

The Wavelet Trie: Maintaining an Indexed Sequence of Strings in Compressed Space

CSI 5335 Paper presentation

Roberto Grossi, Giuseppe Ottaviano

presented by: Petr Praus

April 26th, 2012

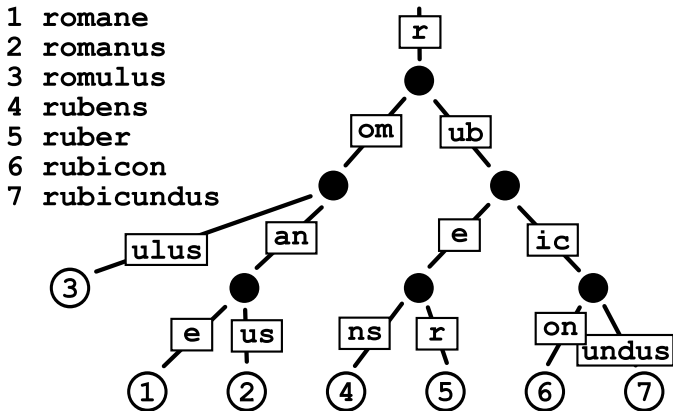
- A lot of things are string sequences.
- Column databases store and index string sequences.
- Great example: access logs

```
188.26.52.117 - - [24/Apr/2013:03:35:48 -0500] "GET /img/welcome/corner.png
188.26.52.117 - - [24/Apr/2013:03:35:49 -0500] "GET /img/welcome/arrowDown.gif
188.26.52.117 - - [24/Apr/2013:03:35:49 -0500] "GET /img/welcome/regionals.jpg
```

- Pretty similar, huh?
- I heard indexes make stuff faster → **indexed sequence of strings**
- Rank query: Number of requests for `/img/welcome/corner.png`?
- Select query: Position of i -th occurrence of `/img/welcome/corner.png`
- We can do prefix operations too

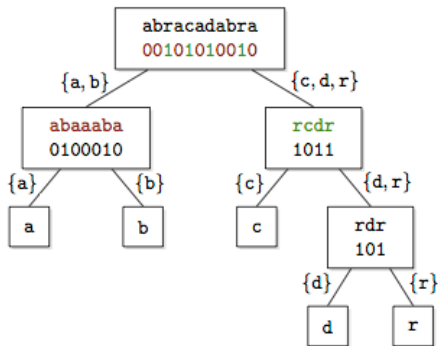
Patricia Trie

- Space-efficient trie.
- Node has always at least two children.



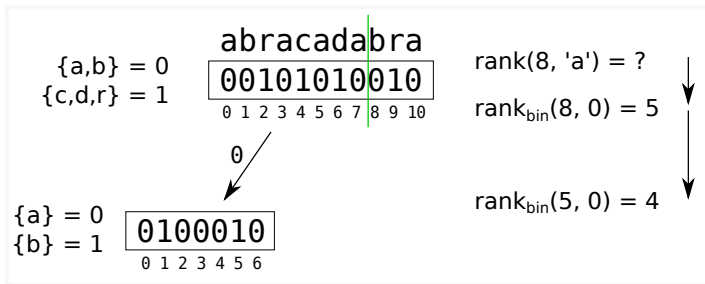
Wavelet Tree

- Organizes a string into a balanced binary tree of bit vectors.
- Alphabet $\Sigma = \{a, b, c, d, r\} \rightarrow \{0, 0, 1, 1, 1\}$
- At root, we have *ambiguity*, reducing ambiguity towards leaves



Efficient Computation of Rank in Wavelet Tree

- $\text{rank}(8, a) =$ how many a 's before position 8
- $\text{rank}_{\text{bin}}(\text{pos}, s)$ binary rank, # of occurrences of s before pos



E.g.: # of requests to `/img/welcome/corner.png` before April 10th.

Mutable & Compressed Indexed Sequences

- Sequences can change over time:
Insert(s, pos), *Append(s)*, *Delete(pos)*
- Alphabet not always known in advance
- Traditional approach, store explicitly (e.g. array), make an index
- Space-inefficient, we want to query the compressed representation
- **Succinct data structure** – uses space close to lower information-theoretic bound

Wavelet Trie to rescue

- Wavelet Tree + Patricia Trie
- Compressed data structure
- Can support *Insert*, *Append*, *Delete*, and dynamic alphabet
- Static, Append-only, Fully-dynamic
-

Thank you.

- <http://alexbowe.com/wavelet-trees/>