Extended Relations Defined by Regular Expressions

András Benczúr[a], Gyula István Szabó[a]

[a] Eötvös Loránd University Budapest, Information Systems Department 1117 Budapest, Pázmány Péter sétány 1/C, Hungary
abenczur@inf.elte.hu
gyula@szaboo.de

Abstract

In our [1] paper, we rephrased the notion of extended tuple as a sentence from a regular language generated by grammar G where the nonterminals of the grammar are the attribute names of the tuple. Finite sets of extended tuples play the role of extended relation instances. Then we introduce the dual language, which generates the accepted tuple-types of the extended relation. This presentation discusses another possible specification of the regularly extended relational data model, namely, it directly uses regular expression on attribute names to define relational schema instances. This approach is closer to the style of element declaration in XML DTD or XML Schema.
Definition of the regular relational schema and instances:
Let U be the set of attribute names, and RS be a regular expression over U. The language defined by RS is denoted by L(RS). A sentence W from L(RS) specifies a regular tuple type over U. We may associate specific domains for each attribute from U. A table instance of type W consists of a finite collection of tuples of type W. A regular relational instance over L(RS) is a finite set of table instances of tuple types from L(RS).
A regular database is a finite set of regular relation schemas, with their instances.
The basic operations of relational algebra are extended to regular relations. The complexity in defining the operation is in that we have to define first the effect on finite set of tuple types, then on the corresponding table instances. We give also the extension of functional dependencies and join dependencies. The main technique uses a special vertex-labeled graph representation of the regular expression RS. Projections, selections and joins are based on subsequence selection from an attribute sequence $W \in L(RS)$.

Keywords: relational databases, regular languages, XML data types, functional and join dependencies .

*MSC:*94-06

References

[1] SZABÓ, G. I, BENCZÚR, A., Functional Dependencies on Extended Relations Defined by Regular Languages. *Annals of Mathematics and Artificial Intelligence,* DOI: 10.1007/s10472-013-9352-z

[2] SZABÓ, G. I, BENCZÚR, A., Functional Dependencies on Extended Relations Defined by Regular Languages. Proceedings of Foundations of Information and Knowledge Systems, Kiel, Germany, March 2012, Lecture Notes in Computer Science, 7153 (eds: T. Lukasiewicz, A. Sali), Springer, p. 385–404