Hashing (cont.)

$h_{a,b}(x) = ((ax+b) \bmod p) \bmod m$

Thm: The family $\mathcal{H} = \{ h_{a,b} \mid a,b \in \mathbb{Z}_p \}$ is 2-universal for any prime $p \geq m$.

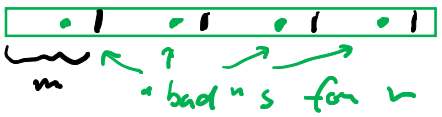
Pf: $x \neq y$ fixed. For a pair $a,b \in \mathbb{Z}_p$ define

$r = (ax+b) \bmod p$
$s = (ay+b) \bmod p$
(inner values of $h_{a,b}$)

$\Rightarrow$ $(a,b) \mapsto (r,s)$ is bijective map:
regular system of 2 equations over $\mathbb{Z}_p$ ($x \neq y$) (a,b unknown)

$\Rightarrow$ if (a,b) uniformly random, then (r,s) uniformly random

$h_{a,b}(x) = h_{a,b}(y)$ iff $r \equiv s \bmod m$
collision — (r,s) "bad"

$\mathbb{Z}_p$: 

For each r there is $\leq \lceil p/m \rceil$ "bad"s

$$\Pr[h_{a,b}(x) = h_{a,b}(y)] \leq \frac{p \lceil p/m \rceil}{p^2} \leq \frac{p+m-1}{m \cdot p} \leq \frac{2p-1}{m \cdot p} < \frac{2}{p}$$

(bad pairs / all pairs; $m \leq p$; $\lceil p/m \rceil \leq \frac{p+m-1}{m}$)

(if p large enough, then $(1+\varepsilon)$-univ.)

Note: The family $\mathcal{H}' = \{ h_{a,b} \mid a,b \in \mathbb{Z}_p, a \neq 0 \}$ is 1-universal for any prime $p \geq m$. (exercise)

ex: $\mathcal{H} = \{h_1, h_2\}$, $h_i : \{0,1\}^2 \to \{0,1\}$, $h_i(x_1 x_2) = x_i$ (projection of the i-th bit)

x	00	01	10	11
$h_1(x)$	0	0	1	1
$h_2(x)$	0	1	0	1

$x \neq y$
$\Pr_{h \in \mathcal{H}}[h(x) = h(y)] = \begin{cases} 0 & \text{if } x = y \\ 1/2 & \text{else} \end{cases}$

$\Rightarrow$ $\mathcal{H}$ is 1-universal

but: $\Pr[h(00) = 0] = 1$, or (h does not map equally to all bucket)
$\Pr[h(01) = 0 \wedge h(10) = 1] = 1/2$ ($h(01)=0$, $h(x)=1$ not independent)

$\Rightarrow$ does not "behave as" a totally random function (vulnerable)

(k,c)-independent families

Def: A family $\mathcal{H}$ of functions $h: U \to [m]$ is **(k,c)-independent** for some $k \geq 1$ and $c \geq 1$ if for every distinct $x_1, \dots, x_k \in U$ and every $a_1, \dots, a_k \in [m]$ (do not have to be distinct)

$$\Pr_{h \in \mathcal{H}}[h(x_1) = a_1 \wedge \dots \wedge h(x_k) = a_k] \leq \frac{c}{m^k}.$$

- $\Pr_{h \in \mathcal{H}}$: uniformly
- $h(x_1) = a_1 \wedge \dots \wedge h(x_k) = a_k$: k distinct elements are hashed "independently"
- $\frac{c}{m^k}$: at most c-times worse than totally random

Note: $\mathcal{H}$ is k-independent if (k,c)-independent for some constant c. (independent on k, m)

- (k,c)-independent for $k > 1 \Rightarrow (k-1, c)$-independent
- $(2,c)$-independent $\Rightarrow$ c-universal (exercises)
- $(1,c)$-independent does not have to be c-universal:

  $\mathcal{H} = \{h_i \mid i \in [m]\}$, $h_i(x) = i$ (constant function)

  $\Pr[h(x) = a] = 1/m$ but $\Pr[h(x) = h(y)] = 1$
  $\forall x, \forall a$ $\forall x \neq y$

How to construct k-independent families?

Polynomial hashing

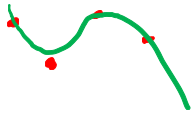
evaluating polynomial at x $h_t: \mathbb{Z}_p \to \mathbb{Z}_p$

$$h_t(x) = \sum_{i=0}^{k-1} t_i x^i \pmod p \quad \text{where } k \geq 1,\ t = (t_0, \dots, t_{k-1}) \in \mathbb{Z}_p^k,\ p \text{ prime}$$

Thm: The family $\mathcal{H} = \{h_t \mid t \in \mathbb{Z}_p^k\}$ is $(k,1)$-independent for any $k \geq 1$, prime p.

Pf: $x_1, \dots, x_k \in \mathbb{Z}_p$ distinct, $a_1, \dots, a_k \in \mathbb{Z}_p$

By Lagrange's interpolation, there is a **unique** polynomial h of degree at most $k-1$ s.t. $h(x_i) = a_i$ $\forall i$.



$$\Rightarrow \Pr_h[h(x_1) = a_1 \wedge \dots \wedge h(x_k) = a_k] = \frac{1}{p^k} \quad \square$$

h: choice of the polynomial

In particular, $\mathcal{H} = \{(ax + b) \bmod p \mid a, b \in \mathbb{Z}_p\}$ is $(2,1)$-independent.

Note: $h_t: \mathbb{Z}_p \to \mathbb{Z}_p$ useless for pure hashing, but from $\mathcal{P}_2$ we can construct other k-independent families.

Composition mod m

does not need to be prime

Lem: Let $\mathcal{H}'$ be a $(2,c)$-independent family of $h': U \to [r]$, $r \geq m$. Then $\mathcal{H} = \mathcal{H}' \bmod m = \{ h'(x) \bmod m \mid h' \in \mathcal{H}' \}$ is $(2,4c)$-independent and $2c$-universal.

we already know

ex: $\mathcal{H} = \mathcal{P}_2 \bmod m$ is $(2,4)$-independent and 2-universal.

Pf: a) $2c$-universality: $x_1 \neq x_2$, $h = h' \bmod m$, $h' \in \mathcal{H}'$

$$\Pr[h(x_1) = h(x_2)] = \sum_{i_1 = i_2 \bmod m} \Pr[h'(x_1) = i_1 \wedge h'(x_2) = i_2] \leq \sum_{i_1 = i_2 \bmod m} \frac{c}{r^2} \quad \text{($2c$)-ind.}$$

$$\leq r \cdot \lceil r/m \rceil \frac{c}{r^2} \leq r \frac{2r}{m} \frac{c}{r^2} = \frac{2c}{m} \qquad \lceil r/m \rceil \leq \frac{r+m-1}{m} \leq \frac{2r}{m}$$

choice of i_1; i_2 s.t. $i_2 = i_1 \bmod m$

b) $(2,4c)$-independence: $x_1 \neq x_2$, $a_1, a_2 \in [m]$ (fixed)

$$\Pr[h(x_1) = a_1 \wedge h(x_2) = a_2] = \sum_{\substack{i_1 = a_1 \bmod m \\ i_2 = a_2 \bmod m}} \Pr[h'(x_1) = i_1 \wedge h'(x_2) = i_2] \leq \sum_{\substack{i_1 = a_1 \\ i_2 = a_2}} \frac{c}{r^2} \quad \text{($2c$)-ind}$$

$$\leq \lceil r/m \rceil^2 \frac{c}{r^2} \leq \left(\frac{2r}{m}\right)^2 \frac{c}{r^2} = \frac{4c}{m^2} \qquad \square$$

choices of i_1, i_2

Note: straightforward generalization $\mathcal{H}'$ (k,c)-ind $\Rightarrow$ $\mathcal{H}$ $(k, 2^k c)$-ind.

stronger requirement

Lem. Let $\mathcal{H}'$ be a (k,c)-independent family of $h': U \to [r]$, $r \geq 2km$.

Then $\mathcal{H} = \mathcal{H}' \bmod m$ is $(k, 2c)$-independent.

much better than

(without proof)

ex: $\mathcal{P}_k \bmod m = \{ h_t \bmod m \mid t \in \mathbb{Z}_p^k \}$ is $(k,2)$-independent if

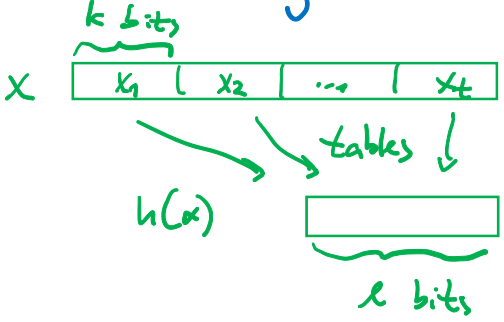
$$h_t(x) = \sum_{i=0}^{km} t_i x^i \pmod p$$

$p \geq 2km$.

prime

Tabulation hashing

idea: totally random function is possible for small universe



$h: [2^{k \cdot t}] \to [2^\ell]$ (u, m)

$h(x) = \bigoplus_{i=1}^{t} T_i(x_i)$ ($\oplus$ = bitwise xor)

all chosen independently

$T_i : [2^k] \to [2^\ell]$ totally random table

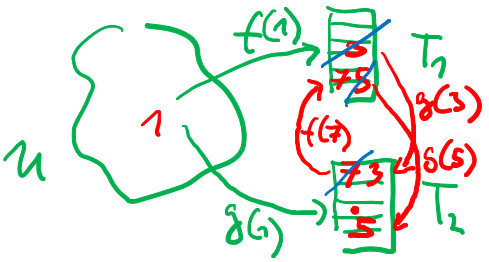
- t tables of 2^k entries per ℓ bits $\Rightarrow 2^k \cdot t \cdot \ell$ bits (vs $2^{kt} \cdot \ell$ for totally random)
- fast, cache friendly (choose k s.t. fits into cache)

Claim: Tabulation hashing (parameterized by the tables $T_1, \dots, T_t$) is 3-independent but not 4-independent. (exercise)

Note: Nevertheless, tabulation has some nice properties (4th moment bounds, Chernoff-type bounds)

Cuckoo hashing ← method of resolving collisions

2 hash tables, 2 hashing functions, conflicts resolved by replacing to the other table until success or timeout $\sim \log n$ steps — then rehash all items with new h-functions.



invariant: every item x is either at $T_1[f(x)]$ or $T_2[g(x)]$

$\Rightarrow$ FIND, DELETE in $O(1)$ time (worst case).

← for a variant with a single table (and 2 h-functions)

Thm: Let $\varepsilon > 0$, $m \geq (2+\varepsilon)n$ and f, g be chosen uniformly in random from a $\lceil 6 \log n \rceil$-independent family. Then Cuckoo hashing with $\lceil 6 \log n \rceil$ insertion timeout has $O(1)$ expected time for INSERT. (without proof)

Note: Tabulation hashing suffices although only 3-independent.