USING TREE PATTERNS FOR FLEXIBLE HANDLING OF XML KEYS

By

Jing Wang

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
AT
MASSEY UNIVERSITY
PALMERSTON NORTH, NEW ZEALAND
DECEMBER 2007

Table of Contents

Abstract

Previous research shows that decision problem of XML keys is far more intricate than its relational counterpart. We review key constraints in the context of XML as introduced by Buneman et al [9, 10] and later refined by Hartmann et al [19, 20]. In this thesis we investigate three path expression languages for defining XML keys. They are path expression $\mathcal{PE}$, XPath expression $\mathcal{XE}$ and tree pattern $\mathcal{TP}$. We focus on a special tree pattern $\mathcal{TP}$ called *safe tree pattern*, denoted by $\mathcal{SP}$, which has been proved to be equivalent to XML keys where Q are in $\mathcal{XE}^{/,//}$,Q' are in $\mathcal{XP}^{/,//}$ and $P_1, \ldots, P_k$ are in $\mathcal{XE}^{/,*}$.

We propose a set of inference rules that is sound and complete for the implication of XML keys in the form of $\mathcal{SP}$. Our new containment result of $\mathcal{SP}$ is in PTIME. This result indicates a PTIME algorithm for deciding XML key implication, and shows that reasoning about XML key in $\mathcal{SP}$ is practically efficient.

Acknowledgements

I would like to express my gratitude to the many people who have made this thesis possible.

Most of all I would like to thank Professor Sven Hartmann, my supervisor, for his guidance, constant support and endless patience during this research.

Special thanks go to my colleges in the *(former)* Department of Information Systems at Massey University who have supported me throughout this year although it not possible to list all their names here individually.

Lastly, and most importantly, I am forever indebted to my family for their understanding and encouragement. I am grateful to my husband for his support. Without his this work would never have come into existence.

Jing Wang

December 1, 2007

Chapter 1

Introduction

1.1 Motivation

The World Wide Web Consortium promotes extensible Markup Language (XML) [8] and related standards, including XML Schema [14], XPath [5], XQuery [18], and XSLT [24]. XML is a very simple and flexible text format derived from SGML [13], which had been available for more than a decade without really influencing the development of Web applications. Originally, XML was designed to meet the challenges of large-scale electronic publishing, Nowadays, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. XML is emerging as a data model in logical database design, also the XML documents are to serve double duty as databases. XML is self-explanatory and easily parsed. However, despite the high degree of syntactic flexibility, the formal semantic of the XML and related standards are not well developed. We need to define and prove theory that supports XML data modelling, data specification and data management. Various work have address the semantic of XML databases. Naturally

the study of integrity constraints has received increased attentions from the database research community because integrity constraints are an essential part of database scheme definition languages. They play fundamental roles of semantic specification, data validation, integrity control, query optimisation, and etc. Various work have attempted to define different classes of integrity constraints. Therefore, some classes of integrity constraints including keys have been made available for XML [9, 10, 20, 19].

Keys are of importance in databases. They are an integral part of database practice and theory in the context of the relational database model. In particular, keys are used to identify object within the database and reference one object from another. Keys and referential integrity constraints constitute an essential class of constraints on the validation of data. Since the same key constraints are required for XML databases, there is a need to investigate issues like XML key specifications and related issues such as satisfiability and implication problem. Intuitively, researchers attempt to apply the some approaches which have been used in the context of relation database model to XML data model. However, due to hierarchical structure of XML data, the problems of XML key definition and associated satisfiability and implication problems are much more complicated and harder than in the RDM. The *ID/IDREF* in XML DTD is too weak in term of expressive power. On the other hand side, the *key and keyref* in the XML Schema using full XPath expression is too complicate for reasoning about. There is a need to find a new way to define XML keys which are powerful and expressive enough and also still can be reasoned about efficiently. This is the motivation of our work.

Various approaches for XML keys have been proposed, e,g, [9, 10, 20, 19]. Independently from any specifications such a document type definition (DTD) or XML

Schema definition (XSD), these keys are based on the representation of XML data as trees and have been defined in term of path expressions. Their associated decision problems such as satisfiability and implication are of our research interests.

1.2 Objectives

The main goal of this thesis is to investigate key implication problem by using XML tree pattern. In this thesis we continue the study of path-expression-based key specification presented in [9, 10, 20, 19]. We review XML key constraints and observe that in order to define XML key constraints the choice of a path language can be crucial. Generally speaking, We need a suitable path language to select nodes in XML which is expressive enough to allow reasonable navigation and simple enough to allow efficient reasoning.

- The first part of our work is to examine different approaches of expressing node selection queries, study translation and discover equivalences between them.

- In the second part of our work, we focus on tree patterns as a convenient way to express paths, thus, to define XML keys. Our focus is on defining a *safe tree pattern* and reasoning about the containment problem and eventually investigating its computational complexity.

- Last, in accordance with other work relating to XML keys, we are interested in a generalised XML key in the form of the *safe tree pattern*. We provide a set of inference rules for the implication of XML keys in the *safe tree pattern*. We need to prove that their (finite) implication problem is finitely axiomatisable. Consequently we can then apply the result of the containment problem of *safe*

tree pattern to the key implication problem. The time complexity of the *safe tree pattern* containment is in PTIME, hence, the implication problem can be decided in PTIME as well. Therefore, we show that XML keys defined in the form the *safe tree pattern* can be maintained efficiently by database systems for XML applications.

1.3 Outline of the Thesis

The reminder of this thesis is organised as follows: In Chapter 2 we present some preliminary definitions used throughout this thesis, illustrate the lack of effective and efficient integrity constraints in the current XML scheme specification languages, namely, DTD and XSD and at the end of this chapter we present a running example used in this thesis. In the next two chapters three path languages, namely, Path expression $\mathcal{PE}$, XPath expression $\mathcal{XE}$ and tree pattern $\mathcal{TP}$, have been examined for selecting nodes in XML trees. We limit our study of query languages to *downward* queries only. Upward paths, for example, *parent*, *ancestor* or *sibling* paths, are not relevant to our study and we leave them out. We prove semantic equivalence between the query languages and show how to translate from one to another. The query containment problem is our main interest of study. Chapter 5 we present our result of using *safe* tree pattern handling XML keys. In Chapter 6 we conclude the work. Related work is discussed in Chapter 6.1 and finally we outline some possible research directions in Chapter 6.3