String Processing Algorithms 2015 - Week 2 Exercises

Rodion Efremov

October 31, 2015

Exercise 1

Outline algorithms that find the most frequent symbol in a give string.

(a) for ordered alphabet, and

(b) for integer alphabet.

The algorithms should be as fast as possible. What are their (worst case) time complexities? Consider also the case where $\sigma \gg n$.

Solution

Algorithm 1: MostFrequentSymbol(S)

let f **be an empty map** $f\colon \Sigma \to \mathbb{N}$
$\mu =$ **nil**
$L_\mu = 0$
for $i = 1$ **to** $|S|$ **do**
    if $S[i]$ **is not mapped in** f **then**
        $f(S[i]) = 1$
        if $L_\mu = 0$ **then**
            $L_\mu = 1$
            $\mu = S[i]$
    else
        $f(S[i]) = f(S[i]) + 1$
        if $L_\mu < f(S[i])$ **then**
            $L_\mu = f(S[i])$
            $\mu = S[i]$
return μ

Exercise 2

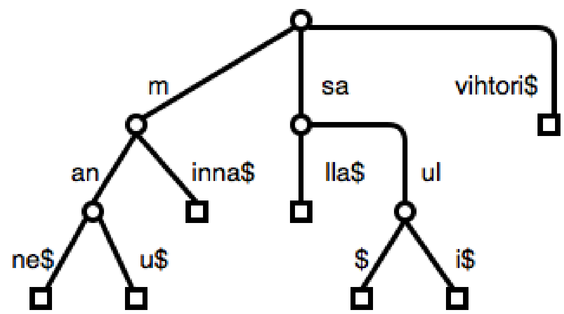
Let $\mathcal{R} = \{\texttt{manne}, \texttt{manu}, \texttt{minna}, \texttt{salla}, \texttt{saul}, \texttt{sauli}, \texttt{vihtori}\}$.

(a) Give the compact trie of $\mathcal{R}$.

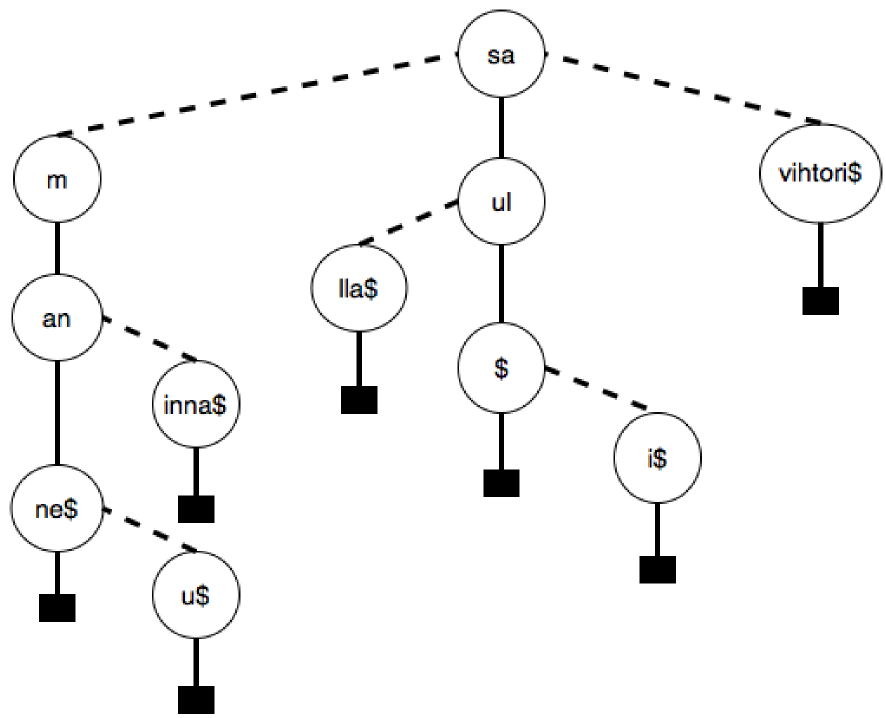
(b) Give the balanced compact ternary trie of $\mathcal{R}$.

Solution

(a)



(b)



Exercise 3

Exercise 4

Prove

(a) Lemma 1.14: For $i \in [2..n]$, $LCP_{\mathcal{R}}[i] = lcp(S_i, \{S_1, \ldots, S_{i-1}\})$.

(b) Lemma 1.15: $\Sigma LCP(\mathcal{R}) \leq \Sigma lcp(\mathcal{R}) \leq 2 \cdot \Sigma LCP(\mathcal{R})$.

Solution

(a)

We need an auxiliary lemma first:

Lemma 1 (Distance lemma)**.** *If $S_1 < S_2 < S_3$, $lcp(S_1, S_3) \leq lcp(S_2, S_3)$.*

Proof. The proof is by contradiction: Let $l_{13} = lcp(S_1, S_3)$ and $l_{23} = lcp(S_2, S_3)$. Assume the opposite that $l_{13} > l_{23}$. Now

$$S_1 = a_1 a_2 \ldots a_{l_{23}} \ldots a_{l_{13}} b_1 b_2 \ldots,$$
$$S_2 = a_1 a_2 \ldots a_{l_{23}} c_1 c_2 \ldots,$$
$$S_3 = a_1 a_2 \ldots a_{l_{23}} \ldots a_{l_{13}} d_1 d_2 \ldots.$$

Since $c_1 > a_{l_{23}+1}$, we must have also $S_2 > S_3$, which is a contradiction. □

Example:

$$S_1 : aaaab$$
$$S_2 : aaaba$$
$$S_3 : aaabc$$

Now assume that $i \in [2...n]$. We have that

$$lcp(S_i, \{S_1, \ldots, S_{i-1}\}) = \max\{lcp(S_i, S_{i-1}), lcp(S_i, \{S_1, \ldots, S_{i-2}\})\}.$$

Because by distance lemma for any $j = 1, 2, \ldots, i-2$, $lcp(S_j, S_i)$ cannot exceed $lcp(S_i, S_{i-1})$, we must have that

$$lcp(S_i, \{S_1, \ldots, S_{i-1}\}) = lcp(S_i, S_{i-1}) = LCP_{\mathcal{R}}[i],$$

as expected.

(b)

$$
\begin{aligned}
\Sigma lcp(\mathcal{R}) &= \sum_{S \in \mathcal{R}} lcp(S, \mathcal{R} \setminus \{S\}) \\
&\le \sum_{i \in [1..n-1]} lcp(S_i, S_{i+1}) + \sum_{i \in [2..n]} lcp(S_{i-1}, S_i) && \text{(by distance lemma)} \\
&= \sum_{i \in [2..n]} lcp(S_{i-1}, S_i) + \sum_{i \in [2..n]} lcp(S_{i-1}, S_i) \\
&= 2 \sum_{i \in [2..n]} lcp(S_{i-1}, S_i) \\
&= 2 \sum_{i \in [2..n]} LCP_{\mathcal{R}}[i] \\
&= 2 \sum_{i \in [1..n]} LCP_{\mathcal{R}}[i] && \text{(since } LCP_{\mathcal{R}}[1] = 0\text{)} \\
&= 2 \cdot \Sigma LCP(\mathcal{R}).
\end{aligned}
$$

What comes to the lower bound of $\Sigma lcp(\mathcal{R})$, we have

$$
\begin{aligned}
\Sigma lcp(\mathcal{R}) &= lcp(S_1, S_2) + lcp(S_{n-1}, S_n) \\
&\quad + \sum_{i \in [2..n-1]} \max\{lcp(S_i, S_{i-1}), lcp(S_i, S_{i+1})\} && \text{(by distance lemma)} \\
&\ge \sum_{i \in [2..n]} lcp(S_{i-1}, S_i) \\
&= \sum_{i \in [2..n]} LCP_{\mathcal{R}}[i] \\
&= \sum_{i \in [1..n]} LCP_{\mathcal{R}}[i] && \text{(since } LCP_{\mathcal{R}}[1] = 0\text{)} \\
&= \Sigma LCP(\mathcal{R}),
\end{aligned}
$$

which concludes the proof.

Example:

$\mathcal{R}$	$LCP_{\mathcal{R}}$	$lcp(S, \mathcal{R})$
aaaa	0	3
aaab	3	3
abba	1	1
baab	0	0
$\sum$	4	7