

On Randomized Representations of Graphs Using Short Labels*

Pierre Fraigniaud

CNRS and Univ. Paris Diderot
pierre.fraigniaud@liafa.jussieu.fr

Amos Korman

CNRS and Univ. Paris Diderot
amos.korman@liafa.jussieu.fr

Abstract

Informative labeling schemes consist in labeling the nodes of graphs so that queries regarding any two nodes (e.g., are the two nodes adjacent?) can be answered by inspecting merely the labels of the corresponding nodes. Typically, the main goal of such schemes is to minimize the *label size*, that is, the maximum number of bits stored in a label. This concept was introduced by Kannan et al. [STOC'88] and was illustrated by giving very simple and elegant labeling schemes, for supporting adjacency and ancestry queries in n -node trees; both these schemes have label size $2 \log n$. Motivated by relations between such schemes and other important notions such as *universal* graphs, extensive research has been made by the community to further reduce the label sizes of such schemes as much as possible. The current state of the art *adjacency* labeling scheme for trees has label size $\log n + O(\log^* n)$ by Alstrup and Rauhe [FOCS'02], and the best known *ancestry* scheme for (rooted) trees has label size $\log n + O(\sqrt{\log n})$ by Abiteboul et al., [SICOMP 2006].

This paper aims at investigating the above notions from a probabilistic point of view. Informally, the goal is to investigate whether the label sizes can be improved if one allows for some probability of mistake when answering a query, and, if so, by how much. For that, we first present a model for probabilistic labeling schemes, and then construct various probabilistic one-sided error schemes for the adjacency and ancestry problems on trees. Some of our schemes significantly improve the bound on the label size of the corresponding deterministic schemes, while the others are matched with appropriate lower bounds showing that, for the resulting guarantees of success, one cannot expect to do much better in term of label size.

Regular submission to SPAA 2009

*Both authors are supported by COST Action 295 "DYNAMO", ANR project ALADDIN, and INRIA project GANG.

1 Introduction

Network representations play an important role in many domains of computer science, ranging from data structures and graph algorithms, to parallel and distributed computing, and communication networks. Traditional network representations are usually global in nature. That is, in order to retrieve useful information, one must access a global data structure representing the entire network, even if the desired information is solely local, pertaining to only a few nodes. In contrast, the notion of *informative labeling schemes* suggests the use of a *local* representation of the network. The principle is to associate a label with each node, selected in a way that enables to infer information about any two nodes *directly* from their labels, without using *any* additional sources of information. Hence in essence, this method bases the entire representation on the set of labels alone. Obviously, labels of unrestricted size can be used to encode any desired information, including in particular the entire graph structure. The focus is thus on informative labeling schemes which use labels as *short* as possible.

The notion of informative labeling schemes was introduced in [9], and was illustrated there by giving very simple and elegant labeling schemes for supporting *adjacency* and *ancestry* queries in n -node (rooted) trees. Both schemes have label size $2 \log n$ bits. That paper also observes a strong relation between adjacency labeling schemes and *induced universal* graphs. Precisely, it was shown that there exists an adjacency labeling scheme with label size k for a graph family \mathcal{G} if and only if there exists an *induced universal* graph for \mathcal{G} with 2^k nodes (i.e., a graph \mathcal{U} that contains every graph in \mathcal{G} as induced subgraph).

Due to the above relation, a considerable amount of research has been devoted to improve the upper bound on the label size of adjacency labeling schemes on trees [4, 10]. Indeed, a small improvement in the label size has a significant impact on the size of the resulting universal graph. In [10] an adjacency labeling scheme using labels of size $\log n + O(\sqrt{\log n})$ bits is presented, and in [4] the label size was further reduced to $\log n + O(\log^* n)$. This current state of the art bound implies the existence of an induced universal graph for the family of n -node trees with $2^{O(\log^*(n))} n$ nodes, hence slightly more than linear. Proving or disproving the existence of an adjacency labeling scheme for trees using label of size $\log n + O(1)$ remains a central open problem in the design of informative labeling schemes.

In parallel to these aforementioned researches on adjacency labeling schemes on trees, much efforts have been devoted to improve the label size of *ancestry* labeling schemes on rooted trees. Apart from the purely theoretic interest, these studies have also been motivated by direct applications to XML search engines. As it turns out, reducing the length of the label size of an ancestry labeling scheme, even by a small factor, is critical for the reduction of memory cost and for performance improvement of the search engines. For more details regarding XML search engines, and their relation to ancestry schemes, see, e.g., [1, 2, 7]. The current state of the art upper bound for ancestry labeling is $\log n + O(\sqrt{\log n})$ bits [1], which is still far from the known $\log n + \Omega(\log \log n)$ lower bound [3]. In addition, the hidden constant in front of the $\sqrt{\log n}$ additive factor of the scheme in [1] is rather large, and makes this factor be the dominated one, in typical XML trees. As a result, [11] suggested another ancestry labeling schemes whose worst case bound is $1.5 \log n + O(1)$ but performs better than the scheme of [1] for typical XML data.

This paper aims at investigating the above notions from a probabilistic point of view. Our goal is to check whether the label sizes can be improved (and if so by how much) if one allows for some probability of mistake when answering a query. Let us now give an informal description of the model we use. A formal description and discussion appears in Section 2.

Probabilistic Labeling Schemes. We focus on one-sided error labeling schemes, with fixed guarantee $p \in (0, 1]$. More specifically, let us consider a boolean predicate f defined on pairs of vertices of graphs belonging to a graph family \mathcal{G} . Let us denote by $\lambda(u)$ the label given to node u by the labeling schemes. A

(probabilistic) one-sided error f -labeling scheme with guarantee p for the graph family \mathcal{G} is a randomized labeling scheme such that, for any $G \in \mathcal{G}$, and any $(u, v) \in V(G) \times V(G)$,

- if the predicate f is satisfied by (u, v) , then the probability that the scheme accepts $(\lambda(u), \lambda(v))$ is at least p , and
- if the predicate f is not satisfied by (u, v) , then the probability that the scheme accepts $(\lambda(u), \lambda(v))$ is 0.

The probability space potentially depends on both the randomized assignment of labels λ to the nodes, and on the randomized decision on whether to accept the pair $(\lambda(u), \lambda(v))$, taken solely as a function of the labels $\lambda(u)$ and $\lambda(v)$.

Our Results. This paper introduces the notion of probabilistic labeling schemes and demonstrates its fruitfulness by establishing tradeoffs between the sizes of the labels and the guarantee p , for the important adjacency and ancestry problems in trees.

We first study one-sided error adjacency and non-adjacency labeling schemes in trees, the latter actually yielding richer developments than the former. Specifically, we prove that, for any $k \geq 1$, there is a one-sided error adjacency labeling scheme for n -node trees with guarantee $1 - \frac{1}{2^k}$, using labels of size at most $\log n + O(k)$ bits. It turns out that this bound is essentially tight (in the sense that the $\log n$ term cannot be avoided). Indeed, we prove that any one-sided error adjacency scheme in n -node trees requires labels of size at least $\log n - O(1)$ bits, even for constant guarantee.

For non-adjacency on trees, we design a one-sided error labeling scheme with guarantee $1 - \frac{1}{2^k}$, using labels of size at most $2(k+1)$. This result has an important consequence on the design of universal graphs. In particular, it allows us to construct, for any $n \geq 1$, a graph \mathcal{U} on $O(n)$ vertices such that, any n -node tree T can be mapped to \mathcal{U} in such a way that: (1) every two adjacent nodes in T are mapped to two adjacent nodes in \mathcal{U} , and (2) every two different non-adjacent nodes in T are, with probability at least $1 - \frac{1}{\sqrt{n}}$, mapped to two different non-adjacent nodes in \mathcal{U} .

We also design one-sided error labeling schemes for ancestry and non-ancestry in (rooted) trees. We prove that, for any $k > 1$, there exists a one-sided error ancestry labeling scheme for trees, with labels on $\log n - k/2 + O(\sqrt{\log n})$ bits, and guarantee $1/2^k$. Although this guarantee may appear quite small, this is close to optimal using labels of that size. Indeed, we prove that any one-sided error ancestry labeling scheme with label size $\log n - k/2$ cannot have better guarantee than $O(1/2^{\frac{k}{2}})$. For non-ancestry, we design a one-sided error scheme with guarantee $\frac{1}{2}$ using $\lceil \log n \rceil$ -bit labels. Again, this label size is optimal (up to an additive constant) for such a guarantee.

Finally, we note that our constructions are simple and therefore can easily be implemented in practice. See Table 1 for a summary of most of our results.

Outline. Our paper is organized as follows. In Section 2, we present the model for probabilistic labeling schemes. Upper and lower bounds on one-sided error schemes for the adjacency and non-adjacency predicates on trees are presented in Section 3, and the ones about ancestry and non-ancestry are given in Section 4. Conclusions and directions for future work appear in Section 5.

		Deterministic	Probabilistic
Adjacency:	Lower bound	$\log n - O(1)$	$\log n + \log p - O(1)$ for guarantee p
	Upper bound	$\log n + O(\log^* n)$ [4]	$\log n + O(k)$ for guarantee $1 - 1/2^k$
Non-adjacency:	Lower bound	$\log n - O(1)$	—
	Upper bound	$\log n + O(\log^* n)$ [4]	$2(k + 1)$ for guarantee $1 - 1/2^k$
Ancestry:	Lower bound	$\log n + \Omega(\log \log n)$ [3]	$\log n + \log p - O(1)$ for guarantee p
	Upper bound	$\log n + O(\sqrt{\log n})$ [1]	$\log n - k/2 + O(\sqrt{\log n})$ for guarantee $1/2^k$
Non-ancestry:	Lower bound	$\log n + \Omega(\log \log n)$ [3]	$\log n + \log p - O(1)$ for guarantee p
	Upper bound	$\log n + O(\sqrt{\log n})$ [1]	$\lceil \log n \rceil$ for guarantee $1/2$

Table 1: Summary of results: deterministic vs. probabilistic labeling

2 Model

Let us first recall the formal definition of informative labeling schemes (see [9, 14] for more details). A vertex-labeling of the graph G is a function λ assigning a label $\lambda(u) \in \mathbb{N}$ to every node $u \in V(G)$. Let f be a function defined on pairs of vertices of graphs belonging to some graph family \mathcal{G} . In this paper, we mainly focus on boolean functions, but most concepts introduced in the paper apply to any function. An f -labeling scheme for \mathcal{G} is a pair $(\mathcal{M}, \mathcal{D})$, where \mathcal{M} is called the *marker*, and \mathcal{D} is called the *decoder*, satisfying the following. The marker is a function that, given any $G \in \mathcal{G}$, returns a vertex-labeling $\lambda = \mathcal{M}(G)$ for G . The decoder is a function that, given a pair of label $(\ell, \ell') \in \mathbb{N} \times \mathbb{N}$, returns a boolean. These marker and decoder must satisfy that, for any $G \in \mathcal{G}$, and any $(u, v) \in V(G) \times V(G)$,

$$f(u, v) = \mathcal{D}(\lambda(u), \lambda(v)). \quad (1)$$

We define a *probabilistic* f -labeling scheme for \mathcal{G} as an f -labeling scheme for \mathcal{G} in which the marker and/or the decoder can be probabilistic. A probabilistic marker is a random function \mathcal{M} that, given any $G \in \mathcal{G}$, returns a (random) vertex-labeling $\lambda = \mathcal{M}(G)$ for G . A probabilistic decoder is a random function \mathcal{D} that, given a pair of labels (ℓ, ℓ') , returns a (random) boolean $\mathcal{D}(\ell, \ell')$. These random functions \mathcal{M} and \mathcal{D} are independent, in the sense that, for any boolean x , for any two labels (ℓ, ℓ') , for any graph $G \in \mathcal{G}$, and any $(u, v) \in V(G) \times V(G)$,

$$\Pr[\mathcal{D}(\ell, \ell') = x \mid \ell = \lambda(u) \wedge \ell' = \lambda(v)] = \Pr[\mathcal{D}(\ell, \ell') = x].$$

The analog of Equation 1 in the probabilistic setting depends on what guarantee one wants to achieve. In this paper, we focus on probabilistic (one-sided error) f -labeling scheme, defined as follow.

Definition 1 *Let f be a boolean function defined on pairs of vertices of graphs belonging to a graph family \mathcal{G} , and let $p \in (0, 1]$. A (probabilistic) one-sided error f -labeling scheme with guarantee p for \mathcal{G} is a pair $(\mathcal{M}, \mathcal{D})$ of probabilistic marker and/or decoder such that, for any $G \in \mathcal{G}$, and any $(u, v) \in V(G) \times V(G)$, the following holds.*

- $f(u, v) = 1 \Rightarrow \Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] \geq p$;
- $f(u, v) = 0 \Rightarrow \Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] = 0$;

where $\lambda = \mathcal{M}(G)$.

The main criterium for measuring the quality of a (deterministic) labeling scheme is the size of the labels, i.e., the maximum number of bits used to encode a label produced by the marker, over all nodes in all graphs in \mathcal{G} . Usually, the label size is expressed as a function of the number of nodes, hence one is typically interested in studying the maximum number of bits in a label produced by the marker, among all nodes in graphs in \mathcal{G}_n where $\mathcal{G}_n = \{G \in \mathcal{G}, |V(G)| \leq n\}$. We use a similar measure for evaluating probabilistic labeling schemes and define the *label size* as the maximum number of bits used in a label produced by the (probabilistic) marker, taken over all nodes in all graph in \mathcal{G}_n , and over all possible coin tosses.

The other quantities that are also sometimes considered for measuring the quality of a labeling scheme are the time-complexities of the marker and the decoder. That is, although the definition of labeling scheme is expressed in terms of functions marker and decoder, one may be particularly interested in the case where these functions are computable by an algorithm with low time-complexity. This paper will provide such functions, but our main concern remains the size of the labels. In fact, before going further, we want to point out that in a context in which the size of the label is the main issue, then there is no need to consider probabilistic decoders. Indeed, Definition 1 provides two sources of randomness for a labeling scheme, in both the marker and the decoder. It is a simple observation that only one source of randomness is sufficient for minimizing the label size complexity, the one provided by the marker:

Observation 1 *Let f be a boolean function, and $p \in (0, 1]$. Let $(\mathcal{M}, \mathcal{D})$ be a one-sided error f -labeling scheme with guarantee p for a class of graphs \mathcal{G} . Then there exists a deterministic decoder \mathcal{D}' , such that $(\mathcal{M}, \mathcal{D}')$ is a one-sided error f -labeling scheme with guarantee p for \mathcal{G} .*

Proof. Let \mathcal{D}' be defined by:

$$\forall i, j \geq 0, \quad \mathcal{D}'(i, j) = \begin{cases} 1 & \text{if } \Pr[\mathcal{D}(i, j) = 1] > 0 \\ 0 & \text{otherwise} \end{cases}$$

By this setting of \mathcal{D}' , we get

$$\mathcal{D}'(i, j) \geq \Pr[\mathcal{D}(i, j) = 1].$$

On the other hand, for any $(u, v) \in V(G) \times V(G)$, we have:

$$\Pr[\mathcal{D}'(\lambda(u), \lambda(v)) = 1] = \sum_{i, j} \mathcal{D}'(i, j) \cdot \Pr[\lambda(u) = i \wedge \lambda(v) = j] \quad (2)$$

Therefore we get

$$\Pr[\mathcal{D}'(\lambda(u), \lambda(v)) = 1] \geq \sum_{i, j} \Pr[\mathcal{D}(i, j) = 1] \cdot \Pr[\lambda(u) = i \wedge \lambda(v) = j].$$

By the independence of the marker \mathcal{M} and the decoder \mathcal{D} , the r.h.s. of this latter inequality is equal to $\Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1]$. Therefore

$$\Pr[\mathcal{D}'(\lambda(u), \lambda(v)) = 1] \geq \Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1].$$

Assume first that $f(u, v) = 1$. Then

$$\Pr[\mathcal{D}'(\lambda(u), \lambda(v)) = 1] \geq \Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] \geq p$$

as required. Assume now that $f(u, v) = 0$. In this case, for any pair of labels (i, j) such that

$$\Pr[\lambda(u) = i \text{ and } \lambda(v) = j] > 0,$$

we must have $\Pr[\mathcal{D}(i, j) = 1] = 0$. Hence $\mathcal{D}'(i, j) = 0$ too for such pairs. By Eq. 2, we get

$$\Pr[\mathcal{D}'(\lambda(u), \lambda(v)) = 1] = 0$$

as required. This completes the proof of the observation. \square

From Observation 1, we can restrict our concern to one-sided error labeling schemes in which only the marker is probabilistic.

Another observation is that, as opposed to the definition of the complexity class RP [13], the choice of p is not arbitrary in our definition. Indeed, if only the decoder would have been probabilistic, then one could repeatedly apply it to increase the success guarantee. However, since the marker is randomized, it is not clear whether one can use repetition techniques to transform a given scheme with guarantee p to another with a higher guarantee.

As a final remark, we point out that, to satisfy a prescribed error guarantee $p > 0$, different applications of the marker must result in different pairs of nodes labeled "incorrectly", so that to satisfy the guarantee. For instance, in the case of an adjacency labeling scheme, since any pair of adjacent nodes must be certified adjacent by the scheme with probability at least p , this implies that any fixed pair of adjacent nodes must not remain undetected for all applications of the scheme (in fact, it must be detected for at least a fraction p of the applications). This makes one-sided error labeling quite different from labeling with slack [6] in which a correct answer is required but for a small fraction of pairs. In the setting of adjacency labeling with slack, there might be pairs of adjacent nodes that are never detected, whereas one-sided error adjacency schemes insure that any pair of adjacent nodes will be detected with positive probability.

3 Adjacency and Non-Adjacency Schemes in Trees

In this section we design one-sided error labeling schemes for the adjacency and non-adjacency functions in trees. In the deterministic setting, there is no difference between adjacency labeling schemes and non-adjacency labeling schemes. In contrast, the results below show that there is a huge difference between the two problems in the probabilistic setting. We begin by describing our one-sided error *non*-adjacency labeling scheme because it is extremely simple, and, in addition, it turns out to have richer developments than the adjacency scheme.

3.1 One-sided Error Non-Adjacency Schemes

According to the general definition of one-sided error labeling schemes, a one-sided error non-adjacency labeling scheme $(\mathcal{M}, \mathcal{D})$ with guarantee p for a class of graphs \mathcal{G} satisfies: for any $G \in \mathcal{G}$, and any $(u, v) \in V(G) \times V(G)$,

- $\{u, v\} \notin E(G) \Rightarrow \Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] \geq p$;
- $\{u, v\} \in E(G) \Rightarrow \Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] = 0$;

where $\lambda = \mathcal{M}(G)$.

Theorem 1 *For any $k \geq 1$, there exists a one-sided error non-adjacency labeling scheme with guarantee $1 - \frac{1}{2^k}$ for the class of trees, using $2(k + 1)$ -bit labels.*

Proof. Let T be a tree. The marker proceeds as follows. The tree is rooted at an arbitrary node r which is assigned the label $\lambda(r) = (\lambda_1(r), \lambda_2(r))$, where both sublabeled $\lambda_i(r)$, $i = 1, 2$, are chosen uniformly at random in $\{0, \dots, 2^{k+1} - 1\}$. The labeling of the nodes proceeds downward the tree: a node $u \neq r$, with parent v in T , is labeled $(\lambda_1(u), \lambda_2(u))$ where

$$\lambda_1(u) = \lambda_2(v) \quad \text{and} \quad \lambda_2(u) \text{ is chosen uniformly at random in } \{0, \dots, 2^{k+1} - 1\}.$$

Note that each sublabel $\lambda_i(u)$, $i \in \{1, 2\}$, can be encoded using at most $k + 1$ bits. To be able to distinguish the two sublabels within a label, we let each sublabel be encoded using precisely $k + 1$ bits, by padding zeros to the left of the encoding, if necessary. In particular, we get that the label size of our scheme is $2(k + 1)$.

Observe now, that by this labeling, if two nodes are adjacent, then the first label in the pair of labels of one of the nodes is equal to the second label in the pair of labels of the other node. We thus define the decoder as follows:

$$\mathcal{D}((\ell_1, \ell_2), (\ell'_1, \ell'_2)) = \begin{cases} 1 & \text{if } \ell_1 \neq \ell'_2 \text{ and } \ell'_1 \neq \ell_2 \\ 0 & \text{otherwise.} \end{cases}$$

Let u, v be two non-adjacent nodes. We have $\Pr[\lambda_i(u) = \lambda_j(v)] = \frac{1}{2^{k+1}}$ for any $i, j \in \{1, 2\}$ with $i \neq j$. It follows that the probability that \mathcal{D} returns 0 for that pair is at most $\frac{1}{2^k}$, and thus, the scheme has guarantee $1 - \frac{1}{2^k}$. \square

Theorem 1 has an important consequences on the implicit representation of graphs. Recall that a graph G is an *induced* subgraph of a graph \mathcal{U} if there exists a one-to-one mapping ϕ from $V(G)$ to $V(\mathcal{U})$ such that

$$\forall u, v \in V(G), \quad \{u, v\} \in E(G) \iff \{\phi(u), \phi(v)\} \in E(\mathcal{U}).$$

Also recall that a graph G is a *partial* subgraph of a graph \mathcal{U} if there exists a one-to-one mapping ϕ from $V(G)$ to $V(\mathcal{U})$ such that

$$\forall u, v \in V(G), \quad \{u, v\} \in E(G) \Rightarrow \{\phi(u), \phi(v)\} \in E(\mathcal{U}).$$

A graph \mathcal{U} is *universal* for a graph family \mathcal{G} if every graph in \mathcal{G} is a partial subgraph of \mathcal{U} [15]. This notion has applications to parallel computing and circuit design [5]. A graph \mathcal{U} is called *induced-universal* for a graph family \mathcal{G} if every graph in \mathcal{G} is an induced subgraph of \mathcal{U} . This more restricted notion of universality enables to relate the size of a universal graph for \mathcal{G} with the size of the graphs in \mathcal{G} [4]. There is in fact a strong relation between this second notion and adjacency labeling schemes: a graph family \mathcal{G} has an adjacency labeling scheme with label size k if and only if there exists an induced-universal graph for \mathcal{G} , with 2^k nodes [9]. Proving or disproving the existence of a universal graph with a linear number of nodes for the class of n -node trees is a central open problem in the design of informative labeling schemes.

Theorem 1, and the scheme described in its proof, have direct consequences to universal graph: for any $k \geq 1$, there exists a graph \mathcal{U} on 4^{k+2} nodes in which any tree T can be mapped to \mathcal{U} via a randomized mapping ϕ , in such a way that (1) $\{u, v\} \in E(T) \Rightarrow \{\phi(u), \phi(v)\} \in E(\mathcal{U})$, and (2) $\{u, v\} \notin E(T) \Rightarrow \{\phi(u), \phi(v)\} \notin E(\mathcal{U})$ with probability at least $1 - 1/2^k$. We give an explicit definition of the graph \mathcal{U} . For a binary string $b \in \{0, 1\}^\ell$, we denote $b = (b_1, \dots, b_\ell)$. We set:

$$V(\mathcal{U}) = \{(x, y) \in \{0, 1\}^{k+2} \times \{0, 1\}^{k+2} \mid x_1 \neq y_1\},$$

and edge set

$$E(\mathcal{U}) = \{\{(x, y), (x', y')\} \in V(\mathcal{U}) \times V(\mathcal{U}) \mid x' = y\}.$$

This setting insures that \mathcal{U} has no self-loops. The mapping ϕ is then defined from the random labeling λ corresponding to the marker described in the proof of Theorem 1, as follows. For the root r ,

$$\phi(r) = (0\lambda_1(r), 1\lambda_2(r)).$$

For a node $u \neq r$ whose parent v in T is mapped to $\phi(v) = (b\lambda_1(v), \bar{b}\lambda_2(v))$ for some $b \in \{0, 1\}$, we set

$$\phi(u) = (\bar{b}\lambda_1(u), b\lambda_2(u)).$$

This setting implies that two adjacent nodes in T are mapped to two adjacent nodes in \mathcal{U} . As a consequence, we get the following theorem.

Theorem 2 *For any $k \geq 1$, there exists a graph \mathcal{U} on 4^{k+2} vertices such that any tree T can be mapped into \mathcal{U} in such a way that two adjacent nodes in T are mapped to two adjacent nodes in \mathcal{U} , and every two different non-adjacent nodes in T are, with probability at least $1 - 1/2^k$, mapped to two different non-adjacent nodes in \mathcal{U} .*

Note that the graph \mathcal{U} is ‘good’ for all trees T , no matter which size they are. This is because some different nodes in T may be mapped to the same node in \mathcal{U} . However, this cannot be the case for two adjacent nodes in T , and the probability that two given different non-adjacent nodes in T are mapped to the same node in \mathcal{U} is at most $1/2^k$. Obviously, this embedding cannot be one-to-one for $k < \frac{1}{2} \log n - 2$, since for such k , \mathcal{U} has less than n nodes. Nevertheless, by fixing $k = \frac{1}{2} \log n + O(1)$, we get the following corollary:

Corollary 1 *There exists a graph \mathcal{U} of size $O(n)$ such that any n -node tree T can be mapped into \mathcal{U} , in such a way that two adjacent nodes in T are mapped to two adjacent nodes in \mathcal{U} , and every two different non-adjacent nodes in T are, with probability at least $1 - O(\frac{1}{\sqrt{n}})$, mapped to two different non-adjacent nodes in \mathcal{U} .*

3.2 One-sided Error Adjacency Schemes

We now turn to investigate probabilistic adjacency schemes in trees. Recall that a one-sided error adjacency labeling scheme with guarantee p for a class of graphs \mathcal{G} satisfies: for any $G \in \mathcal{G}$, and any $(u, v) \in V(G) \times V(G)$, (1) $\{u, v\} \in E(G) \Rightarrow \Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] \geq p$, and (2) $\{u, v\} \notin E(G) \Rightarrow \Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] = 0$, where $\lambda = \mathcal{M}(G)$.

Theorem 3 *For any $k \geq 1$, there exists a one-sided error adjacency labeling scheme with guarantee $1 - \frac{1}{2^k}$ for the class of trees, using labels on $\log n + 3k + O(1)$ bits in n -node trees. Moreover, this scheme insures that two different nodes get two different labels.*

Proof. The scheme is based on the existence on a (deterministic) adjacency labeling scheme (μ, δ) for forests whose trees are of bounded depth [8]. This latter scheme uses labels of size $\log n + 3 \log d + O(1)$ in n -node forests with trees of depth at most d .

Let $k \geq 1$, and let $d = 2^k$. Let T be an n -node tree. The probabilistic marker \mathcal{M} performs as follows. The tree T is rooted at an arbitrary node r . The level of a node in T is then defined top-down: r has level 0, its children have level 1, and so on. The marker chooses $\ell \in \{0, \dots, d-1\}$ uniformly at random. T is then transformed into a forest F by removing all edges between nodes at level $\ell + qd$ and nodes at level $\ell + qd + 1$ for all $q \geq 0$. The forest F has n nodes, and all trees in F have depth at most d . The marker μ is then used to mark the nodes of F . The decoder \mathcal{D} decodes the labels like δ .

Clearly, the decoder returns 0 for any two non-adjacent nodes in T , with probability 1. On the other hand, for any edge e of T , the probability that e is not in F is $1/d = 1/2^k$. Therefore, the probability that the decoder returns 0 for two adjacent nodes is $1 - 1/2^k$, as claimed. \square

Similarly to Corollary 1, we get the following corollary:

Corollary 2 For any $n \geq 1$, and any constant $\epsilon > 0$, there exists a graph \mathcal{U} on $O(n)$ nodes in which any n -node tree T can be one-to-one embedded in such a way that two non-adjacent nodes in T are mapped to two non-adjacent nodes in \mathcal{U} , and, with probability at least $1 - \epsilon$, two adjacent nodes in T are mapped to two adjacent nodes in \mathcal{U} .

We conclude this section by proving a lower bound on the label size of one-sided error adjacency labeling schemes. Indeed, the adjacency labeling scheme and the non-adjacency labeling scheme presented above differ significantly in the sense that, for a constant error guarantee, one is using constant size labels while the other is using labels of logarithmic size. Somewhat surprisingly, we show that this difference cannot be avoided. Indeed $(\log n)$ -bit labels are required for one-sided error adjacency labeling schemes with constant guarantee.

Theorem 4 For any $p \in (0, 1]$, any one-sided error adjacency labeling scheme with guarantee p for the class of trees use labels on at least $\log n + \log p - O(1)$ bits in n -node trees.

Proof. Let us consider the n -node path $P_n = (u_0, u_2, \dots, u_{n-1})$, and consider a one-sided error adjacency labeling scheme $(\mathcal{M}, \mathcal{D})$ with guarantee p for P_n . Assume that the scheme is using $m \leq n$ different integers to label the nodes of P_n . Note that the label size of the scheme is at least $\log m$. Our goal is thus to show that m is large. Let λ_i be the (random) label given to u_i by $\mathcal{M}(P_n)$. Let X_i be the random variable defined for $1 \leq i \leq n - 2$ by

$$X_i = \begin{cases} 1 & \text{if there exists } j > i \text{ such that } \lambda_j = \lambda_i \\ 0 & \text{otherwise.} \end{cases}$$

The random variable X_i depends on the random labeling λ . Assume that $X_i = 1$. Then $\mathcal{D}(\lambda_{i-1}, \lambda_i) = 0$. Indeed, there exists $j > i$ such that $\lambda_j = \lambda_i$ but u_j and u_{i-1} are not adjacent, and therefore one must have $\mathcal{D}(\lambda_{i-1}, \lambda_j) = 0$. Thus $X_i = 1$ enforces an error for the test of adjacency at $\{u_{i-1}, u_i\}$. Therefore, one must have $\Pr[X_i = 1] \leq 1 - p$, from which we derive $\mathbb{E}X_i \leq 1 - p$. Let $X = \sum_{i=1}^{n-2} X_i$. We get

$$\mathbb{E}X = \sum_{i=1}^{n-2} \mathbb{E}X_i \leq (n-2)(1-p).$$

On the other hand, for any fixed labeling λ , consider any label $\ell \in \{1, \dots, m\}$, and let $u_{i_1}, u_{i_2}, \dots, u_{i_r}$ be the r nodes labeled ℓ , with $i_1 < i_2 < \dots < i_r$. We get $X_{i_1} = X_{i_2} = \dots = X_{i_{r-1}} = 1$. Consequently, $X(\lambda) \geq (n-2) - m$. We therefore get that,

$$\mathbb{E}X = \sum_{\lambda} X(\lambda) \cdot \Pr[\mathcal{M}(P_n) = \lambda] \geq (n-2) - m.$$

Combining the two bounds on $\mathbb{E}X$, we get $m \geq p(n-1)$. Thus, the label size is at least $\log n + \log p - O(1)$, as claimed. \square

4 Ancestry and Non-Ancestry Schemes

We now turn our attention to one-sided error labeling schemes for ancestry and non-ancestry in rooted trees. Let T be a tree rooted at node r . A node u is an *ancestor* of a node v iff $u \neq v$ and u is on the shortest path from v to r in T . A one-sided error ancestry labeling scheme with guarantee p for a class of rooted trees \mathcal{T} satisfies: for any $T \in \mathcal{T}$, and any $(u, v) \in V(T) \times V(T)$,

- if u is an ancestry of v in T then $\Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] \geq p$;
- if u is not an ancestry of v in T then $\Pr[\mathcal{D}(\lambda(u), \lambda(v)) = 1] = 0$;

where $\lambda = \mathcal{M}(G)$. In the deterministic setting, the best known ancestry labeling scheme has label size $\log n + O(\sqrt{\log n})$ in n -node trees, and no ancestry labeling scheme can have labels of size smaller than $\log n + \Omega(\log \log n)$. In contrast, we prove that, in the probabilistic setting, there are adjacency labeling schemes for trees using labels of size smaller than $\log n$ bits.

Theorem 5 *For any integer $k \geq 1$, there exists a one-sided error ancestry labeling scheme with guarantee $1/2^k$ for the class of trees, using labels on $\log n - k/2 + O(\sqrt{\log n})$ bits.*

Proof. Let k be an integer, and let T be an n -node tree rooted at node r . The marker \mathcal{M} selects $n^{1-k/(2 \log n)}$ nodes chosen uniformly at random among the nodes of T . Let S be the set of selected nodes. The tree T is transformed into another tree T' by keeping only the nodes in S . More precisely, an edge of T is called “void” if either none of its endpoints belong to S , or exactly one of its endpoints belongs to S and it is the parent of the other endpoint. T' is obtained from T by contracting all void edges. Note that this transformation preserves the ancestry relation. Then the marker applies the scheme in [1] for labeling the nodes in S , producing labels of $\log |S| + O(\sqrt{\log |S|})$ bits, that is, $\log n - \frac{k}{2} + O(\sqrt{\log n})$ bits. Nodes not in S are labeled -1 . The decoder \mathcal{D} is then defined as follows. Let \mathcal{D}' be the decoder defined in [1] for the deterministic ancestry labeling scheme.

$$\mathcal{D}(\ell, \ell') = \begin{cases} 0 & \text{if } \ell = -1 \text{ or } \ell' = -1 \\ \mathcal{D}'(\ell, \ell') & \text{otherwise.} \end{cases}$$

Let u and v be two nodes in T , and assume that u is an ancestor of v . The probability that u and v are selected by the marker is

$$\left(\frac{n^{1-k/(2 \log n)}}{n} \right)^2 = \frac{1}{n^{k/\log n}} = \frac{1}{2^k},$$

and therefore the scheme has guarantee $\frac{1}{2^k}$. □

Now, let us focus on non-ancestry labeling schemes.

Theorem 6 *There exists a one-sided error non-ancestry labeling scheme with guarantee $\frac{1}{2}$ for the class of rooted trees, using $\lceil \log n \rceil$ -bit labels.*

Proof. Consider the following probabilistic marker \mathcal{M} acting on a tree T rooted at node r . The marker first randomly chooses, for every node v , an ordering of v 's children in a uniform manner, i.e., every ordering of the children of v has the same probability to be selected by the marker. Then, according to the resulted orderings, the marker performs a DFS traversal that starts at r . Along the resulted DFS tour, the marker labels the nodes it visits by consecutive integers, with r labeled 0. Given two labels i, j , the decoder outputs:

$$\mathcal{D}(i, j) = \begin{cases} 0 & \text{if } i < j; \\ 1 & \text{otherwise.} \end{cases}$$

Let u and v be two nodes in T , and let λ_u and λ_v denote their labels. Clearly, if u is an ancestor of v , then $\lambda_u < \lambda_v$, no matter which orderings were chosen by the marker. Thus, in this case, $\mathcal{D}(\lambda_u, \lambda_v) = 0$ with probability 1. Now, if u is not an ancestor of v , we consider two cases. First, if u is a descendant of v then $\lambda_u > \lambda_v$ and therefore $\mathcal{D}(\lambda_u, \lambda_v) = 1$. If, however, u and v are non-comparable, i.e., neither one is an ancestor of the other, then the probability that the DFS tour visited u before it visited v is precisely $1/2$, i.e., $\Pr[\lambda_u < \lambda_v] = 1/2$. Hence the guarantee for non-ancestry is $1/2$. □

The result below shows that the bound of Theorem 5 is almost tight in the case of ancestry labeling, for $p = 1/2^k$, and the bound of Theorem 6 is tight for $p = 1/2$.

Theorem 7 *For any $p \in (0, 1]$, any one-sided error ancestry (resp., non-ancestry) labeling scheme with guarantee p for the class of trees use labels on at least $\log n + \log p - O(1)$ bits in n -node trees.*

Proof. We place ourselves in the same setting as in the proof of Theorem 4, and we use the same notations, including the random variables X_i , now defined for $i = 0, \dots, n-2$. Consider first a one-sided error ancestry labeling scheme $(\mathcal{M}, \mathcal{D})$. Assume that $X_i = 1$, i.e., that there exists $j > i$ such that $\lambda_i = \lambda_j$. Since u_j is not an ancestor of u_{i+1} , we must have $\mathcal{D}(\lambda_j, \lambda_{i+1}) = 0$. Therefore, $\mathcal{D}(\lambda_i, \lambda_{i+1}) = 0$. But u_i is in fact an ancestor of u_{i+1} . Thus $X_i = 1$ enforces an error for the test of ancestry between u_i and u_{i+1} . Therefore, as for the adjacency case, one must have $\Pr[X_i = 1] \leq 1 - p$, and hence we get

$$\mathbb{E}X = \sum_{i=0}^{n-2} \mathbb{E}X_i \leq (n-1)(1-p).$$

On the other hand, $X(\lambda) \geq (n-1) - m$ for any vertex-labeling λ , and thus

$$\mathbb{E}X = \sum_{\lambda} X(\lambda) \cdot \Pr[\mathcal{M}(P_n) = \lambda] \geq (n-1) - m.$$

Combining the two bounds on $\mathbb{E}X$, we get $m \geq (n-1)p$, establishing the result for ancestry. We complete the proof by noticing that by turning the path upside down, the same arguments as above hold for non-ancestry as well. \square

5 Conclusion and Further Study

This paper introduces the notion of probabilistic labeling scheme, and illustrates it by investigating two important labeling problems on trees, namely adjacency and ancestry. Our one-sided error adjacency scheme and, especially, our one-sided error non-adjacency scheme also relate to the existence of small graphs \mathcal{U} for which every tree can be mapped into using a probabilistic mapping that preserves properties of T . This may be found useful in understanding relations between universal graphs and probabilistic embeddings. In addition, apart from its theoretical interest, our one-sided error ancestry and non-ancestry schemes can possibly be implemented in XML search engines, for improving their memory cost.

Naturally, this study opens further directions of research for probabilistic labeling schemes on other types of graph families, and other types of functions, as was done in the deterministic setting. Moreover, other versions of probabilistic labeling schemes can be considered as well, such as, e.g., zero-sided error labeling schemes. Finally, we note that even though our schemes are time efficient, our main concern was the tradeoff between the label size and the probability of success guarantee. Other research on the subject may also include time considerations, in which case a probabilistic decoder may potentially play a role.

Acknowledgements. The authors are thankful to Nicolas Schabanel for fruitful discussions on the topic of this paper.

References

- [1] S. Abiteboul, S. Alstrup, H. Kaplan, T. Milo and T. Rauhe. Compact labeling schemes for ancestor queries. *SIAM Journal on Computing* 35:1295–1309, 2006.
- [2] S. Abiteboul, H. Kaplan, and T. Milo. Compact labeling schemes for ancestor queries. In 12th ACM-SIAM Symp. on Discrete Algorithms (SODA), pages 547-556, 2001.
- [3] S. Alstrup, P. Bille and T. Rauhe. Labeling Schemes for Small Distances in Trees. *SIAM J. Discrete Math* 19(2):448–462, 2005.
- [4] S. Alstrup and T. Rauhe. Small induced-universal graphs and compact implicit graph representations. In 43rd IEEE Symp. on Foundations of Computer Science (FOCS), pages 53-62, 2002.
- [5] M. Capalbo. A small universal graph for bounded-degree planar graphs. In 10th ACM-SIAM symposium on Discrete algorithms (SODA), pages 156-160, 1999.
- [6] T.-H. Chan, M. Dinitz, and A. Gupta. Spanners with Slack. In 14th European Symposium on Algorithms (ESA), LNCS 4168, pp. 196-207, 2006.
- [7] E. Cohen, H. Kaplan, and T. Milo. Labeling dynamic XML trees. In 21st ACM Symp. on Principles of Database Systems (PODS), pages 271-281, 2002.
- [8] P. Fraigniaud and A. Korman. Compact Ancestry Labeling Schemes for Trees of Small Depth. Submitted, 2009. (see <http://arxiv.org/pdf/0902.3081>).
- [9] S. Kannan, M. Naor, and S. Rudich. Implicit Representation of Graphs. *SIAM J. on Discrete Math* 5: 596-603, 1992.
- [10] H. Kaplan and T. Milo. Short and simple labels for small distances and other functions. In Workshop on Algorithms and Data Structures (WADS), pages 246-257, 2001.
- [11] H. Kaplan, T. Milo and R. Shabo. A Comparison of Labeling Schemes for Ancestor Queries. In 19th ACM-SIAM Symp. on Discrete Algorithms (SODA), pages 954-963, 2002.
- [12] A. Korman, D. Peleg, and Y. Rodeh. Constructing Labeling Schemes Through Universal Matrices. *Algorithmica*, to appear.
- [13] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [14] D. Peleg. Informative labeling schemes for graphs. *Theoretical Computer Science* 340(3):577-593, 2005.
- [15] R. Rado. Universal graphs and universal functions. *Acta Arithmetica* 9:331-340, 1964.
- [16] M. Thorup and U. Zwick. Compact routing schemes. In 13th ACM Symp. on Parallel Algorithms and Architecture (SPAA), pages 1–10, 2001.