Towards a Normal Form and a Query Language for Extended Relations Defined by Regular Expressions

András Benczúr, Eötvös Loránd University, Faculty of Informatics, Budapest, Hungary

Gyula I. Szabó, Eötvös Loránd University, Faculty of Informatics, Budapest, Hungary

ABSTRACT

This paper introduces a generalized data base concept that unites relational and semi structured data models. As an important theoretical result we could find a quadratic decision algorithm for the implication problem of functional and join dependencies defined on the united data model. As practical contribution we presented a normal form for the new data model as a tool for data base design. With our novel representations of regular expressions, a more effective searching method could be developed. XML elements are described by XML schema languages such as a DTD or an XML Schema definition. The instances of these elements are semi-structured tuples. A semi-structured tuple is an ordered list of (attribute: value) pairs. We may think of a semi-structured tuple as a sentence of a formal language, where the values are the terminal symbols and the attribute names are the non-terminal symbols. In the authors' former work (Szabó and Benczúr, 2015) they introduced the notion of the extended tuple as a sentence from a regular language generated by a grammar where the non-terminal symbols of the grammar are the attribute names of the tuple. Sets of extended tuples are the extended relations. The authors then introduced the dual language, which generates the tuple types allowed to occur in extended relations. They defined functional dependencies (regular FD - RFD) over extended relations. In this paper they rephrase the RFD concept by directly using regular expressions over attribute names to define extended tuples. By the help of a special vertex labeled graph associated to regular expressions the specification of substring selection for the projection operation can be defined. The normalization for regular schemas is more complex than it is in the relational model, because the schema of an extended relation can contain an infinite number of tuple types. However, the authors can define selection, projection and join operations on extended relations too, so a lossless-join decomposition can be performed. They extended their previous model to deal with XML schema indicators too, e.g., with numerical constraints. They added line and set constructors too, in order to extend their model with more general projection and selection operators. This model establishes a query language with table join functionality for collected XML element data.

KEYWORDS

Functional Dependencies, Join Dependency, Normalization, Regular Language, XML

INTRODUCTION

XML has evolved to become the de-facto standard format for data exchange over the World Wide Web. XML was originally developed to describe and present individual documents, it has also been used to build databases. Our original motivation for the introduction of the regular relational data model (Szabó and Benczúr, 2015) was to find a good representation of the XML ELEMENT type

DOI: 10.4018/JDM.2016040102

declaration for substructure specification. The instances of a given element type in an XML document can be considered as a collection of data of complex row types. The set of attribute names in the row types are the element names occurring in the DTD declaration of the element. In the case of recursive regular expression in the element declaration, there are possibly infinite number of different row types for the element instances. The same attribute name may occur several times in a type instance. This leads to the problem of finding a formal way to define the projection operator, similar to the relational algebra, on the syntactical structure of the data type.

That is necessary to define the left and right side of a functional dependency. We defined the attribute sequence by a traversal on the vertex labeled graph associated to the regular expression of the DTD.

This form is also good to define attribute subsequences for the projection operator, for the selection operator and for equijoin operator. Set operations can be extended in a straightforward way, so this leads to the full extension of relational algebra operators. Using the extension of projection and equijoin (or natural join) the join dependency can be defined in the same way as in the relational model.

Motivation: Our previous model (Szabó and Benczúr, 2015) could be effectively used for handling functional dependencies (FD). In the relational model FDs offer the basis for normalization (e.g. BCNF), to build non-redundant, well-defined database schema. But our former model cannot handle the join operation among instances (that is used to support lossless join decomposition) because the projection of a schema according to a set of nodes or two joined schemas would not necessarily lead to a new, valid schema. We need an improved model for regular data bases. To denote a regular language, we can use regular expressions, our actual model bases upon a graph representation for regular expressions. This model is more redundant than our last one, but it is capable for handling database schema normalization.

Contributions: The main contribution of this paper is the concept of extended relations over the graph representation for regular expressions. We rephrase regular functional dependencies and also define regular join dependencies that constrain extended relations. We determine the schema of an extended relation as *(IN, ..., OUT)* traversals on the graph representation for a given regular expression. We apply the classical Chase algorithm to a counterexample built on this graph. In this way, we show that the logical implication is decidable for this class of functional and join dependencies. An extended abstract of this paper (Benczúr and Szabó, 2014) appeared in the Proceedings of the 18th East-European Conference on Advances in Databases and Information Systems (ADBIS 2014).

RELATED WORK

As far as we know, each XML functional dependency (XFD) concept involves regular expressions or regular languages. Arenas and Libkin (2004) prove different complexities for logical implication concerning their tree tuples XFD model according to the involved regular expressions. They prove quadratic time complexity in case of simple regular expressions. Our new model represents all possible instances of the regular expression at the same time and so it differs from theirs. Liu, Tan, and Chen (2013) proposed an approach to automatically extract attribute dependencies from a database application. They graph-based method has similarity to ours.

The notion data words have been introduced by Bouyer et al. (2003), based upon finite automata of Kaminski et al. (1994). Data words are pairs of a letter from a finite alphabet and a data from an infinite domain. Our concept differs substantially from data words: we assign data values (selected from infinite domains) to letters (from a finite alphabet) after generating a sentence by a regular expression. For data words letters and data values are processed together. Libkin and Vrgoč (2012) define regular expression for data words. They analyze the complexity of the main decision problems (non-emptiness, membership) for these regular expressions. Their model is similar to ours but our point

Related Content