10. k-independent hash functions

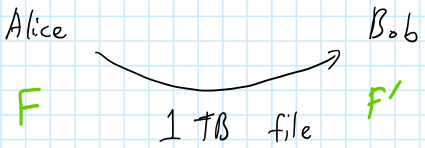
Thursday, September 30, 2021 2:02 AM

Last time: Intro to universal hash functions

Today: k-independent hash families

Application to signatures/checksums

Problem:



How to check for correct receipt?

One solution: hashing with universal hash.

Check if hash(F) = hash(F').

Problem: Assign a unique signature $s(x)$ $\forall x \in S$, $|S| = n$.
i.e. want $s(x) \neq s(y)$ $\forall x, y \in S$, $x \neq y$.

Choose universal hash function $s : U \to [n^3]$. ← how big the range?

$$\Pr_s\left[\exists \{x,y\} \subseteq S : s(x) = s(y)\right] \le \sum_{\{x,y\} \subseteq S} \Pr_s\left[s(x) = s(y)\right] \leqq \frac{\binom{n}{2}}{n^3} < \frac{1}{2n}.$$

union bound

birthday problem

Application to sampling from sets

Let $A \subseteq U$. Can we determine some statistic on A without storing all of it?

Recall LogLog cardinality estimation

e.g. $|A|$ or polling

Let's use sampling: Let $h : U \to [m]$ and let $t \ll m$.

Let's sample $x \in U$ if $h(x) < t$. Let $S_{h,t}(A) = \{x \in A \mid h(x) < t\}$.

Then $\mathbb{E}\left[|S_{h,t}(A)|\right] = |A| \cdot \frac{m}{t}$ Right?

So we can estimate $|A|$.

WRONG Because we didn't actually include uniformity in the definition of universal

---

Strong universality (a.k.a. 2-independence)

A random hash function $h : [u] \to [m]$ is strongly universal if

A random hash function $h : [u] \longrightarrow [m]$ is strongly universal if

$$\text{Prob}[h(i_1) = j_1 \wedge h(i_2) = j_2] = \frac{1}{m^2}.$$

Lemma: Strong universality implies universality.

Observation 3.1 Strong universality is equivalent to the statement that each key is hashed uniformly into $[m]$ and that every two distinct keys are hashed independently.

proof. Forward case. Let $h : [u] \to [m]$ be strongly universal. Let $x \neq y \in [u]$

Clearly $\forall q \in [m]$, $\text{Prob}[h(x) = q] = \sum_{r \in [m]} \Pr[h(x) = q \wedge h(y) = r] = \frac{m}{m^2} = \frac{1}{m}$, so uniformity holds.

Furthermore, $\Pr[h(x) = q \mid h(y) = r] = \dfrac{\Pr[h(x) = q \wedge h(y) = r]}{\Pr[h(y) = r]} = \dfrac{\frac{1}{m^2}}{\frac{1}{m}} = \underset{\substack{\uparrow \\ \text{independence}}}{\frac{1}{m}} = \Pr[h(x) = q]$.

Backward case: If $h(x)$ and $h(y)$ are independent and uniform,

$$\Pr[h(x) = q \wedge h(y) = r] = \Pr[h(x) = q] \cdot \Pr[h(y) = q] = \frac{1}{m^2}.$$

Also called 2-independence to focus on the independence of two events, rather than on collision probabilities.

k-independence

Def. $\mathcal{H}$ is a k-wise independent hash family if

$$\forall\, i_1 \neq i_2 \neq \cdots \neq i_k \in [u] \quad \text{and} \quad \forall\, j_1, \ldots, j_k \in [m],$$

$$\underset{h \sim \mathcal{H}}{\text{Prob}}(h(i_1) = j_1 \wedge \cdots \wedge h(i_k) = j_k) = \frac{1}{m^k}.$$

i.e. independence of k-different events.

Ex. The set $\mathcal{H}$ of ALL functions $[u] \longrightarrow [m]$ is k-wise ind. $\forall\, k$.

$|\mathcal{H}| = m^u$, so $h \in \mathcal{H}$ is representable in $u \lg m$ bits.

↑ This is just picking an ideal hash function which maps everything i.i.d. uniformly.

Ex. Let $u = m = q$, where q is a prime power.

↓ Galois field

Ex. Let $u = m = q$, where q is a prime power   Galois field

Let $\mathcal{H}_{\text{poly-}k}$ be the set of all deg $k-1$ polynomials in $\mathbb{F}_q[x]$.

Claim: $\mathcal{H}_{\text{poly-}k}$ is a k-wise independent family.

proof. Lagrange interpolation. If we know $i_1, \dots, i_k$ distinct and $j_1, \dots, j_k$, then

$$p(x) = \sum_{r=1}^{k} \left(\frac{\prod_{y \in [k] \setminus \{r\}} x - i_y}{\prod_{y \in [k] \setminus \{r\}} i_r - i_y} \right) \cdot j_r \quad \text{satisfies} \quad p(i_r) = j_r \quad \forall r.$$

i.e.

$$p(x) = \frac{(x-i_2)(x-i_3)\cdots(x-i_k)}{(i_1-i_2)(i_1-i_3)\cdots(i_1-i_k)} j_1 + \frac{(x-i_1)(x-i_3)\cdots(x-i_k)}{(i_2-i_1)(i_2-i_3)\cdots(i_2-i_k)} j_2 + \cdots + \frac{(x-i_1)\cdots(x-i_{k-1})}{(i_k-i_1)\cdots(i_k-i_{k-1})} j_k$$

When $p(i_2)$ $\qquad \| \; 0 \qquad\qquad \| \; j_2 \qquad\qquad \| \; 0$

Notice the $p(x)$ is still of course a polynomial, of degree $k-1$.

Furthermore, $p(x)$ as given above is the unique poly of degree $\le k-1$ where $p(i_r) = j_r$ $\forall r$.

Why? Suppose some other poly $f(x)$ has $\deg(f) \le k-1$ and $f(i_r) = j_r$. $\forall r$.

Then $g(x) = p(x) - f(x)$ has deg at most $k-1$ and $g(i_r) = 0$ $\forall r$.

Further $g(x) \neq 0$, since $p(x) \neq f(x)$.

But also $g(x) = c(x-i_1)(x-i_2)\cdots(x-i_k)$, so g is of degree $k > k-1$, a contradiction. Thus, $p(x) = f(x)$.

Thus, $p(x) = \alpha_{k-1} x^{k-1} + \dots + \alpha_1 x + \alpha_0$, so it is determined by k elements of $\mathbb{F}_q$.

Thus $|\mathcal{H}_{\text{poly-}k}| = q^k$.

But as just shown, exactly one such polynomial goes through all of (i_r, j_r),

so $\displaystyle \Pr_{h \sim \mathcal{H}_{\text{poly-}k}} \left(h(i_1) = j_1 \wedge \cdots \wedge h(i_k) = j_k \right) = \frac{1}{q^k}.$ ☑

Aside: Each $h \in \mathcal{H}_{\text{poly-}k}$ is representable using $k \lg q$ bits.

Practical choices for q,   e.g. $P = x^{64} + x^4 + x^3 + x + 1$

Practical choices for q

e.g. $P = x^{64} + x^4 + x^3 + x + 1$

Could use $\mathbb{F}_q$, where $q = 2^{64}$?

cannot be factored

Recall: $\mathbb{F}_{2^{64}} = \mathbb{F}_2[X]/(P)$, where P is an irreducible poly in $\mathbb{F}_2[X]$ of degree n.

$z \in \mathbb{F}_{2^{64}}$ can be written $\underbrace{a_{n-1}x^{n-1} + \cdots + a_1 x + a_0}_{\in \{0,1\}} \in \mathbb{F}_2[x]$.

$\Rightarrow$ can write as $a_{n-1}a_{n-2}\cdots a_0$, a 64-bit binary number.

Addition:

$$\begin{array}{r} a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \\ + \; b_{n-1}x^{n-1} + \cdots + b_1 x + b_0 \\ \hline (a_{n-1}+b_{n-1})x^{n-1} + \cdots + (a_0 + b_0) \end{array} \qquad \begin{array}{r} a_{n-1} \cdots a_0 \\ \text{XOR } \; b_{n-1} \cdots b_0 \\ \hline (a_{n-1} \oplus b_{n-1}) \cdots (a_0 \oplus b_0). \end{array}$$

Multiplication requires Euclidean division by P, which can be expensive.

Prime fields: Let p be a large prime.

Then $\mathbb{F}_p \cong \mathbb{Z}/p\mathbb{Z}$, and multiplication just requires mod p.

Alternative choice:

Mersenne Primes are prime numbers of the form $2^n - 1$.

Recall: If n is composite, so is $2^n - 1$

proof.

$$2^{ab} - 1 = (2^a - 1)(1 + 2^a + 2^{2a} + \cdots + 2^{(b-1)a})$$
$$= (2^b - 1)(1 + 2^b + 2^{2b} + \cdots + 2^{(a-1)b})$$

☒

So n must be prime if $2^n - 1$ is prime.

OEIS A000043: Mersenne exponents $n = 2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127$

Often, we choose $p = 2^{31} - 1$, $2^{61} - 1$, or $2^{89} - 1$.

Why?

Claim: If $p = 2^q - 1$, p and q prime, then

$$x \equiv x \bmod 2^q + \lfloor x / 2^q \rfloor \pmod{p}.$$

Claim If $p = 2^q - 1$, p and q prime, then

$$x \equiv x \bmod 2^q + \lfloor x/2^q \rfloor \pmod p.$$

proof. Let $x = a\,2^q + b$, where $b < 2^q$

$\uparrow$ upper bits of x $\qquad$ $\uparrow$ lower q bits of x

$$x \bmod p \equiv (a \bmod p)\underbrace{(2^q \bmod p)}_{\substack{2^q \bmod 2^q - 1 \\ = 1}} + (b \bmod p)$$

$$\equiv (a+b) \bmod p$$

But $a = \lfloor x/2^q \rfloor$ is the upper bits of x
$b = x \bmod 2^q$ is the lower bits of x

$$\Rightarrow \qquad x \equiv x \bmod 2^q + \lfloor x/2^q \rfloor \pmod p$$

▨

$(x >> q)$ $\lfloor x/2^q \rfloor$ is a bit-shift to the right by q.

$(x \& p)$ $\lfloor x \bmod 2^q \rfloor$ is a bit-mask operation by p.

Then $y = x \bmod p$ can be computed by

$$\begin{cases} y \leftarrow (x \& p) + (x >> q); \\ \text{if } (y >= p), \quad y \leftarrow y - p. \end{cases}$$

which are all fast bit operations.

Universal hashing of variable-length strings

Consider $x_0 x_1 \cdots x_d \in [u]^d$ $\quad$ (each $x_i \in [u]$)

Can we construct an almost universal hash family $[u]^d \to [q]$?

Let q be a prime power, and work over $\mathbb{F}_q$.

$$\text{Let} \quad P_{x_0 \cdots x_d}(\alpha) = \sum_{i=0}^{d} x_i \alpha^i$$

Let $h_a(x_0 \cdots x_d) = P_{x_0 \cdots x_d}(a)$, where $a \in \mathbb{F}_q$ uniformly drawn.

Claim:
If $y_0 \cdots y_{d'}$ is some other string with $d' \leq d$, then

$$\Pr_{a \in \mathbb{F}_q}\left[h_a(x_0 \cdots x_d) = h_a(y_0 \cdots y_{d'}) \right] \leq \frac{d}{q}.$$

proof. $h_a(x_0 \cdots x_d) = h_a(y_0 \cdots y_{d'})$

$$\Rightarrow \underbrace{P_{x_0 \cdots x_d}(a) - P_{y_0 \cdots y_d'}(a)}_{\text{degree } d \text{ poly in } \mathbb{F}_q[\alpha].} = 0$$

By fundamental thm of algebra, the poly $P_{x_0 \cdots x_d} - P_{y_0 \cdots y_d'}$ has at most d distinct roots.

So the prob. a random $a \in \mathbb{F}_q$ is a root is at most $\frac{d}{q}$. $\square$