Lesson 4
Entropy
4-4: Prefix-Free Complexity

Jan Reimann

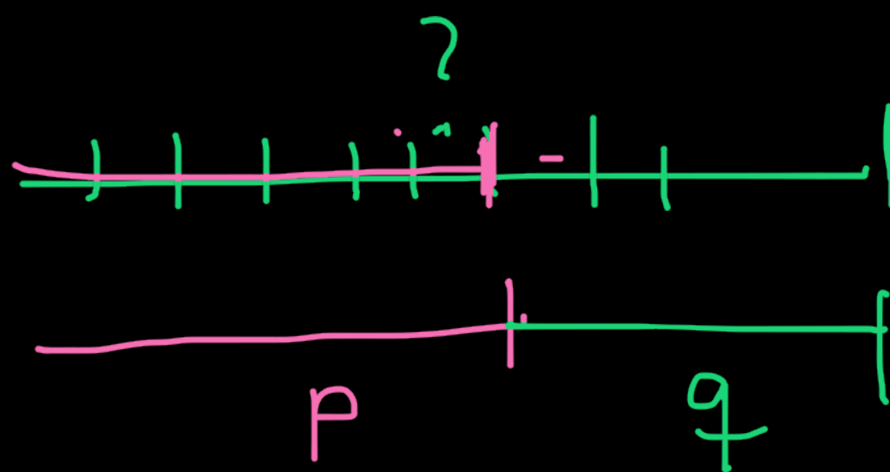
Math 574, Topics in Logic
Penn State, Spring 2014

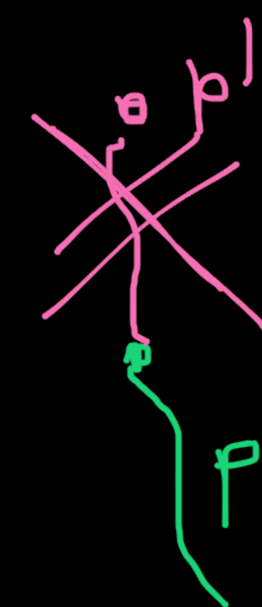
Prefix-Free Machines

Problem: If we concatenate two programs $\boldsymbol{p}, \boldsymbol{q}$ for $\boldsymbol{\sigma}, \boldsymbol{\tau}$, respectively, we cannot use it as a program for $\langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle$, since we lose the information where $\boldsymbol{p}$ ends and $\boldsymbol{q}$ starts.

$$C(\sigma, \tau) \leq^{+} C(\sigma) + C(\tau)$$

p q





Solution: Require that machines are prefix-free:

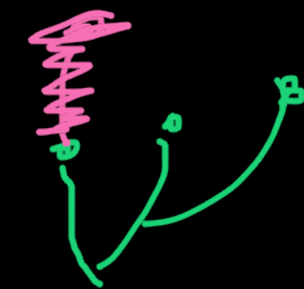
$$M(p)\downarrow \implies M(p')\uparrow \text{ for all } p' \sqsupset p.$$

In other words: the domain of a prefix-free machine is a prefix-free set of strings.

Universal Prefix-Free Machines

We define a variant of Kolmogorov complexity based on prefix-free Turing machines.

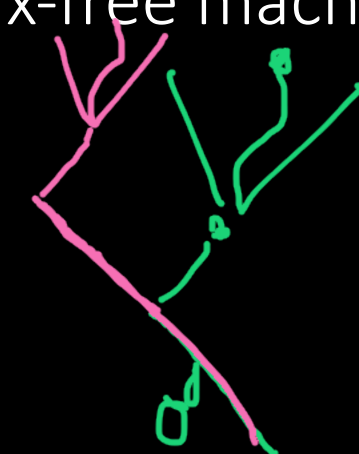
Idea: Use a universal prefix-free machine, i.e. a machine universal among the prefix-free machines.



Observations:

1. There exists an effective listing $(M'_d)_{d\in\mathbb{N}}$ of all prefix-free TMs.

   Modify an enumeration (M_e) of all TMs so that if we encounter a TM whose domain is not prefix-free, we change it to a prefix-free machine.



2. If we have such a listing $(M'_d)_{d\in\mathbb{N}}$, define

$$U'(0^d 1p) = \underline{M'_d(p)}.$$

   Then U' is prefix-free and can emulate all other prefix-free machines.

Prefix-Free Complexity

We define the prefix-free Kolmogorov complexity as

$$K(\sigma) = C_{U'}(\sigma) = \min\{|p| \colon U'(p) = \sigma\}.$$

The invariance theorem for prefix-free complexity is proved just like the ``plain'' case.

Now we easily get

$$K(\sigma, \tau) \leqslant^{+} K(\sigma) + K(\tau)$$

Suppose p, q are U'-programs for σ, τ, respectively. Then there is a prefix-free machine M' such that $M'(p \frown q) = \langle \sigma, \tau \rangle$:

Using bootstrapping, M' tests every initial segment ϑ of its input to see whether $U'(\vartheta)\downarrow$.

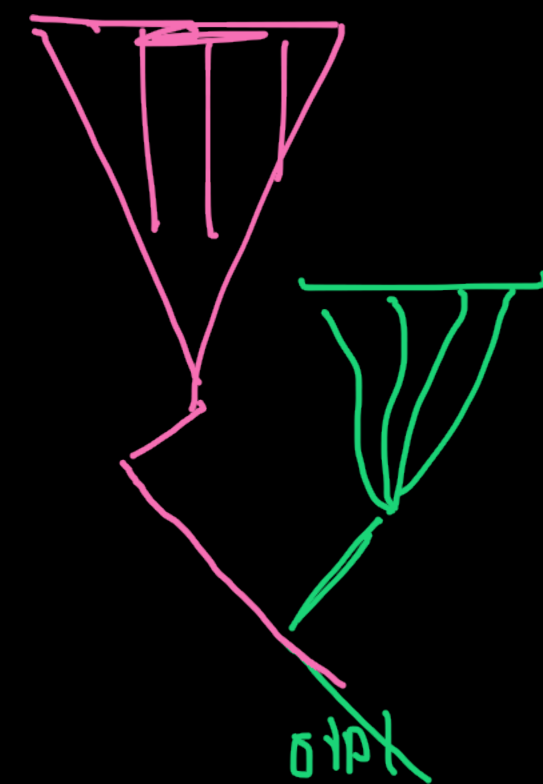
If so, output the result and run U' on the remaining string.

An Upper Bound on K

On the other hand, we no longer have the bound

$$K(\sigma) \leqslant^{+} |\sigma|.$$

The copy-machine $M(p) = p$ is not prefix-free!



How complex can a string get with respect to K?

We can implement a ``prefix-free version'' of the copy machine: $M'(0^{|p|}1p) = p$. This gives an upper bound of $K(\sigma) \leqslant^{+} 2|\sigma|$.

A Better Bound

The prefix-free copy machine $M'(0^{|p|}1p) = p$ is rather inefficient:

Instead of giving the length of p (as 0s), we could just give a shortest program for $|p|$.

U' prefix-free

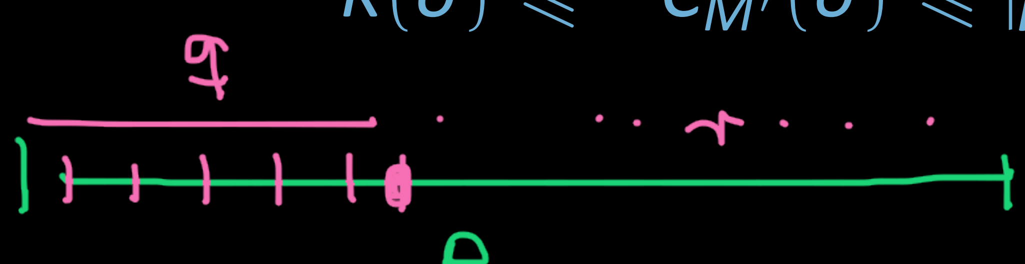
THM: $K(\sigma) \leqslant^{+} \underbrace{|\sigma|}_{\text{bits}} + K(|\sigma|)$

information through length

Proof:

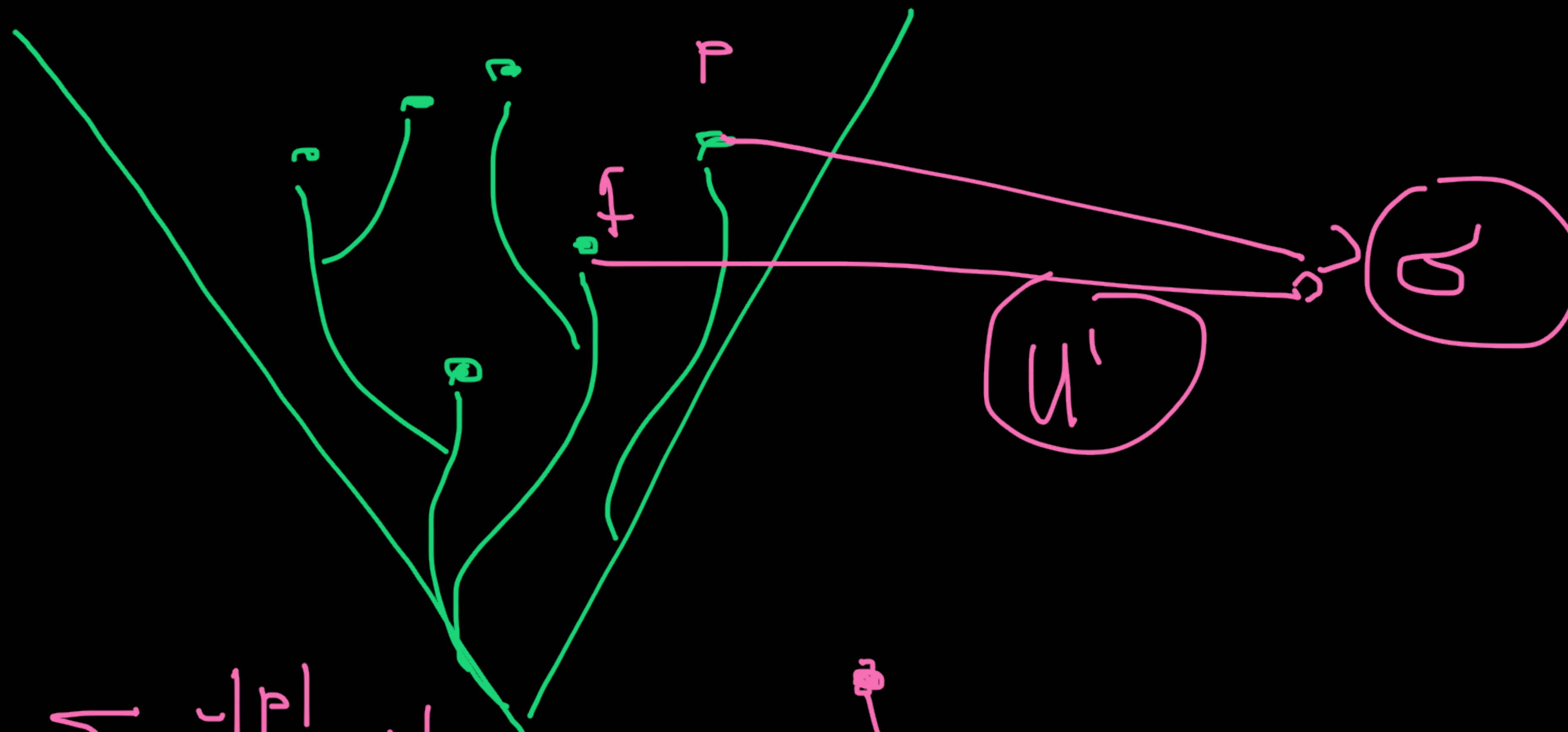
- Consider the ~~prefix~~-free machine M' that on input p, scans p for a "splitting" $p = q \frown r$ such that $U'(q) = |r|$.
- Since U' is prefix-free, there is at most one such splitting.
- If it exists, put $M'(p) = r$ (otherwise $M'(p)\uparrow$).
- Now use the invariance theorem: If p is a shortest U'-program for $|\sigma|$, then $M'(p \frown \sigma) = \sigma$ and hence

$$K(\sigma) \leqslant^{+} C_{M'}(\sigma) \leqslant |p \frown \sigma| = |p| + |\sigma| = \underbrace{K(|\sigma|)}_{=|p|} + |\sigma|.$$



$U'(q) = |r|$

Algorithmic Probability



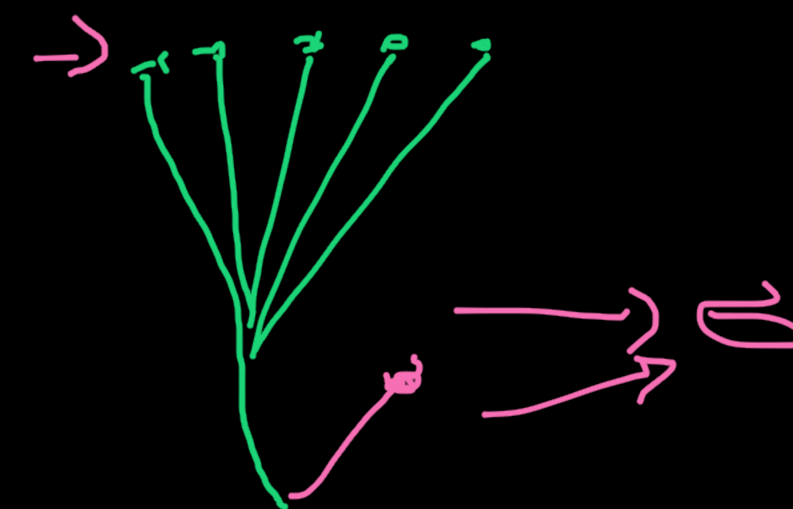
$$\sum 2^{-|p|} \le 1$$

$U'(p)\downarrow$

Algorithmic Probability

Define

$$P(\sigma) = \sum_{U'(p)=\sigma} 2^{-|p|}$$



Q: How is *P* related to *K*?

Could it be that

$$K =^{+} -\log P?$$

$\longrightarrow$ Coding Theory