CS157 Homework 6

BY TQIAN AND SILAO_XU

March 12, 2013

Problem 3

1. When choosing a hash function, we want to make sure that collisions are unlikely. One way to ensure this is to randomly choose a hash function from a large family, where different functions in the hash function family scramble elements in different ways. A hash function family H is a family of functions $\{h_p\} : X \to Y$, where p ranges over *parameters* in a set P. Throughout this problem, we will let m be the size of the range of the hash function family, $m = |Y|$. Typically, a hash function is parameterized by several parameters; for example, if h is parameterized by triples $p = (p_1,\ p_2,\ p_3)$, where p_1 ranges over some set P_1, p_2 ranges over some set P_2, and p_3 ranges over some set P_3, then the universe of parameters P consists of all values of these triples. Specifically, $P = P_1 \times P_2 \times P_3$, and $|P| = |P_1| \cdot |P_2| \cdot |P_3|$.

   A hash function family H is called universal if for each pair $a, b \in X$ with $a \neq b$, at most $\frac{|P|}{m}$ out of the $|P|$ parameters p make a and b collide as $h_p(a) = h_p(b)$.

   For each of the following hash function families, either prove it is universal or give a counterexample. Additionally, compute how many bits are needed to choose a random element of the family (namely, compute $\log_2 |P|$ in each case).

   The notation $[m]$ denotes the set of integers $\{0, 1, 2, \ldots, m-1\}$.

   a) (3 points) $H = \{h_p : p \in [m]\}$ where m is a fixed prime and

   $$h_p(x) = px \bmod m.$$

   Each of these functions is parameterized by an integer p in $[m]$, and maps an integer x in $[m]$ to an output in $[m]$.

   The hash function family $H = \{h_p : p \in [m]\}$ where m is a fixed prime is universal if and only if for each pair $a, b \in X$ with $a \neq b$, at most $\frac{|P|}{m} = 1$ out of the $|P|$ paramaters p make a and b collide as $h_p(a) = h_p(b)$.

   Suppose $h_p(a) = h_p(b)$ and $a \neq b$, then we have

   $$\begin{aligned} pa \bmod m &= pb \bmod m \\ pa \bmod m - pb \bmod m &= 0 \\ p(a-b) \bmod m &= 0 \end{aligned}$$

   Because $a \neq b$ (and thus $a - b \neq 0 \bmod m$), m is bigger than a and b each, $(a-b)$ cannot be zero modulo m. Also m is prime, so p uniform at random. So left-hand side euqally likely to be any of $\{0,\ 1,\ 2,\ \ldots,\ m-1\}$ and there is $\frac{1}{m}$ probability to make the left-hand side of above equation become 0, which implies $\mathrm{Prob}[h_p(x) = h_p(y)] = \frac{1}{m}$.

   We conclude that for each pair $a, b \in X$ with $a \neq b$, at most $\frac{|P|}{m} = 1$ out of the $|P|$ parameters p could make a and b collide as $h_p(a) = h_p(b)$, the hash function family $H = \{h_p : p \in [m]\}$ is thus universal.

The number of bits for choosing a random element of the family is

$$\log_2(|P|) = \log_2(m)$$

b) (3 points) $H = \{h_{p1,p2}: p_1, p_2 \in [m]\}$ where m is a fixed prime and

$$h_{p1,p2}(x_1, x_2) = (p_1 x_1 + p_2 x_2) \bmod m.$$

Each of these functions is parameterized by a pair of intergers p_1 and p_2 in $[m]$, and maps a pair of integers x_1 and x_2 in $[m]$ to an output in $[m]$.

Suppose $h_{p_1,p_2}(x_1, x_2) = h_{p_1,p_2}(x_1', x_2')$ and the pair of intergers (x_1, x_2) and (x_1', x_2') differs in the first element, namely $x_1 \neq x_1'$, then we have

$$p_1(x_1 - x_1') \bmod m \;=\; p_2(x_2' - x_2) \bmod m$$

Our goal is to prove that p_1 uniform at random.

- Firstly, based on the "*Principle of Deferred Decisions*", we first fixed the choices for p_2, and then considered the effect of the random choice of p_1 given fixed p_2. So the right-hand side of above equation should be some fixed number ranges over $\{0, 1, 2, ..., m-1\}$, but p_1 still randomly ranges over $[m]$.

- Second of all, integer x_1 and x_1' are in $[m]$, so m is bigger than x_1, x_1' each. Together with the given condition that $x_1 \neq x_1'$, $x_1 - x_1'$ couldn't zero modulo m.

- Thirdly, m is prime.

Based on the above three ingredients, p_1 uniform at random and therefore the left-hand side equally likely to be any of $\{0,\ 1,\ ...,\ m-1\}$, which implies $\text{Prob}[h_{p_1,p_2}(x) = h_{p_1,p_2}(y)] = \frac{1}{m}$.

We conclude that for each pair $(x_1, x_2), (x_1', x_2') \in X$ with the pairs x_1, x_1' and x_2, x_2' not equivalent at the same time, at most $\frac{|P|}{m} = 1$ out of the $|P|$ parameters p_1, p_2 could make $(x_1,\ x_2)$ and $(y_1,\ y_2)$ collide as $h_{p_1,p_2}(x_1,\ x_2) = h_{p_1,p_2}(y_1,\ y_2)$, the hash function family $H = \{h_p(x) = p_1 x_1 + p_2 x_2 \bmod m\}$ is thus universal.

The number of bits for choosing a random element of the family is

$$\log_2(|P|) = \log_2(m^2)$$

c) (3 points) H is as in part 3.1b except m is now a fixed power of 2 (instead of a prime).

We'll give a counter-example such that the hash family $H = \{h_{p_1,p_2}(x_1,\ x_2) = p_1 x_1 + p_2 x_2 \bmod m\}$ where m is a fixed power of 2 is not universal.

Suppose $x_1 = 0, x_2 = 0$, then $h_{p_1,p_2}(x_1, x_2) = 0$ and suppose $x_1' = 0, x_2' = 2$, then

$$h_{p_1,p_2}(x_1 - x_2) \bmod m \;=\; 2p_2 \bmod m$$

If $h_{p_1,p_2}(x_1, x_2) = h_{p_1,p_2}(x_1', x_2') = 0$ then

$$2p_2 \bmod m = 0$$

Since m is a fixed power of 2, and $p_2 \in [m]$, so $p_2 = 0$ or $p_2 = \frac{m}{2}$, and p_1 could be any value in $[m]$.

Thus, we have $2m$ pairs of p_1, p_2 such that

$$\begin{aligned} h_{p_1,0}(0,0) &= h_{p_1,0}(0,2) && (p_1 \in [m]) \\ h_{p_1,\frac{m}{2}}(0,0) &= h_{p_1,\frac{m}{2}}(0,2) && (p_1 \in [m]) \end{aligned}$$

Since for pair $x_1 = 0$, $x_2 = 0$ and $x_1' = 0$, $x_2' = 2$, $|P| = 2m$, $\frac{|P|}{m} = 2$ out of the $|P| = m^2$ parameter pairs p_1, p_2 making x_1, x_2 and x_1', x_2' collide as $h_{p_1,p_2}(x_1, x_2) = h_{p_1,p_2}(x_1', x_2')$, so the hash function family $H = \{h_{p_1,p_2}(x_1, x_2) = p_1x_1 + p_2x_2 \bmod m\}$ is not universal.

d) (4 points) H is the set of all functions from pairs $x_1, x_2 \in [m]$ to $[m]$.

Suppose $P = P_1 \times P_2$, $|P| = n$ where n is unknown. The hash function family $H = \{h_p : X \rightarrow Y\}$ is universal only if for each pair of inputs x and $x' \in [m]$ with $x \neq x'$, at most $\frac{|P|}{m} = 1$ out of the $|P|$ parameters make a collision as $h_p(x_1, x_2) = h_p(x_1', x_2')$.

Case 1: m is prime

Suppose $h_{p_1,p_2}(x_1, x_2) = h_{p_1,p_2}(x_1', x_2')$ and the pair of intergers (x_1, x_2) and (x_1', x_2') differs in the first element, namely $x_1 \neq x_1'$, then we have

$$p_1(x_1 - x_1') \bmod m \;=\; p_2(x_2' - x_2) \bmod m$$

Our goal is to prove that p_1 uniform at random.

- Firstly, based on the "*Principle of Deferred Decisions*", we first fixed the choices for p_2, and then considered the effect of the random choice of p_1 given fixed p_2. So the right-hand side of above equation should be some fixed number ranges over $\{0, 1, 2, ..., m-1\}$, but p_1 still randomly ranges over $[n]$.

- Second of all, integer x_1 and x_1' are in $[m]$, so m is bigger than x_1, x_1' each. Together with the given condition that $x_1 \neq x_1'$, $x_1 - x_1'$ couldn't zero modulo m.

- Thirdly, m is prime.

Based on the above three ingredients, p_1 uniform at random and therefore the left-hand side equally likely to be any of $\{0, 1, ..., m-1\}$, which implies $\text{Prob}[h_{p_1,p_2}(x) = h_{p_1,p_2}(y)] = \frac{1}{m}$.

We conclude that for each pair $(x_1, x_2), (x_1', x_2') \in X$ with the pairs x_1, x_1' and x_2, x_2' not equivalent at the same time, at most $\frac{|P|}{m}$ out of the $|P|$ parameters p_1, p_2 could make (x_1, x_2) and (x_1', x_2') collide as $h_{p_1,p_2}(x_1, x_2) = h_{p_1,p_2}(x_1', x_2')$, the hash function family H is thus universal.

The number of bits for choosing a random element of the family is

$$\log_2(|P|) = \log_2(n)$$

Case 2: m is not prime

We will give a counter-example showing that the hash function family $H = \{h_p : X \rightarrow Y\}$ is not universal.

Suppose d divides m, let $x_1=0, x_2=0, x_1'=0, x_2'=d$, then if $h_p(x_1, x_2)=h_p(x_1', x_2')$, we have

$$h_{p_1,p_2}(x_1-x_2) \bmod m = dp_2 \bmod m$$

Both left-hand side and right-hand side of above equation range over $[m]$, so let $h_{p_1,p_2}(x_1, x_2)=h_{p_1,p_2}(x_1', x_2')=0$, *i.e.*

$$dp_2 \bmod m = 0$$

Because d divides m and $p_2 \in [n]$, p_2 coud either be 0 or $\frac{m}{d}$. We haven't condition on p_1 and thus p_1 could be any value in $[n]$. Now we have $2n$ pairs of (p_1, p_2) such that

$$\begin{aligned} h_{p_1,0}(0,0) &= h_{p_1,0}(0,d) && (p_1 \in [n]) \\ h_{p_1,\frac{m}{d}}(0,0) &= h_{p_1,\frac{m}{d}}(0,d) && (p_1 \in [n]) \end{aligned}$$

Since for pairs $x_1=0$, $x_2=0$ and $x_1'=0$, $x_2'=d$, there are $\frac{2n}{m}$ out of n parameter pairs p_1, p_2 making x_1, x_2 and x_1', x_2' collide as $h_{p_1,p_2}(x_1, x_2)=h_{p_1,p_2}(x_1', x_2')$, so the hash function family H is not universal in this case.

2. (3 points) Hacking a hash function: suppose for a member of the hash function family from part 3.1b you have found two inputs $(x_1,\ x_2)$ and $(x_1',\ x_2')$ that hash to the same value. Describe how to find further inputs that collide.

   (Suppose you are interacting with a server, and you start to suspect that the server is using a hash function like this. This sort of technique might be used to crash the server, if their hash function data structures are not implemented well.)

   Because we have found inputs (x_1, x_2) and (x_1', x_2') collide, we have

$$\begin{aligned} (p_1x_1 - p_2x_2) \bmod m &= (p_1x_1' - p_2x_2') \bmod m \\ ((p_1x_1+p_2x_2)-(p_1x_1'+p_2x_2')) \bmod m &= 0 \bmod m \\ ((p_1x_1+p_2x_2)+k(p_1x_1+p_2x_2-(p_1x_1'+p_2x_2'))) \bmod m &= (p_1x_1+p_2x_2) \bmod m && (k \in Z^+) \\ (p_1(x_1+k(x_1-x_1'))+p_2(x_2+k(x_2-x_2'))) \bmod m &= (p_1x_1+p_2x_2) \bmod m \\ h_{p_1,p_2}(x_1+k(x_1-x_1'), x_2+k(x_2-x_2')) &= h_{p_1,p_2}(x_1, x_2) \end{aligned}$$

   Similarly, we could also derive that

$$h_{p_1,p_2}(x_1'+k(x_1-x_1'), x_2'+k(x_2-x_2')) = h_{p_1,p_2}(x_1', x_2') \qquad (k \in Z^+)$$

   So we could find futher inputs pairs (x_1'', x_2'') as long as it commits to the form

$$\begin{aligned} x_1'' &= x_1+k(x_1-x_1') \\ x_2'' &= x_2+k(x_1-x_1') && (k \in Z^+) \end{aligned}$$

   or

$$\begin{aligned} x_1'' &= x_1'+k(x_1-x_1') \\ x_2'' &= x_2'+k(x_1-x_1') && (k \in Z^+) \end{aligned}$$

3. (7 points) A much stronger property than universal hashing is *k-independent hashing*. A hash function family $\{h_p\}: X \to Y$ is k-independent if for any distinct $x_1, \ldots, x_k \in X$ and any $y_1, \ldots, y_k \in Y$, for exactly a $\frac{1}{m^k}$ fraction of parameters p we will have $h_p(x_1)=y_1$ and $h_p(x_2)=y_2$ and ...$h_p(x_k)=y_k$.

   For a prime m consider the family of hash functions from $[m]$ to $[m]$ parameterized by $p=(p_0, \ldots, p_{k-1})$, where $h_p(x)=p_0+p_1x+p_2x^2+\ldots+p_{k-1}x^{k-1} \bmod m$. Show that this family is k-independent.

(Hint: Recall the familiar fact that for any k distinct real numbers x_1, ..., x_k, and any k real numbers y_1, ..., y_k, there is a *unique* degree $k-1$ polynomial that passes through these k pairs $(x_i,\ y_i)$. The same fact is true modulo a prime m. Assume and use this fact.)

Intuitively, if a hash-function is 10-independent, this means that for any 10 elements, their hash destinations will look as though they had been chosen uniformly at random. This is very useful for analyzing (and preventing) unfortunate hashing patterns involving up to 10 elements, because you can deduce that such patterns will occur no more often than if the hash destinations had been chosen at random.

Given the hash function family $H=\{h_p: X\rightarrow Y\}$ from $[m]$ to $[m]$, there are C_m^k ways of choosing k distinct inputs $x_1, x_2, ..., x_k$ from X and m^k different output because of m different values for each $y_i\in Y$, $i\in\{1, 2, ..., k\}$. So there will be $C_m^k\cdot m^k$ number of ways of mapping X to Y and the hash function family H is parameterized by the parameter set p where $|p|=C_m^k\cdot m^k$.

Assume we have the key claim: for any k distinct real numbers x_1, ..., x_k, and any k real numbers $y_1, ..., y_k$, there is a *unique* degree $k-1$ polynomial that passes through these k pairs (x_i, y_i). The same fact is true modulo a prime m. Based on this fact, there exists a hash function

$$h_p(x)=p_0+p_1x+p_2x^2+...+p_{k-1}x^{k-1} \bmod m$$

such that $X\rightarrow Y$ from $[m]$ to $[m]$ is parameterized by a unique parameter set.

For output y_1, y_2, ..., y_k, we can choose a distinct input x_1, x_2..., x_k from $[m]$ such that $h_p(x_i)=y_i$ for each $i\in\{1,\ 2,\ ...,\ k\}$. The number of the parameters is equivalent to choosing k input from $[m]$, namely C_m^k, which is exactly a $\frac{1}{m^k}$ fraction of parameter set $|p|$ and thus the hash function family $\{h_p\}: X\rightarrow Y$ is k-independent.

4. (7 points) The hash function family $h_p(x)$ of the previous part is a bit odd because it maps $[m]$ to itself. Consider instead a prime q that is smaller than m, and consider instead a new hash function family $h'_p(x)$ from $[m]$ to $[q]$ computed as: $h'_p(x)=h_p(x) \bmod q=(p_0+p_1x+p_2x_2+\ldots+p_{k-1}x^{k-1} \bmod m) \bmod q$—the hash function from the previous part, parameterized identically, but then taken modulo q. This new hash function family $h'_p(x)$ will *not* be k-independent, but in many cases it will be "close enough for practical purpose". (Note that, unlike for the previous part of this problem, the range of h' has size q instead of m.)

   Find bounds on the fraction of parameters p such that $h'_p(x_1)=y_1$ and $h'_p(x_2)=y_2$ and ... $h'_p(x_k)=y_k$, when $x_1, ..., x_k$ are distinct elements of $[m]$ and $y_1, ..., y_k$ are elements of $[q]$.

   Use these bounds and the approximation $e^x\approx 1+x$ for small x to show that (subject to this approximation), when q is smaller than m/k, then the fraction of p such that $h'_p(x_1)=y_1$ and $h'_p(x_2)=y_2$ and ... $h'_p(x_k)=y_k$ is within a factor of e of $\frac{1}{q^k}$. (Thus, the hash function family $h'_p(x)$ is "e-close to being k-independent".)