

Chapter XV

On the Computation of Recursion in Relational Databases

Yangjun Chen*
University of Winnipeg, Canada

ABSTRACT

A composite object represented as a directed graph is an important data structure which requires efficient support in CAD/CAM, CASE, office systems, software management, Web databases and document databases. It is cumbersome to handle such an object in relational database systems when it involves recursive relationships. In this chapter, we present a new encoding method to support the efficient computation of recursion. In addition, we devise a linear time algorithm to identify a sequence of reachable trees (w.r.t.) a directed acyclic graph (DAG), which covers all the edges of the graph. Together with the new encoding method, this algorithm enables us to compute recursion w.r.t. a DAG in time $O(e)$, where e represents the number of edges of the DAG. More importantly, this method is especially suitable for a relational environment.

** The author is supported by NSERC 239074-01 (242523) (Natural Science and Engineering Council of Canada)*

INTRODUCTION

It is a general opinion that relational database systems are inadequate for manipulating composite objects which arise in novel applications such as Web and document databases (Mendelzon, Mihaila & Milo, 1997; Abiteboul et al., 1997; Chen & Aberer, 1998, 1999), CAD/CAM, CASE, office systems and software management (Banerjee et al., 1988; Teuhola, 1996). Especially when recursive relationships are involved, it is cumbersome to handle them in a relational system. To mitigate this problem to some extent, many methods have been proposed, such as *join index* (Valduriez & Borel, 1986) and *clustering of composition hierarchies* (Haskin & Lorie, 1982), as well as the encoding scheme (Teuhola, 1996).

In this chapter, we present a new encoding method to facilitate the computation of recursive relationships of nodes in a DAG. In comparison with Teuhola's, our method is simple and space-economical. Specifically, the problem of Teuhola's, the so-called *signature conflicts*, is removed.

BACKGROUND

A composite object can be generally represented as a directed graph. For example, in a CAD database, a composite object corresponds to a complex design, which is composed of several subdesigns (Banerjee et al., 1988). Often, subdesigns are shared by more than one higher-level design, and a set of design hierarchies thus forms a directed acyclic graph (DAG). As another example, the citation index of scientific literature, recording reference relationships between authors, constructs a directed cyclic graph. As a third example, we consider the traditional organization of a company, with a variable number of manager-subordinate levels, which can be represented as a tree hierarchy. In a relational system, composite objects must be fragmented across many relations, requiring joins to gather all the parts. A typical approach to improving join efficiency is to equip relations with hidden pointer fields for coupling the tuples to be joined (Carey et al., 1990). Recently, a new method was proposed by Teuhola (1996), in which the information of the ancestor path of each node is packed into a fix-length code, called the *signature*. Then, the operation to find the transitive closure w.r.t. a directed graph can be performed by identifying a series of signature intervals. No joins are needed. Using Teuhola's method, CPU time can be improved up to 93% for trees and 45% for DAGs in comparison with a method which performs a SELECT command against each node, where the relation to store edges is equipped with a clustering index on the parent nodes (Teuhola, 1996).

In this chapter, we follow the method proposed in Teuhola (1996), but using a different encoding approach to pack "ancestor paths." For example, in a tree hierarchy, we associate each node v with a pair of integers (α, β) such that if v' , another node associated with (α', β') , is a descendant of v , some arithmetical relationship between α and α' as well as β and β' can be determined. Then, such relationships can be used to find all descendants of a node, and the recursive closure w.r.t. a tree can be computed very efficiently. This method can be generalized to a DAG or a directed graph containing cycles by decomposing a graph into a sequence of trees (forests), in which the approach described above can be employed. As we will see later, a new method can be developed based on the techniques mentioned above, by which recursion can be evaluated in $O(e)$

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/computation-recursion-relational-databases/9215

Related Content

Beyond OpenURL: Technologies for Linking Library Resources

George Boston and Randle J. Gedeon (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 1405-1419).

www.irma-international.org/chapter/beyond-openurl-technologies-linking-library/7981

Knowledge Representation: A Conceptual Modeling Approach

Cecil Eng Huang Chua, Veda C. Storey and Roger H. Chiang (2012). *Journal of Database Management* (pp. 1-30).

www.irma-international.org/article/knowledge-representation-conceptual-modeling-approach/62030

Countermeasures for Protecting Legally Sensitive Web-Powered Databases and Web Portals

Theodoros Evdoridis and Theodoros Tzouramanis (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 2079-2084).

www.irma-international.org/chapter/countermeasures-protecting-legally-sensitive-web/8021

User Resistance to Software Migration: The Case on Linux

Hee-Woong Kim, Hock Chuan Chan and So-Hyun Lee (2014). *Journal of Database Management* (pp. 59-79).

www.irma-international.org/article/user-resistance-to-software-migration/109932

Text Detection Model for Historical Documents Using CNN and MSER

Rankang Li, Shanxiong Chen, Fujia Zhao and Xiaogang Qiu (2023). *Journal of Database Management* (pp. 1-23).

www.irma-international.org/article/text-detection-model-for-historical-documents-using-cnn-and-mser/322086