15-451 Assignment 03

Karan Sikka
ksikka@cmu.edu
Collaborated with Dave Cummings and Sandeep Rao.
Recitation: A
September 18, 2014

1: Balanced Splay Trees

The outline of the proof is as follows:

1. We show a lower bound on the decrease in potential when performing a deep splay, $3lg(n)+1$.
2. We show an upper bound on the increase in potential when performing a splay, $3lg(n)+1$.
3. From these facts, we conclude that since we start with a perfectly balanced tree (min potential), a deep splay can only occur at most once after a splay, due to restrictions on the potential function.

(1) Lower bound on decrease in potential during deep splay.

From the access lemma, we know that:

$$\text{a.c. of splay} \leq 3(r(y) - r(x)) + 1$$

Where y is the root node and x is the node being splayed.
We know that $r(y) = lg(n)$ and in the case of a deep splay, $r(x) = 0$ since it's terminal, subtree size is 1, $lg(1) = 0$.

We also know the definition of amortized cost in terms of the potential function. Then in the case of a deep splay,

$$\#\text{ of rots} + \Delta\Phi \leq 3lg(n) + 1$$

In the case of a deep splay, the number of rotations is $d - 1$ since a rotation moves the terminal node 1 level closer to the root.
And a deep splay only occurs when the tree is off-balance which means that $d \geq 6lg(n) + 3$. Combining these facts, we do the following algebra:

$d - 1 + \Delta\Phi \leq 3lg(n) + 1$
$\implies \Delta\Phi \leq 3lg(n) + 1 - (d - 1)$
$\implies \Delta\Phi \leq 3lg(n) + 1 - (6lg(n) + 3 - 1)$
$\implies \Delta\Phi \leq -(3lg(n) - 1)$

(2) Upper bound on increase in potential of a splay.

Combine the access lemma with the defn of amortized cost to establish the following truth:

$$\Delta\Phi \leq 3(r(y) - r(x)) + 1 - \#\text{ of rots}$$

We know that $r(y) = lg(n)$. We know $r(x) \geq 0$ and same with the number of rotations, so we can omit them from our upper bound. Thus, we obtain the following.

$$\Delta\Phi \leq 3lg(n) + 1$$

(3) We start with a perfectly balanced tree, so that the potential function is at its minimum. From there, a deep splay can only occur after a splay, and it can only occur once per splay, otherwise the potential function would go lower than it was when the tree was perfectly balanced.

2: Universal Hashing

(a.i.) No it is not universal.

Consider the case when $M < n$ and consider two strings which vary only in their M^{th} character.

When you compute the hash function for these two strings using any hash function in the family, you'll get identical hash values.

This is because the only differing term in the sums formed by the hash computations, representing the M^{th} character, will have a coefficient of M and $M \mod M$ is zero. So the different term can be eliminated and the hash functions are guaranteed to output identical values.

Thus the probability of collision (over all hash functions in the family) on these two keys is 1, and $1 > \frac{1}{M}$.

(a.ii.) It is not 2-universal, because if it were, then it would also be universal, but as we showed in part i, it's not.

(b.i.) Yes, the hash family is universal.

Say you have two strings which vary by exactly 1 character in the k^{th} position.

We will attempt to compute their hash functions without loss of generality with regards to the hash function chosen.

$h_1 = \sum_{i=1}^{n} T[i][S_{1,i}] \mod M$
$h_2 = \sum_{i=1}^{n} T[i][S_{2,i}] \mod M$

Notice that all terms of the form $T[i][S_{\_,i}]$ will be equal between S_1, S_2 because all characters are the same but one.
Encapsulate the sum of these terms into X, and now only the term where $i = k$ differs.

$h_1 = X + T[k][S_{1,k}] \mod M$
$h_2 = X + T[k][S_{2,k}] \mod M$

Furthermore, since any value outputted from table T is identical when modded by M (due to its range of values, we say the following:

$h_1 = X \mod M + T[k][S_{1,k}]$
$h_2 = X \mod M + T[k][S_{2,k}]$

Now, the probability that $h_1 = h_2$ is the same as the probability that $T[k][S_{1,k}] = T[k][S_{2,k}]$, and since the table is filled with values uniformly at random, the probability is $\frac{1}{M}$ because the RHS has 1 out of M equally likely outcomes that will be equal to the LHS.

This showed that if exactly one character varied between two strings, the probability that their hash would collide is equal to $\frac{1}{M}$.
If an additional character varied, the probability of a hash collision would only be less than this,

since there would only be additional ways for the hash functions to be inequal compared to the previous analysis.

Therefore, the probability of a hash collision between any two distinct keys is at most $\frac{1}{M}$, and the hash family is universal.

(b.ii.) The maximum p is the total size of the universe with is 256^n because we couldn't find a p where it's not p-universal.

The p-universal property holds for any fixed distinct p-length sequence of distinct keys.

The distinct property matters because it means that at least one character varies between two strings. This will cause the uniform random distribution of the hash function.

First, we show that the sum of uniformly randomly distributed variables between 0 and $M-1$ modded by M is still uniformly distributed between 0 and $M-1$.
The argument is inductive. It obviously holds for the case where you're summing 1 variable, since it's already uniform random and it's unaffected by mod due to its range.
Then, given a uniform random variable which is the M-modded sum of k uniform random variables, if you add a uniform random variable, the resulting value is still going to be uniform random after taking mod M.

So long as two strings have at least one differing character, the sum of the differing characters will form a uniform random distribution since the output of the T-table is uniform random. The hash functions outputs are always uniformly randomly distributed.