New Algorithms for Text Fingerprinting

Roman Kolpakov

Liapunov French-Russian Institute,

Moscow State University, Moscow, Russia

Mathieu Raffinot

CNRS, Poncelet Laboratory,

Independent University of Moscow, Moscow, Russia

Fingerprints

$\Sigma = \{a_1, \ldots, a_q\}$

$s = s_1 \ldots s_n \in \Sigma^*$

Def:

$C(s) = \{a_i \in \Sigma \mid \exists j \; a_i = s_j\} \subseteq \Sigma$ — *fingerprint* of s

$C_s(i,j) = C(s_i \ldots s_j), \quad s_i \ldots s_j$ — *location* of $C_s(i,j)$

$\mathcal{F}(s) = \{C \subseteq \Sigma \mid \exists i, j \; C = C_s(i,j)\}$

Maximal locations

$C \subseteq \Sigma$

Def: $s_i \ldots s_j$ — *maximal location* of $C = C_s(i,j)$ in s iff

1. if $i > 1$, $s_{i-1} \notin C$
2. if $j < n$, $s_{j+1} \notin C$

Ex: $ab\underline{acadc}$ — maximal location of fingerprint $\{a, c, d\}$

$\mathcal{L}(s)$ — set of all maximal locations in s

Prop: $|\mathcal{F}(s)| \leq |\mathcal{L}(s)| \leq n|\Sigma|$

$|\mathcal{L}(s)|$ can be asymptotically less than $n|\Sigma|$

Ex:

$\Sigma_k = \{a_1, a_2, \ldots, a_k\}, \quad k = 1, 2, \ldots$

$w_1 = a_1$

$w_k = w_{k-1} a_k w_{k-1} \in \Sigma_k^*$ for $k > 1$

$|w_k| \cdot |\Sigma_k| = k \cdot (2^k - 1)$

$\mathcal{L}(w_k) = 2^{k+1} - (k + 2) = o(|w_k| \cdot |\Sigma_k|)$

Our problems

1. Compute the set $\mathcal{F}(s)$
2. For a given $C \subseteq \Sigma$, find if $C \in \mathcal{F}(s)$
3. For a given $C \subseteq \Sigma$, find all maximal locations of C in s

Amir, Apostolico, Landau, Satta 2003:
Problem 1 can be solved in $O(n|\Sigma| \log |\Sigma| \log n)$ time
Problems 2 and 3 can be solved in $O(|\Sigma| \log n)$ time and $O(|\Sigma| \log n + K)$ time respectively (K — size of output)

Didier, Schmidt, Stoye, Tsur 2006:
Problem 1 can be solved in $O(\min\{n|\Sigma| \log |\Sigma|, n^2\})$ time

Our results

Problem 1 can be solved in $O((n + |\mathcal{L}(s)|) \log |\Sigma|)$ time

Problems 2 and 3 can be solved in $O(|\Sigma|)$ time and $O(|\Sigma| + K)$ time respectively (K — size of output)

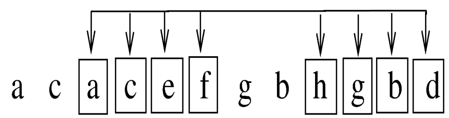
Naming technique

fingerprint arrays $\longrightarrow$ fingerprint names

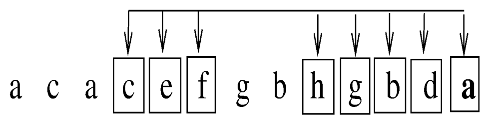
Ex: 10101100 $\longrightarrow$ [7]

<table>
<tr><td colspan="8">[7]</td></tr>
<tr><td colspan="4">[5]</td><td colspan="4">[6]</td></tr>
<tr><td colspan="2">[2]</td><td colspan="2">[2]</td><td colspan="2">[3]</td><td colspan="2">[4]</td></tr>
<tr><td>[1]</td><td>[0]</td><td>[1]</td><td>[0]</td><td>[1]</td><td>[1]</td><td>[0]</td><td>[0]</td></tr>
</table>

1	2	3	4	5	6	7	8	9	10	11	12	13
a	c	a	c	e	f	g	b	h	g	b	d	a



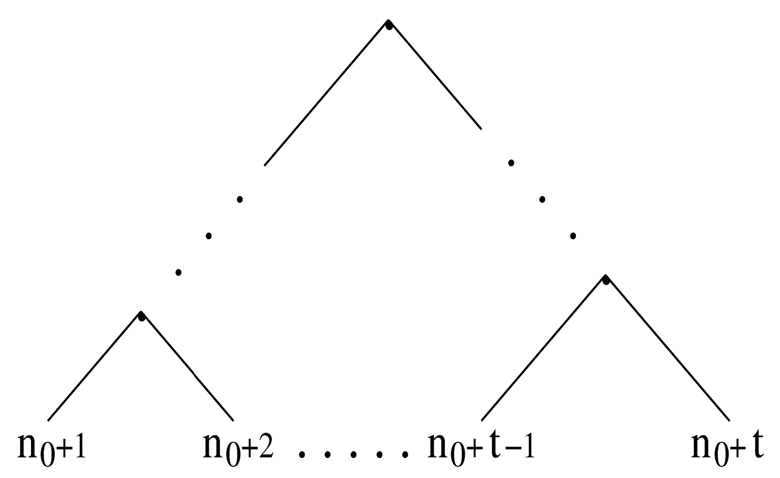
1	2	3	4	5	6	7	8	9	10	11	12	13
a	c	a	c	e	f	g	b	h	g	b	d	
c	a	c	e	f	g	b	h	g	b	d		
		e	f	g	b	h	g	b	d			
		f	g	b	h			d				
		g	b	h	d							
		b	h	d								
		h	d									
		d										



1	2	3	4	5	6	7	8	9	10	11	12	13
a	c	a	c	e	f	g	b	h	g	b	d	**a**
c	a	c	e	f	g	b	h	g	b	d	**a**	
		e	f	g	b	h	g	b	d	**a**		
		f	g	b	h			d	**a**			
		g	b	h	d			**a**				
		b	h	d	**a**							
		h	d	**a**								
		d	**a**									

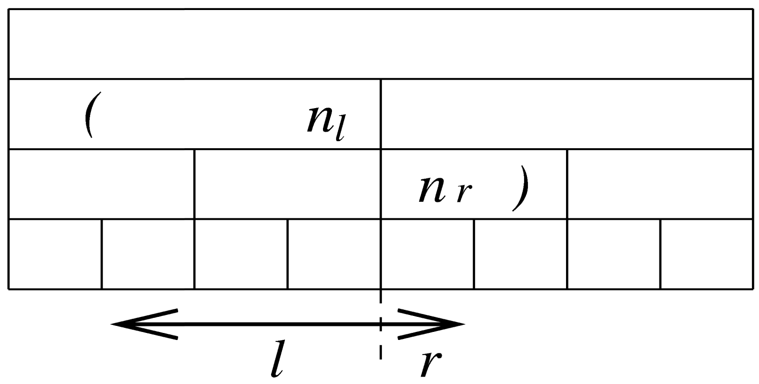
Fingerprint tree

$n_0 + 1, n_0 + 2, \ldots, n_0 + t$ — ordered fingerprint names of $\mathcal{F}(s)$



Edges are labeled by tuples $\{(n_l, n_r), l, r\}$

Tuple $\{(n_l, n_r), l, r\}$ points to the corresponding segment of length $l + r$ in fingerprint array:



Prop: Fingerprint tree can be constructed in $O(|\mathcal{F}(s)| \log |\Sigma|)$ time and $O(|\mathcal{F}(s)|)$ space.

The corresponding segment can be computed from $\{(n_l, n_r), l, r\}$ in $O(l + r)$ time

$$\Downarrow$$

Search of a given fingerprint array in fingerprint tree can be done in $O(|\Sigma|)$ time

$$\Downarrow$$

All maximal locations of a given fingerprint can be retrieved in $O(|\Sigma| + K)$ time (K — size of output)

Conclusions

- $O((n + |\mathcal{F}(s)|) \log |\Sigma|)$ time bound for Problem 1?
- computing fingerprints (common fingerprints) for sets of strings (in particular, regular languages)
- on-line computation of $\mathcal{F}(s)$