorov complexity of σ given τ

$$C(\sigma|\tau) = \min\{|\mu| \mid U^{\tau}(\mu) = \sigma\}$$

The reason to use $\bar{\tau}$ in the definition can be illustrated by the following example

$\min\{|\mu| \mid U^{\sigma}(\mu) = \sigma\}$ is not bounded by a constant, since we need

μ tells where is the end of σ in an oracle



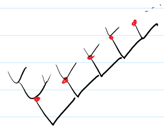
Proposition $C(\sigma^*_C \mid \sigma) = C(C(\sigma) \mid \sigma) \pm O(1)$

Exe Let $A \subset 2^{<\omega}$ be finite. Then for each τ there is $\sigma \in A$ s.t. $C(\sigma \mid \tau) \geq \log|A|$

Prefix-free Machine

Def We say a set $A \subseteq 2^{<\omega}$ is prefix-free (无前缀的), if no string in A is an initial segment of any other string in A. i.e. for any $\sigma, \tau \in A$, $\sigma \not\prec \tau$ and $\tau \not\prec \sigma$

example

(1)  (2)  (3) $\{\bar{\tau} \mid \tau \in 2^{<\omega}\}$

Def A prefix-free function / Machine / Program is a function / Machine / Program such that its domain is a prefix-free set

Note no prefix-free machine return the length of a string $\sigma \mapsto |\sigma|$

Let M be a prefix-free machine and $Z \in 2^{\omega}$, there is at most one $\sigma \prec Z$ s.t. $M(\sigma)\downarrow$.

Proposition (1) There is an effective enumeration of all prefix-free machine / Program / partial computable function.

(2) Therefore, there is a universal prefix-free machine

Proof

(1) One can effectively enumerate all prefix-free machine / Program by

$\Psi_0, \Psi_1, \cdots$ where $\Phi_0, \Phi_1, \cdots$ is the effective enumeration of all machine

To obtain Ψ_e

One just add an "inspector" to every program Φ_e, every time $\Phi_{e,s}(\tau)\downarrow$, it check if there is $\sigma \prec \tau$ or $\tau \prec \sigma$ and $t < s$ s.t. $\Phi_{e,t}(\sigma)\downarrow$. If so prevent it from converge, if not, halt and output $\Phi_e(\tau)$

Note if Φ_e is prefix-free, then $\Psi_e \simeq \Phi_e$ so all prefix-free machine are on the list

(2) Consider the machine U s.t.

$$U(\underbrace{1\cdots1}_{e}0\sigma) \simeq \Psi_e(\sigma)$$

Then U is prefix-free, and U is universal in the sense:

For any prefix-free partial computable function f, there is a string ρ_f s.t.

$$U(\rho_f \sigma) \simeq f(\sigma) \quad \text{for any } \sigma \in 2^{<\omega}$$ □

Convention: (1) If M is a prefix-free machine, we write $K_M(\sigma)$ instead of $C_M(\sigma)$

2) We fix an Universal prefix-free machine U.

Def (prefix-free Kolgomorov complexity)

For any $\sigma \in 2^{<\omega}$, we define the prefix-free (Kolgomorov) complexity of σ

$$K(\sigma) = K_U(\sigma)$$

Similarly we define $K(\sigma \mid \tau) = \min\{|\mu| \mid U^{\tau}(\mu) = \sigma\}$

the universal prefix-free machine

Also similarly, for each $\sigma \in 2^{<\omega}$, we define σ^* to be the leftmost τ s.t.

$$U(\tau) = \sigma \text{ and } |\tau| = K(\sigma)$$

Clearly $C(\sigma) \le K(\sigma)$. But prefix-free complexity is not so bad.

Proposition If $h: 2^{<\omega} \to 2^{<\omega}$ is computable. Then $K(h(\sigma)) \le K(\sigma) + O(1)$

The proof is essentially the same with the case for C

Exe

Since the function $\bar{\sigma} \to \sigma$ is partial computable and prefix-free

so

Proposition $K(\sigma) \le 2|\sigma| + O(1)$, actually $K(\sigma) \le 2|\sigma| + c_m$, where c_m is coding constant for the function $\bar{\sigma} \to \sigma$

Later, we will provide better upper bound

K is approximated from above (the same for C)

For $\sigma \in 2^{<\omega}$, $s \in \mathbb{N}$, define $K_s(\sigma) = \begin{cases} \min\{|\tau| \mid U_s(\tau)\downarrow = \sigma\} & \text{if there is such } \tau \text{ with } |\tau| < 2|\sigma| + c_m \\ 2|\sigma| + c_m & \text{otherwise} \end{cases}$

Note $U_s(\tau)\downarrow$ only for $|\tau| < s$ so

$(s, \sigma) \to K_s(\sigma)$ is a computable function

Note also $K_{s+1}(\sigma) \le K_s(\sigma)$

and $K(\sigma) = \lim_{s \in \mathbb{N}} K_s(\sigma)$

A better upper bound for K

Proposition $K(\sigma) \le K(|\sigma|) + |\sigma| + O(1) \le 2 \log |\sigma| + |\sigma| + O(1)$

Prf Consider the machine M: input τ, it search for ν, ρ s.t. $\tau = \nu\rho$, $U(\nu)$ codes the number n and $|\rho| = n$

if such composition (ν, ρ) founded, output ρ.

Since U is prefix-free, M is prefix-free

For each σ, $M(n^*\sigma) = \sigma$, where n^* is the leftmost ν s.t. $U(\nu)$ codes n.

So $K(\sigma) \le K_M(\sigma) + O(1) = K(|\sigma|) + |\sigma| + O(1)$ □