LSH used in Approximate Similarity Search Under Edit Distance Using Locatlity Sensitive Hashing

Define Hash functions

We define hash functions $h_\rho(x)$ using an underlying function ρ
So if $\rho_1 = \rho_2$ then $h_{\rho 1}(x) = h_{\rho 2}(x)$

Calculating the hashed string

$p = 1/8$
$p_a = \sqrt{p/1+p} = 1/3$
$p_r = \sqrt{p}/(\sqrt{1+p} - \sqrt{p}) = 1/2$

$(r_1, r_2) \leftarrow \rho_1(x_i, |s|)$

$r_1 \leq p_a \implies$
hash-insert, *append* $\perp$

$r_1 > p_a, r_2 \leq p_r \implies$
hash-replace, *append* $\perp$ & $i{+}{+}$

$r_1 > p_a, r_2 > p_r \implies$
hash-match, *append* x_i & $i++$

x_i	$\|s\|$	$\rho_1(x_i, \|s\|)$
a	0	(0.1, 0.7)
b	0	(0.6, 0.3)
c	0	(0.7, 0.6)
$	0	(0.1, 0.4)
a	1	(0.9, 0.6)
b	1	(0.8, 0.3)
c	1	(0.5, 0.9)
$	1	(0, 0.1)
a	2	(0.1, 0.7)
b	2	(0.8, 0.2)
c	2	(0.1, 0.9)
$	2	(0.1, 0.3)
a	3	(0.6, 0.8)
b	3	(0.9, 0.4)
c	3	(0.2, 0.8)
$	3	(0.8, 0.7)
a	4	(0.2, 0.3)
b	4	(0.1, 0.1)
c	4	(0.7, 0.4)
$	4	(0.9, 0.5)
a	5	(0.5, 0.6)
b	5	(0.1, 0.5)
c	5	(0.4, 0.6)
$	5	(0.6, 0)

Figure 1: hash function

Calculating the hashed string

x=abc

$x_1 = a,\ \ s = "", \rho_1(a, 0) = (0.1, 0.7)$

$\implies\ r_1 \leq p_a\ \implies$ **hash-insert**

$x_1 = a,\ \ s = "\perp", \rho_1(a, 1) = (0.9, 0.6)$

$\implies\ r_1 > p_a,\ \ r_2 > p_r\ \implies$ **hash-match**

$x_2 = b,\ \ s = "\perp a", \rho_1(b, 2) = (0.8, 0.2)$

$\implies\ r_1 > p_a,\ \ r_2 \leq p_r\ \implies$ **hash-replace**

$x_2 = b,\ \ s = "\perp a \perp", \rho_1(b, 3) = (0.9, 0.4)$

$\implies\ r_1 > p_a,\ \ r_2 \leq p_r\ \implies$ **hash-replace**

...

The final hash values

$$\begin{array}{ll} x = abc & h_{\rho 1}(x) = \perp a \perp \perp \perp \perp \\ y = bac & h_{\rho 1}(y) = \perp a \perp \perp \perp \perp \\ z = cba & h_{\rho 1}(z) = c \perp \perp a \end{array}$$

x & y have matching hash strings

Analysis of the hash values

x=abc

$x_{i(x,k,\rho)}$	$\|s\|$	$\tau_k(x,\rho)$
a	0	hash-insert
a	1	hash-match
b	2	hash-replace
c	3	hash-insert
c	4	hash-replace
\$	5	hash-replace

y=bac

$y_{i(y,k,\rho)}$	$\|s\|$	$\tau_k(y,\rho)$
b	0	hash-replace
a	1	hash-match
c	2	hash-insert
c	3	hash-insert
c	4	hash-replace
\$	5	hash-replace

Analysing Grid Walk

When $x_{i(x,k,\rho)} \neq y_{i(y,k,\rho)}$

$\tau_k(x,\rho)$	$\tau_k(y,\rho)$	$g_k(x,y,\rho)$
hash-replace	hash-replace	replace
hash-replace	hash-insert	delete
hash-insert	hash-replace	insert
hash-insert	hash-insert	loop
hash-match	-	stop
-	hash-match	stop

When $x_{i(x,k,\rho)} = y_{i(x,k,\rho)}$

$\tau_k(x,\rho)$	$\tau_k(y,\rho)$	$g_k(x,y,\rho)$
hash-match	hash-match	match
hash-replace	hash-replace	match
hash-insert	hash-insert	loop

Gridpath for x and y

x	$\tau_k(x, \rho 1)$	y	$\tau_k(y, \rho 1)$	$g_k(x, y, \rho 1)$
a	hash-insert	b	hash-replace	insert
a	hash-match	a	hash-match	match
b	hash-replace	c	hash-insert	delete
c	hash-insert	c	hash-insert	loop
c	hash-replace	c	hash-replace	match
$	hash-replace	$	hash-replace	match

Transformations

Greedily apply sequence of edits (remove match and loop) to x.
So for the above case, we would have insert(b) at a and delete at b.

Bounds of Collision Probabilities

Lemma 14: If x and y satisfy $ED(x, y) \leq r$, then
$Pr_\rho(h_\rho(x) = h_\rho(y)) \geq p^r - 2/n^2$.

Lemma 15: If x and y satisfy $ED(x, y) \geq cr$, then
$Pr_\rho(h_\rho(x) = h_\rho(y)) \leq (3p)^c r$.

This gives us $(r, cr, p^r - 2/n^2, (3p)^c r)$-sensitive family.