

# Brief Annoucement: On Local Representation of Distances in Trees <sup>\*</sup>

Cyril Gavoille  
gavoille@labri.fr

Arnaud Labourel  
labourel@labri.fr

LaBRI - Université Bordeaux  
France

## ABSTRACT

We consider distributed representation scheme for trees, supporting some special relationships between nodes at small distance. For instance, we show that for a tree  $T$  and an integer  $k$  we can assign local information on nodes such that we can decide for two nodes  $u$  and  $v$  if the distance between  $u$  and  $v$  is at most  $k$  and if so, compute it only using the local information assigned. For trees with  $n$  nodes, the local information assigned by our scheme is binary label of  $\log n + O(k \log(k \log(n/k)))$  bits, improving a recent result of Alstrup, Bille and Rauhe [1].

## Categories and Subject Descriptors

E.1 [Data Structure]: distributed data structures, graphs and networks

## General Terms

Algorithms

## Keywords

distributed representation, trees, distance

## 1. INTRODUCTION

A *distributed representation scheme* is a scheme maintaining global information on a network using local data structures (or labels) assigned to nodes of the network. Such schemes play an important role in the fields of distributed computing. Their goal is to locally store some useful information about the network and make it conveniently accessible. For instance, *implicit representation* or *adjacency scheme* of networks is a distributed representation scheme that support adjacency queries, i.e., adjacency between two nodes can be determined only by examining the local information stored by the two nodes. So, the network can be manipulated by keeping only its labels in memory, any other global information on the graph (like its matrix) can be removed. The goal is to minimize the maximum length of a label associated with a vertex while keeping fast adjacency queries.

---

<sup>\*</sup>Both authors are supported by the French ANR-projects “GéoComp” and “Graal”.

Implicit representation of graphs with short labels have been investigated by Kannan, Naor and Rudich [3] and is widely used in distributed computed, e.g. in [4]. They construct adjacency labeling schemes for several families of graphs including trees with labels of  $2 \log n$  bits. The size of labels for implicit representation of trees was later improved in a non-trivial way to  $\log + \log \log n + O(1)$  by Chung [2]. This result has been further improved by the use of an efficient labeling scheme to  $\log n + O(\log^* n)$  bits<sup>1</sup>.

Unfortunately, implicit representation is not always sufficient. For instance, address based routing between nodes in networks can not be achieved using adjacency. In this paper, we consider labeling schemes for various relationships between nodes of small distance in trees. For instance, we show, that a distributed representation scheme supporting parent and sibling queries can be achieved with labels of size  $\log n + 2 \log \log n + 2$ , improving a recent result of Alstrup, Bille and Rauhe [1].

More generally, we say that two nodes  $v$  and  $w$  with nearest common ancestor  $z$  are  $(k_1, k_2)$ -related if the distance from  $v$  to  $z$  is  $k_1$  and the distance from  $w$  to  $z$  is  $k_2$ . For any integer  $k$ , a *k-relationship scheme* is a distributed representation scheme that supports tests for whether  $v$  and  $w$  are  $(k_1, k_2)$ -related for all nodes  $v$  and  $w$  and all positive integers  $k_1, k_2 \leq k$ . In particular, a 1-relationship scheme supports tests for whether two nodes are  $(0, 0)$ ,  $(1, 0)$ ,  $(0, 1)$  or  $(1, 1)$ -related. We propose a  $k$ -relationship scheme for trees of  $n$  nodes with labels of size  $\log n + O(k \log(k \log(n/k)))$  bits. This improves the scheme presented in [1] that use labels of  $\log n + O(k^2 \log(k \log n))$  bits.

## 2. MAIN RESULT

**THEOREM 1.** *The family of  $n$ -node rooted trees enjoys a  $k$ -relationship scheme with labels of  $\log n + O(k \log(k \log(n/k)))$  bits, and with a constant adjacency query time.*

The basic idea of our scheme is to store into the label of each node  $v$ , an identifier of  $v$  and its  $k$  closest ancestors. Indeed, to test if  $u$  and  $v$  are  $(k_1, k_2)$ -related, it suffices to test if the ancestor at distance  $k_1$  of  $u$  is equal to the ancestor at distance  $k_2$  of  $v$ , and the ancestor at distance  $k_1 - 1$  of  $u$  differs from the ancestor at distance  $k_2 - 1$  of  $v$ .

Let  $T$  be a rooted tree of  $n$  nodes. We define the *k-complex* of a node  $u$  as the set of ancestors of  $u$  at distance at most  $k$ ,  $u$  included. A *branch* of  $T$  is a path leading from a leaf to the root of  $T$ .

---

<sup>1</sup> $\log^* n$  denotes the number of times  $\log$  should be iterated to get a constant.

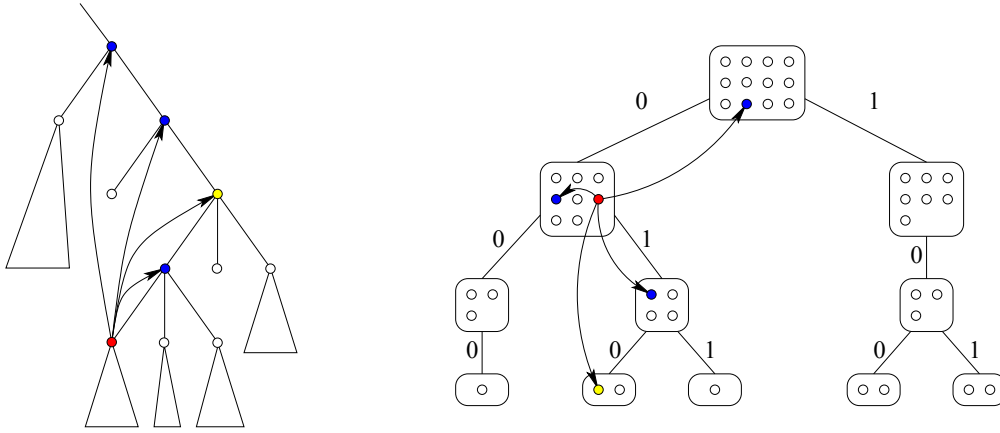


Figure 1: An example of  $k$ -ancestor decomposition.

DEFINITION 1. A  $k$ -ancestor decomposition of a rooted tree  $T$  is a rooted binary tree  $B$  where nodes, called parts, form a partition of  $V(T)$  such that, for any  $k$ -complex  $K$  of  $T$ , the set of parts containing a node of  $K$  is contained in a branch of  $B$ .

Our result highly depends on the following key lemma.

LEMMA 1. Every  $n$ -node tree  $T$  enjoys a  $k$ -ancestor decomposition such that every part of depth  $h$  contains at most  $k \cdot (\log(n/k) - h)$  nodes of  $T$ .

The idea is to construct from  $T$  a graph  $G$  obtained by adding an edge between every  $u$  and its proper ancestors at distance  $\leq k$ . We then observe that every subgraph  $H$  of  $G$  has a subset of  $k + 1$  nodes, called *half-separator*, whose removal leaves  $H$  in connected components with less than  $|V(H)|/2$  nodes. The root of  $B$  is constructed by finding iteratively  $O(\log(n/k))$  half-separators, and by grouping all the resulting connected components in two sets  $V_1, V_2$ , each with less than  $n/2$  nodes. By this way, there is no edges of  $G$  between  $V_1$  and  $V_2$ . The tree  $B$  is completed by performing similarly and recursively the subgraphs induced by  $V_1$  and  $V_2$ . Eventually,  $B$  is a  $k$ -ancestor decomposition since 1)  $k$ -complexes of  $T$  induce cliques in  $G$ ; and 2) all edges of  $G$  belongs to the same branch in  $B$ .

We define the function  $I(h) = \sum_{i=0}^h k \cdot (\log(n/k) - i)$  if  $h \geq 0$  and  $I(h) = 0$  otherwise. Let  $X$  be a part of  $B$  at depth  $h$ . By Lemma 1,  $|X| \leq I(h) - I(h - 1)$ . We denote by  $\text{path}(X)$  the binary word of length  $h$  defining the unique path from the root of  $T$  to  $X$ .

We associated with each  $u \in X$ , its *rank*, a unique integer  $\text{rank}(u) \in [0, |X|)$ , and its *position*, defined by  $\text{pos}(u) = \text{rank}(u) + I(h - 1)$ . We order the nodes of a  $k$ -complex according to their positions. The *apex* of a  $k$ -complex  $K$  is the node  $a \in K$  with the greatest position. Observe that the positions are relative to a branch of  $B$ : every pair of nodes whose parts are on the same branch have distinct positions, and conversely, the parts of any two nodes having the same positions are not related (and so are not on the same branch).

Let  $u$  be a node of  $T$ , let  $K_u$  be its  $k$ -complex,  $a$  the apex of  $K_u$ , and  $B_a$  the part in  $B$  that contains  $a$ . The label of node  $u$  is defined by the following quadruple:

$$\text{label}(u) = (\text{path}(B_a), \text{rank}(a), P_u, D_u)$$

where:

- $P_u = \{\text{pos}(v) \mid v \in K_u, v \neq a\}$ ; and
- $D_u = (d_1, d_2, \dots, d_{k+1})$  where  $d_i$  is the distance in  $T$  from  $u$  to the  $i$ -th node of  $K_u$ .

The label length is  $\log n + O(k \log(k \log(n/k)))$ . We can check that the labels uniquely identify each  $k$ -complex, and that given two  $k$ -complexes  $K_u$  and  $K_v$ , the  $k$ -relation can be determined. Using a more sophisticated data structure, we show how to perform the test in constant time while preserving the  $\log n + O(k \log(k \log(n/k)))$  label length.

### 3. REFERENCES

- [1] S. Alstrup, P. Bille, and T. Rauhe. Labeling schemes for small distances in trees. *SIAM Journal on Discrete Mathematics*, 19(2):448–462, 2005.
- [2] F. R. K. Chung. Universal graphs and induced-universal graphs. *Journal of Graph Theory*, 14:443–454, 1990.
- [3] S. Kannan, M. Naor, and S. Rudich. Implicit representation of graphs. In *20<sup>th</sup> Annual ACM Symposium on Theory of Computing (STOC)*, pages 334–343. ACM Press, may 1988.
- [4] A. Korman, S. Kutten, and D. Peleg. Proof labeling system. In *24<sup>th</sup> Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 9–18. ACM Press, jul 2005.