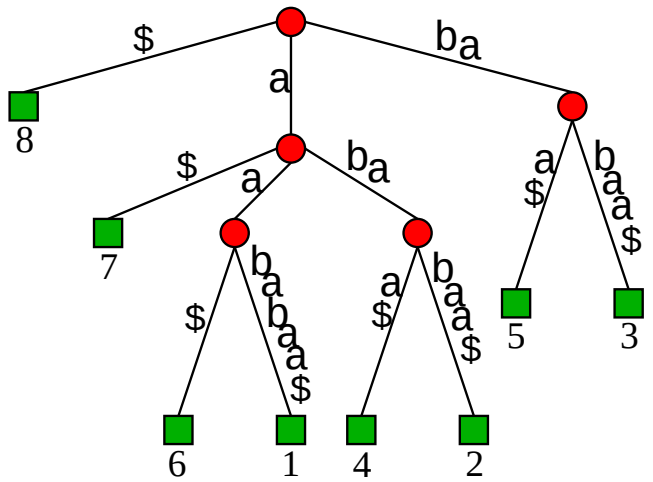
An RMQ-Based Variant of Compressed Suffix Trees

Johannes Fischer

formerly:
Ludwig-Maximilians-Universität München
Institut für Informatik
Algorithmische Bioinformatik

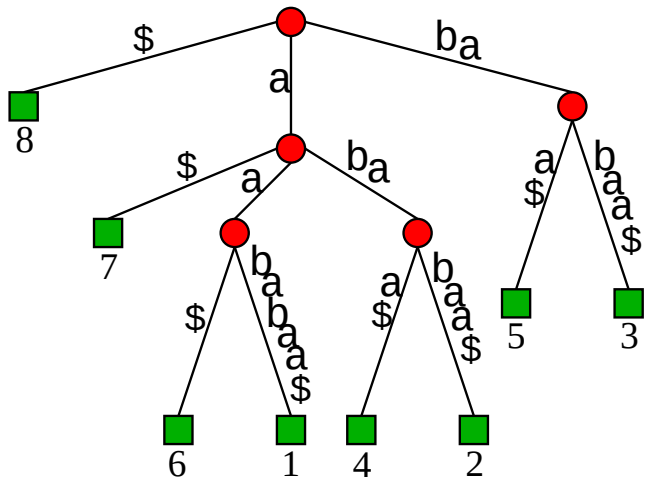
1. November 2007

Different Representations of Suffix Trees



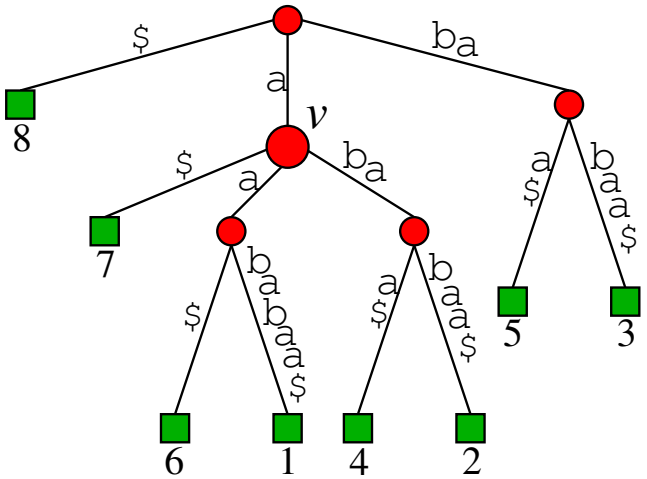
CST [Sadakane'07]	**ESA** [Abouelhoda et al'04]
	(compressed) suffix array: $\|CSA\|$ bits
	LCP-array: $2n + o(n)$
BPS: $4n$	—
navigation: $o(n)$	child-table: $5n + o(n)$
Ψ	suflink: $2n \log n$
LCA: $o(n)$	[no support]
$\|\mathbf{CSA}\| + \mathbf{6n} + \mathbf{o(n)}$	$\|\mathbf{CSA}\| + \mathbf{O(n \log n)}$

Different Representations of Suffix Trees



CST [Sadakane'07]	**ESA** [Abouelhoda et al'04]	**RMQ-CST**
(compressed) suffix array: $\|CSA\|$ bits		
	LCP-array: $2n + o(n)$	
BPS: $4n$	—	—
navigation: $o(n)$	child-table: $5n + o(n)$	RMQ: $2n + o(n)$
Ψ	suflink: $2n \log n$	Ψ
LCA: $o(n)$	[no support]	RMQ
$\|\mathbf{CSA}\| + \mathbf{6n} + \mathbf{o(n)}$	$\|\mathbf{CSA}\| + \mathbf{O(n \log n)}$	$\|\mathbf{CSA}\| + \mathbf{4n} + \mathbf{o(n)}$

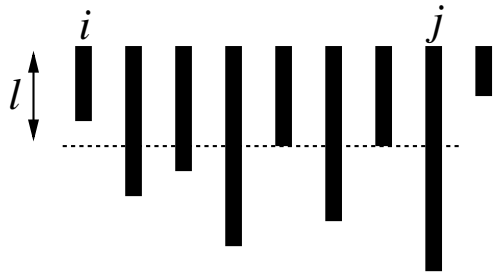
Representing Nodes



((()(()(()())(()()))(()()))) LCP=−1 0 1 2 1 3 0 2

v v

- CST: pair of matching "(...)"
- ESA: LCP-interval $\ell - [i : j]$
  - $\mathrm{LCP}[i] < \ell$ and $\mathrm{LCP}[j+1] < \ell$
  - $\mathrm{LCP}[k] \geq \ell$ for all $i < k \leq j$
  - $\exists k \in [i+1 : j] : \mathrm{LCP}[k] = \ell$



Support of Operations

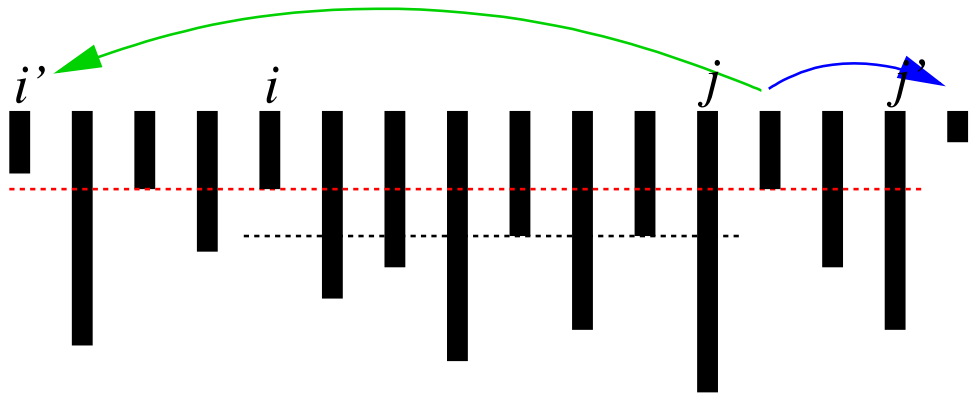
	CST [Sadakane'07]	**RMQ-CST**
root()	$O(1)$	$O(1)$
isleaf(v)	$O(1)$	$O(1)$
child(v, a)	$O(\log \lvert\Sigma\rvert \cdot t_{\mathrm{SA}})$	$O(\log \lvert\Sigma\rvert \cdot t_{\mathrm{SA}})$
firstchild(v)	$O(1)$	$O(t_{\mathrm{SA}})$
sibling(v)	$O(1)$	$O(t_{\mathrm{SA}} \cdot \log\log n)$
parent(v)	$O(1)$	$O(t_{\mathrm{SA}} \cdot \log\log n)$
edge(v, d)	$O(t_{\mathrm{SA}})$	$O(t_{\mathrm{SA}})$
depth(v)	$O(t_{\mathrm{SA}})$	$O(1)$
lca(v, w)	$O(1)$	$O(t_{\mathrm{SA}} \cdot \log\log n)$
suflink(v)	$O(t_{\psi})$	$O((t_{\mathrm{SA}} + t_{\psi}) \cdot \log\log n)$

Finding the Parent

- $\ell - [i : j]$ is current interval
- **task:** find parent-interval $\ell' - [i' : j']$
- w.l.o.g.: $\text{LCP}[i] \leq \text{LCP}[j+1] \rightsquigarrow \ell' = \text{LCP}[j+1]$

Finding the Parent

- $\ell - [i : j]$ is current interval
- **task:** find parent-interval $\ell' - [i' : j']$
- w.l.o.g.: $\text{LCP}[i] \leq \text{LCP}[j+1] \rightsquigarrow \ell' = \text{LCP}[j+1]$
- i' can be found by: $\max\{k < j+1 : \text{LCP}[k] < \text{LCP}[j+1]\}$
- ...and $j' = \min\{k > j+1 : \text{LCP}[k] < \text{LCP}[j+1]\} - 1$

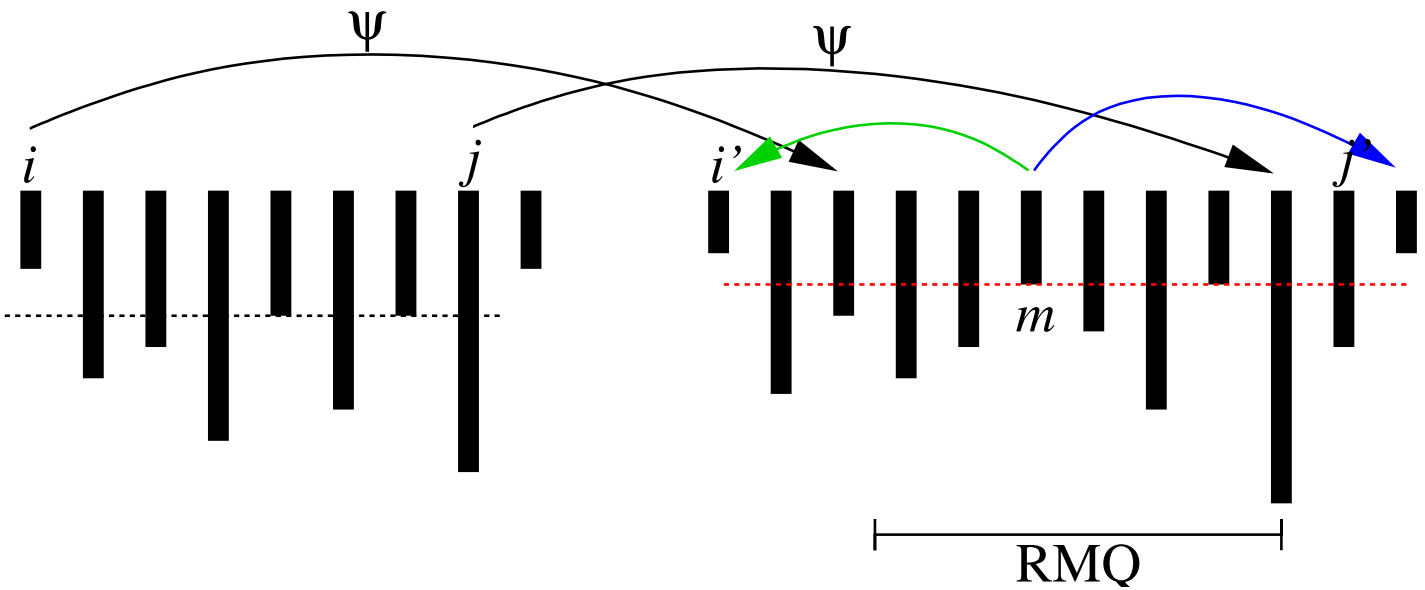


Finding the Suffix Link

- $\ell - [i : j]$ is current interval
- **task:** find suffix-link-interval $\ell' - [i' : j']$ $(\ell' = \ell - 1)$

Finding the Suffix Link

- $\ell - [i : j]$ is current interval
- **task:** find suffix-link-interval $\ell' - [i' : j']$ ($\ell' = \ell - 1$)
- first find longest proper suffix of $T_{SA[i]..n}$ by following $\Psi(i)$
- ...and likewise for $\Psi(j)$
- find ℓ'-index m by a range minimum query
- i' can be found by: $\max\{k < m : \text{LCP}[k] < \text{LCP}[j+1]\}$
- ...and $j' = \min\{k > m : \text{LCP}[k] < \text{LCP}[j+1]\} - 1$

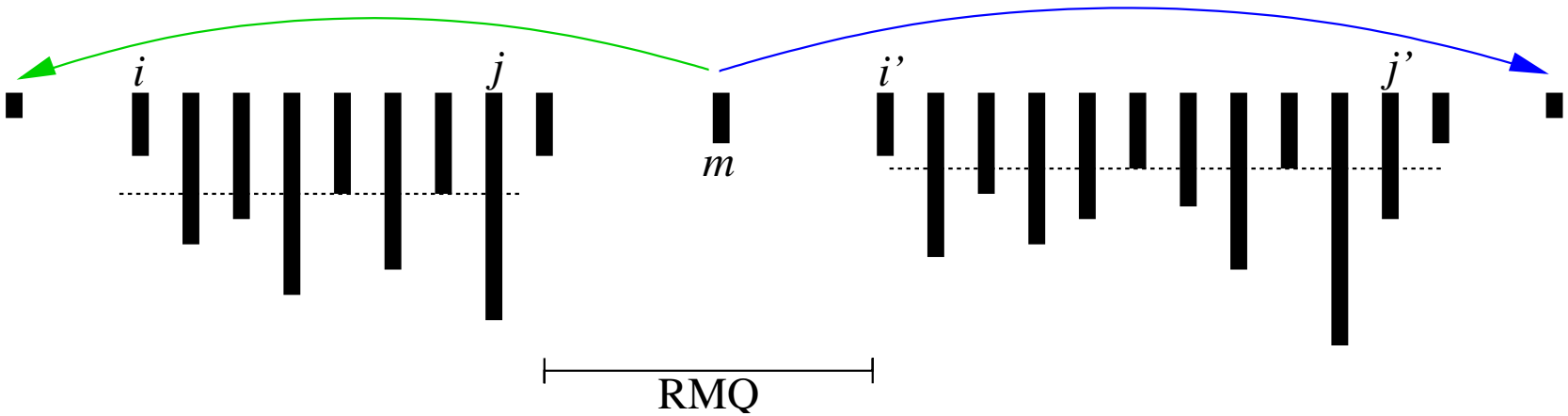


Finding the Lowest Common Ancestor

- $\ell - [i : j]$ and $\ell' - [i' : j']$ are two intervals
- **task:** find their LCA-interval $\ell'' - [i'' : j'']$

Finding the Lowest Common Ancestor

- $\ell - [i : j]$ and $\ell' - [i' : j']$ are two intervals
- **task:** find their LCA-interval $\ell'' - [i'' : j'']$
- let $m = \text{RMQ}(j + 1, i') \rightsquigarrow \ell'' = \text{LCP}[m]$
- i'' can be found by: $\max\{k < m : \text{LCP}[k] < \text{LCP}[j + 1]\}$
- ...and $j'' = \min\{k > m : \text{LCP}[k] < \text{LCP}[j + 1]\} - 1$



Finding Nearest Smaller Values (NSVs)

- Basic operations:
  - $i' = \max\{k < m : \mathrm{LCP}[k] < \mathrm{LCP}[j+1]\}$
  - $j' = \min\{k > m : \mathrm{LCP}[k] < \mathrm{LCP}[j+1]\}$
- $o(n)$ data structure for "nearest smaller value-queries"?
- resembles "findclose"-operation in BPSs [Munro-Raman'01]

Finding Nearest Smaller Values (NSVs)

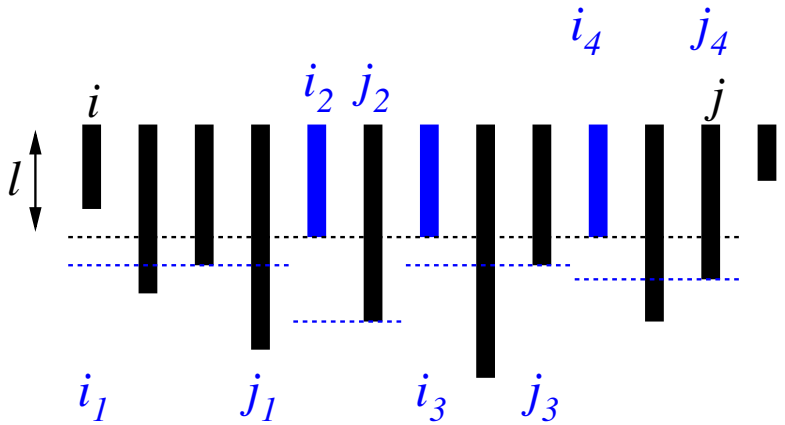
- Basic operations:
  - $i' = \max\{k < m : \mathrm{LCP}[k] < \mathrm{LCP}[j+1]\}$
  - $j' = \min\{k > m : \mathrm{LCP}[k] < \mathrm{LCP}[j+1]\}$
- $o(n)$ data structure for "nearest smaller value-queries"?
- resembles "findclose"-operation in BPSs [Munro-Raman'01]
- same pioneer-concept possible here:
  - divide LCP-array into *blocks*
  - label certain elements pioneers
  - store NSVs for pioneers
  - from these, find NSV-target-blocks of other elements
- scan target block $\rightsquigarrow$ $O(\log n)$ penalty
- with Cartesian Trees: binary search $\rightsquigarrow$ $O(\log \log n)$ penalty

Support of Operations

	CST [Sadakane'07]	**RMQ-CST**
root()	$O(1)$	$O(1)$
isleaf(v)	$O(1)$	$O(1)$
child(v, a)	$O(\log \lvert\Sigma\rvert \cdot t_{\mathrm{SA}})$	$O(\log \lvert\Sigma\rvert \cdot t_{\mathrm{SA}})$
firstchild(v)	$O(1)$	$O(t_{\mathrm{SA}})$
sibling(v)	$O(1)$	$O(t_{\mathrm{SA}} \cdot \log\log n)$
parent(v)	$O(1)$	$O(t_{\mathrm{SA}} \cdot \log\log n)$
edge(v, d)	$O(t_{\mathrm{SA}})$	$O(t_{\mathrm{SA}})$
depth(v)	$O(t_{\mathrm{SA}})$	$O(1)$
lca(v, w)	$O(1)$	$O(t_{\mathrm{SA}} \cdot \log\log n)$
suflink(v)	$O(t_{\psi})$	$O((t_{\mathrm{SA}} + t_{\psi}) \cdot \log\log n)$

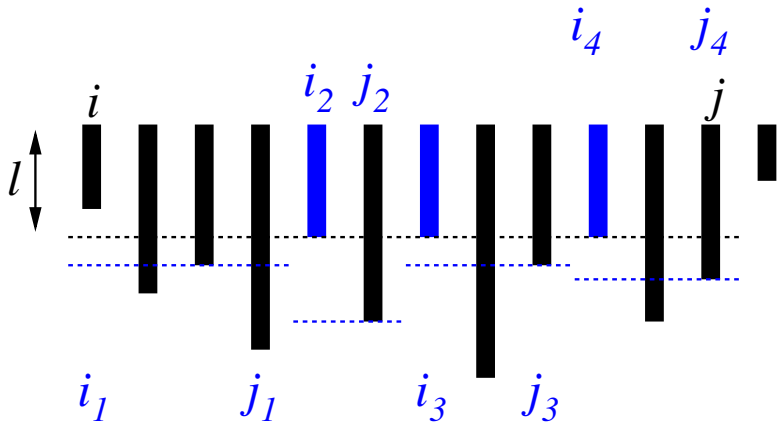
Finding Correct Child-Interval

- $\ell - [i : j]$ current interval
- **task:** given $a \in \Sigma$, find sub-interval of $\ell - [i : j]$ "representing" $T_{\mathrm{SA}[i]\ldots\mathrm{SA}[i]+\ell-1}a$
- the minima in $[l : r]$ bound the child-intervals $\rightsquigarrow$ RMQs!



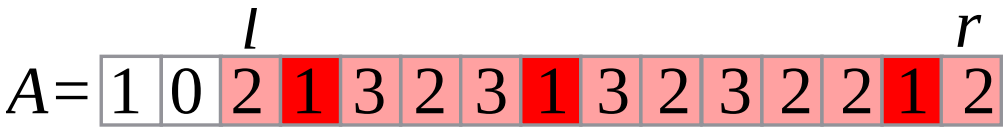
Finding Correct Child-Interval

- $\ell - [i : j]$ current interval
- **task:** given $a \in \Sigma$, find sub-interval of $\ell - [i : j]$ "representing" $T_{\mathsf{SA}[i]\ldots\mathsf{SA}[i]+\ell-1}a$
- the minima in $[l : r]$ bound the child-intervals $\rightsquigarrow$ RMQs!

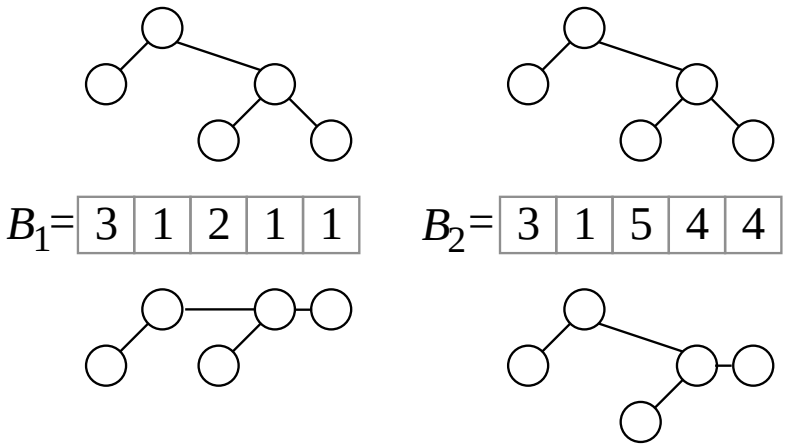


- Stepping through minima gives $O(|\Sigma| \cdot t_{\mathsf{SA}})$ time...
- ...but we can do in $O(\log |\Sigma| \cdot t_{\mathsf{SA}})$ time

Range Median of Minima Queries



- Minimum in $[l : r]$ not unique?
  - usually: $\mathrm{RMQ}(l, r)$ returns position of arbitrary min
  - here: $\mathrm{RMQ}^{\mathrm{med}}(l, r)$ returns median-position (same number of mins to the left and right)
- several new twists. . .
- Cartesian Trees not suitable
  $\rightsquigarrow$ extension to Super-Cartesian Trees



Hypothetical Support of Operations

If we could find NSVs in $O(1)$ time (using $o(n)$ space):

	CST [Sadakane'07]	**RMQ-CST**
root()	$O(1)$	$O(1)$
isleaf(v)	$O(1)$	$O(1)$
child(v, a)	$O(\log \lvert\Sigma\rvert \cdot t_{\mathrm{SA}})$	$O(\log \lvert\Sigma\rvert \cdot t_{\mathrm{SA}})$
firstchild(v)	$O(1)$	$O(t_{\mathrm{SA}})$
sibling(v)	$O(1)$	$O(t_{\mathrm{SA}})$
parent(v)	$O(1)$	$O(t_{\mathrm{SA}})$
edge(v, d)	$O(t_{\mathrm{SA}})$	$O(t_{\mathrm{SA}})$
depth(v)	$O(t_{\mathrm{SA}})$	$O(1)$
lca(v, w)	$O(1)$	$O(t_{\mathrm{SA}})$
suflink(v)	$O(t_{\psi})$	$O(t_{\mathrm{SA}} + t_{\psi})$