

Brief Announcement: On Local Representation of Distances in Trees

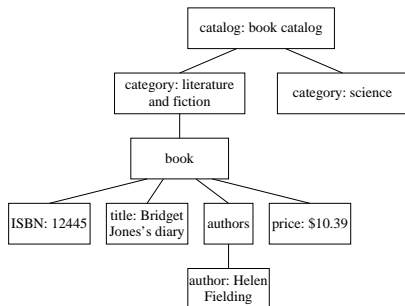
C. Gavoille A. Labourel

Laboratoire Bordelais de Recherche Informatique (LaBRI)
Université Bordeaux

PODC 2007, 12-15 August

XML document

XML is a generic framework for storing any amount of text or any data whose structure can be represented as a tree.



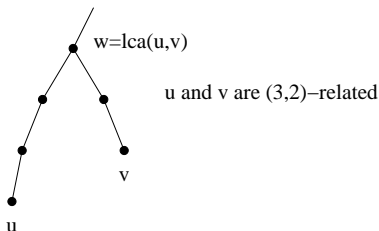
XML search and preprocessing

- ▶ We need a data structure that supports efficiently structural queries such as finding all book items of a specified author and a specified price.
- ▶ We add to the XML documents a big hash table containing all words and tags (like book or price) of the documents.
- ▶ We give to each node of the XML documents a unique label.
- ▶ We associate with each entry in the hash table the labels of all nodes containing the corresponding word or tag.
- ▶ By giving to the nodes suitable labels that permit ancestor queries, computing structural queries can be done by retrieving labels of relevant nodes and select those having the appropriate ancestor relationships.

k -relationship scheme

Definition

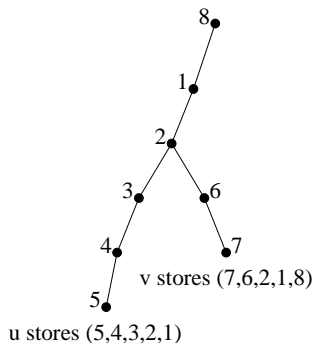
Two nodes u and v are (k_1, k_2) -related if u is at distance k_1 of $w = \text{lca}(u, v)$ and v is at distance k_2 of w .



A k -relationship scheme consists in assigning labels to nodes of a tree such that we can decide if two nodes are (k_1, k_2) -related ($k_1, k_2 \leq k$) only by examining their labels.

Trivial solution

A trivial solution is to store for each node v identifiers of v and its k closest ancestors.



Related works

Theorem (Astrup, Bille, and Rauhe 2003)

- ▶ *A k -relationship scheme can be achieved with $\log n + O(k^2 \log(k \log n))$ bits per node.*
- ▶ *$\log n + \log \log n + \Omega(1)$ bits per node are needed for a 1-relationship scheme (siblings).*

Our result

Theorem

A *k*-relationship scheme can be achieved with $\log n + O(k \log(k \log(\frac{n}{k})))$ bits per node.

$$\begin{array}{ll} \text{2nd order term} & : \quad k^2 \log(k \log n) \quad \rightarrow \quad k \log(k \log(\frac{n}{k})) \\ \text{for } k = 1 & : \quad 5 \log \log n \quad \rightarrow \quad 2 \log \log n \end{array}$$

- ▶ The principle of our scheme is the same as the trivial solution: store for each node v identifiers of v and its k -closest ancestors.
- ▶ The difficulty is to find identifiers such that we can encode the k -closest ancestors of a node in a very efficient way.
- ▶ We achieve this with a special decomposition of the tree using separators.

Concluding remarks and open problems

Our scheme implies that distances in trees can be computed with labels of $\log n + o(\log n)$ bits if the distance is $o(\log n / \log \log n)$. One may ask if it is possible to:

- ▶ Design a distance labeling scheme for trees with $\log n + o(\log n)$ bit labels and for larger distances, say for distances up to $\log n$.
- ▶ Design a distance labeling scheme for small distances for bounded treewidth graphs.