Chapter LXX
On the Query Evaluation in XML Databases

Yanjun Chen
University of Winnipeg, Canada

INTRODUCTION

With the growing importance of XML in data exchange, much research has been done in providing flexible query mechanisms to extract data from XML documents. A core operation for XML query processing is to find all occurrences of a twig pattern Q (or small tree) in a document T. Prior work has typically decomposed Q into binary structural relationships, such as parent-child and ancestor-descendant relations, or root-to-leaf paths. The twig matching is achieved by: (a) matching the binary relationships or paths against XML databases, and (b) using the join algorithms to stitch together all the matching binary relationships or paths. In the worst case, the time for doing joins can be exponential (in the number of query nodes or decomposed paths). In this chapter, we discuss a new algorithm for this task with no path joins involved. The time and space complexities of the algorithm are bounded by O ($|T|\cdot Q_{leaf}$) and O ($T_{leaf}\cdot Q_{leaf}$), respectively, where T_{leaf} stands for the number of the leaf nodes in T and Q_{leaf} for the number of the leaf nodes in Q. Our experiments show that our method is efficient in supporting twig pattern queries.

BACKGROUND

In an XML databases, a set of XML documents is maintained. Abstractly, each document can be considered as a tree structure with each node labeled with an element name from a finite alphabet $\sum$ or a string value, and an edge for the elment-subelement relationship. For instance, the XML document shown in Figure. 1(a) can be represented a tree structure as shown in Figure 1(b).

Queries in XML query languages, such as XPath (Florescu and Kossman, 1999; World Wide Web Consortium, 2007), XQuery (World Wide Web Consortium, 2001), XML-QL (Dutch, Fernandez, Florescu, 1999), and Quilt (Chamberlin,

Figure 1. A sample XML file and its tree representation

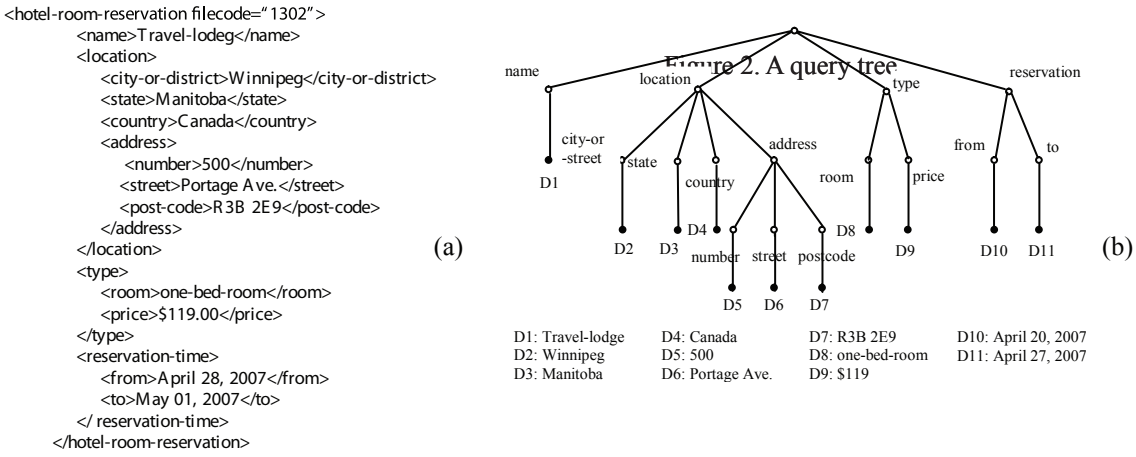

Figure 2. A query tree

et al., 1999; Chamberlin, et al., 2000), typically specify patterns of selection predicates on multiple elements that also have some specified tree structured relations. As an example, consider the query tree shown in Figure 3.

This query asks for any node labeled with *location* (node 2) that is a child of some node labeled with *hotel-room-reservation* (node 1). In addition, *location* node (node 2) is the parent of some *city* node (node 4) and an ancestor of the text '*Portage Ave.*' (node 5). *location* (node 3) can also be the parent of some *address* node (node 7).

The query corresponds to the following XPath expression:

hotel-room[*location*[*city* and //'500 Portage Ave.']]/*location*[*city* and *address*//'500 Portage Ave.'].

In Figure 3, there are two kinds of edges: child edges (*c*-edges) for parent-child relationships, and descendant edges (*d*-edges) for ancestor-descendant relationships. A *c*-edge from node *v* to node *u* is denoted by $v \rightarrow u$ in the text, and represented by a single arc; *u* is called a *c-child* of *v*. A *d*-edge is denoted $v \Rightarrow u$ in the text, and represented by a double arc; *u* is called a *d-child* of *v*. Such a query is often called a twig pattern. In addition, a node in *Q* can be a wildcard '*' that matches any type in *T*.

In any DAG (*directed acyclic graph*), a node *u* is said to be a descendant of a node *v* if there exists a path (sequence of edges) from *v* to *u*. In the case of a twig pattern, this path could consist of any sequence of *c*-edges and/or *d*-edges. Based on these concepts, the tree embedding can be defined as follows.

Definition 1. (*tree embedding*) An embedding of a tree pattern *Q* into an XML document *T* is a mapping f: $Q \rightarrow T$, from the nodes of *Q* to the nodes of *T*, which satisfies the following conditions:

(i) Preserve node label: For each $u \in Q$, *u* and $f(u)$ are of the same label (i.e., $label(u) = label(f(u))$).

(ii) Preserve *c*/*d*-child relationships: If $u \rightarrow v$ in *Q*, then $f(v)$ is a child of $f(u)$ in *T*; if $u \Rightarrow v$ in *Q*, then $f(v)$ is a descendant of $f(u)$ in *T*.

If there exists a mapping from *Q* into *T*, we say, *Q* can be imbedded into *T*, or say, *T* contains *Q*.

Notice that an embedding could map several nodes of the query (of the same type) to the same