Text Indexing: Lecture 5

Simon Gog – gog@ira.uka.de
The LCP-Interval Tree

Definition of an LCP-interval ([1])
An interval \([i, j]\), where \(0 \leq i \leq n - 1\) is called LCP-interval of LCP value \(\ell\) (denoted by \(\ell - [i, j]\)) if
- \(LCP[i] < \ell\) or \(i = 0\)
- \(LCP[k] \geq \ell\) for all \(k \in [i + 1, j]\)
- \(LCP[k] = \ell\) for at least one \(k \in [i + 1, j]\)
- \(LCP[j + 1] < \ell\)

Every index \(k\) with \(i < k \leq j\) and \(LCP[k] = \ell\) is called \(\ell\)-index. There are at most \(\sigma - 1\) \(\ell\)-indices in an LCP-interval.

Note: Each LCP-interval corresponds to a node in the suffix tree.
### The LCP-Interval Tree – Example

<table>
<thead>
<tr>
<th>$i$</th>
<th>SA</th>
<th>$LCP$</th>
<th>$\mathcal{T}[SA[i], n-1]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11</td>
<td>0</td>
<td>$$</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0</td>
<td>i$</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>1</td>
<td>iippi$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>issippi$</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>ississippi$</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>mississippi$</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>0</td>
<td>pi$</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>1</td>
<td>ppi$</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>0</td>
<td>sippi$</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>2</td>
<td>sissippi$</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>1</td>
<td>sippi$</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>3</td>
<td>ssissippi$</td>
</tr>
</tbody>
</table>

- Singleton intervals $\ell - [i, i]$ are omitted
Properties of the LCP-Interval Tree [2]

Overlapping

Two lcp-intervals $\ell - [i, j] \neq m - [p, q]$ cannot overlap, i.e. one of the following cases must hold:

- $[i, j]$ is a subinterval of $[p, q]$, i.e. $p \leq i < j \leq q$.
- $[p, q]$ is a subinterval of $[i, j]$, i.e. $i \leq p < q \leq j$.
- $[i, j]$ and $[p, q]$ are disjoint, i.e. $j < p$ or $q < i$.

Proof: Blackboard
Child interval
An $m$-interval $m\-[p, q]$ is said to be embedded in an $\ell$-interval $[i, j]$ if it is a subinterval of $[i, j]$ and $m > \ell$. The $\ell$-interval is then called the interval enclosing $[p, q]$. If $[i, j]$ encloses $[p, q]$ and there is no other interval embedded in $[i, j]$ that also encloses $[p, q]$, then $[p, q]$ is called a child interval of $[i, j]$ (vice versa, $[i, j]$ is called parent interval of $[p, q]$).

Navigation: child operation
Let $[i, j]$ be an $\ell$-interval. If $i_1 < i_2 < \ldots < i_k$ are the $\ell$-indices in ascending order, then the child intervals of $[i..j]$ are $[i, i_1 - 1], [i_1, i_2 - 1], \ldots [i_k, j]$. 
Properties of the LCP-Interval Tree [2]

Navigation: parent operation

Previous/Next Smaller Value Queries
Let $A$ be an array of length $n$. For $i \in [1, n - 1]$ the previous smaller value function is defined as:

$$ psv(i, A) = \max\{j \mid 0 \leq j < i \land A[j] < A[i]\} $$

Analogously we define the next smaller value function:

$$ nsv(i, A) = \min\{j \mid i < j < n \land A[j] < A[i]\} $$

We omit $A$ in $psv/nsv$ if it is clear from the context.
Exercise

With $n \log n$ bits of space the psv or nsv function can be precomputed and answered in constant time. Devise an linear time algorithm to compute the table.
Properties of the LCP-Interval Tree [2]

Navigation: parent operation

Let $0 < k < n$ and $LCP(k) = \ell$. Then $[psv(k), nsv(k) - 1]$ is an lcp-interval of LCP-value $\ell$.

Proof:

- $LCP[psv(k)] < \ell$ (by definition of $psv(k)$)
- $LCP[m] \geq \ell$ for all $m \in [psv(k) + 1, nsv(k) - 1]$
- $LCP[k] = \ell$ (note that $psv(k) + 1 \leq k \leq nsv(k) - 1$)
- $LCP[nsv(k)] < \ell$ (by definition of $nsv(k)$)
Let \([i, j] \neq [0, n - 1]\) be an lcp-interval with \(LCP[i] = p\) and 
\(LCP[j + 1] = q\)

- case \(p = q\): \(p-[psv(i), nsv(j) - 1]\) is parent of \([i, j]\)
- case \(p > q\): \(p-[psv(i), j]\) is parent of \([i, j]\)
- case \(p < q\): \(q-[i, nsv(i)]\) is parent of \([i, j]\)
Bottom-Up Traversal of Interval Tree

00 \textit{lastInterval} \leftarrow \bot
01 \textit{s.push}(\langle 0, 0, \bot, [] \rangle) \ // \ lcp, \ left \ bound, \ right \ bound, \ children
02 \textbf{for} \ k \leftarrow 1 \ \textbf{to} \ n \ \textbf{do}
03 \ \quad lb \leftarrow k - 1
04 \ \quad \textbf{while} \ \textit{LCP}[k] < \textit{s.top()}.lcp \ \textbf{do}
05 \ \quad \textit{s.top()}.rb \leftarrow k - 1
06 \ \quad \textit{lastInterval} \leftarrow \textit{pop()}
07 \ \quad \textbf{process}(\textit{lastInterval})
08 \ \quad lb \leftarrow \textit{lastInterval}.lb
09 \ \quad \textbf{if} \ \textit{LCP}[k] \leq \textit{s.top()}.lcp \ \textbf{then}
10 \ \quad \textit{s.top()}.children.append(\textit{lastInterval})
11 \ \quad \textit{lastInterval} \leftarrow \bot
12 \ \textbf{if} \ \textit{LCP}[k] > \textit{s.top()}.lcp \ \textbf{then}
13 \ \quad \textbf{if} \ \textit{lastInterval} \neq \bot \ \textbf{then}
14 \ \quad \textit{s.push}(\langle \textit{LCP}[k], lb, \bot, [\textit{lastInterval}] \rangle)
15 \ \quad \textit{lastInterval} \leftarrow \bot
16 \ \textbf{else} \ \textit{s.push}(\langle \textit{LCP}[k], lb, \bot, [] \rangle)
Bottom-Up traversal of Interval Tree

Application

Problem
Find all substrings having at least \( p \) and at most \( q \) occurrences in \( \mathcal{T} \), where \( 1 < p \leq q \).

Solution

00 \textbf{process} (lastInterval)
01 \textbf{foreach} \langle \ell, i, j, [] \rangle \textbf{in} lastInterval.children \textbf{do}
02 \quad \textbf{if} \ p \leq (j - i + 1) \ \textbf{and} \ (j - i + 1) \leq q \ \textbf{then}
03 \quad \textbf{output} (lastInterval.lcp, \ell, [i..j])
04 \quad lastInterval.children \leftarrow []