Kraft Inequality

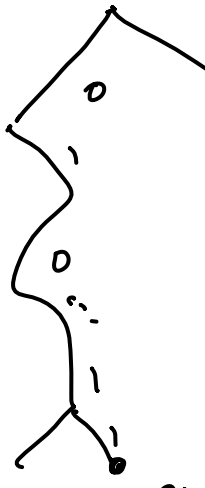
If P is a prefix code, then

$$\sum_{x \in P} \frac{1}{2^{|x|}} \leq 1.$$

Proof. (Probabilistic proof)

Toss a fair coin independently at random denoting Heads by 0, tails by 1, until we hit some string $x \in P$. [The experiment is not guaranteed to halt]

Let E_x be the event that the experiment halts with $x \in P$ as output.



$$P(E_x) = \frac{1}{2^{|x|}}.$$

Note: If x & y are in P, then E_x & E_y are disjoint.

So $\Pr\left(\bigcup_{x \in P} E_x\right)$

$= \sum_{x \in P} \Pr(E_x)$

$= \sum_{x \in P} \frac{1}{2^{|x|}}$

≤ 1

E_x.

$P(E_x \mid E_y) = P(E_x)$ if E_x & E_x are independent.

But if E_x & $\bar{E}_y$ are disjoint $P(E_x \mid E_y) = 0$, so $P(E_x) = 0$. If the

Space is finite, $E_x = \phi$.

Concatenation of Pairs of strings

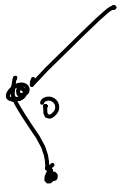
0 0

0 01

01 0

01 01

Converse may not hold $\hat{P} = \{0, 01\}$ is not a prefix code



But $\frac{1}{2^{|01|}} + \frac{1}{2^{|011|}} = 1/2 + 1/4 \leq 1$.

Mc Millan's Inequality

$\hat{P}$ satisfies $\sum_{x \in \hat{P}} \frac{1}{2^{|x|}} \leq 1$ iff $\hat{P}$ is a uniquely decodable code.

Self-delimiting Turing Machines.

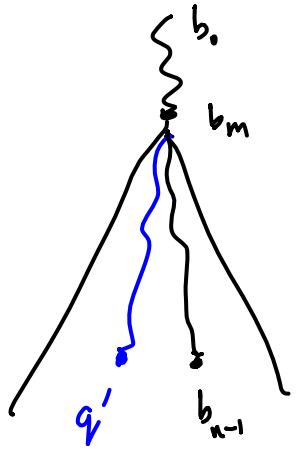
We restrict the set of programs to be a prefix set.

We produce a computable enumeration $\phi'_1, \phi'_2, \ldots$ of unary partial computable functions each of whose domain is a prefix-free set.

We know the set of all Turing machines $T_1, T_2, \ldots$ is computably enumerable, hence the set of all part. comp. functions $\varphi_1, \varphi_2, \ldots$ (whose domains may or may not be prefix-free) is c.e.

Idea: Simulate T_1 as long as domain of T_1 is a prefix free set. O/W our new machine differs from T_1 while ensuring that the new domain is prefix-free.

$\hat{T}_i$ (Simulate T_i if $\mathrm{dom}(T_i)$ is a prefix code)



1. Set $p = \lambda$, $m = -1$.

2. Input $b_0 \cdots b_{n-1}$

3. Dove tail over all strings q, the computation $T_i(pq)$.

Let q' be a string that makes $T_i(pq)$ halt, if one such exists. If $q' = \lambda$, then output $\phi_i(p)$.

Otherwise $m = m+1$, $p = b_0 \cdots b_{m-1} b_m$.

Let T_i compute φ_i, and $\hat{T}_i$ compute $\hat{\varphi}_i$.

Claim If $\mathrm{dom}(T_i)$ is a prefix code, then $\mathrm{dom}(T_i) = \mathrm{dom}(\hat{T}_i)$ and $\varphi_i = \hat{\varphi}_i$.

Claim In all cases, $\mathrm{dom}(\hat{T}_i)$ is a prefix code.