Compile-Time Path Expansion in Lore

Jason McHugh, Jennifer Widom
Stanford University
{mchughj,widom}@db.stanford.edu, http://www-db.stanford.edu

Abstract

Semistructured data usually is modeled as labeled directed graphs, and query languages are based on declarative *path expressions* that specify traversals through the graphs. *Regular* (or *generalized*) path expressions use regular expression operators to specify traversal patterns. Regular path expressions typically are evaluated at run-time by exploring the database graph. However, if the database includes a structural summary such as a *DataGuide*, then an alternative approach is to expand regular path expressions at compile-time using the structural summary, reducing the run-time overhead of database exploration. This paper describes algorithms for compile-time regular path expression expansion in the context of the *Lorel* query language for semistructured data, and reports on performance results conducted on the *Lore* system illustrating the benefits of compile-time expansion.

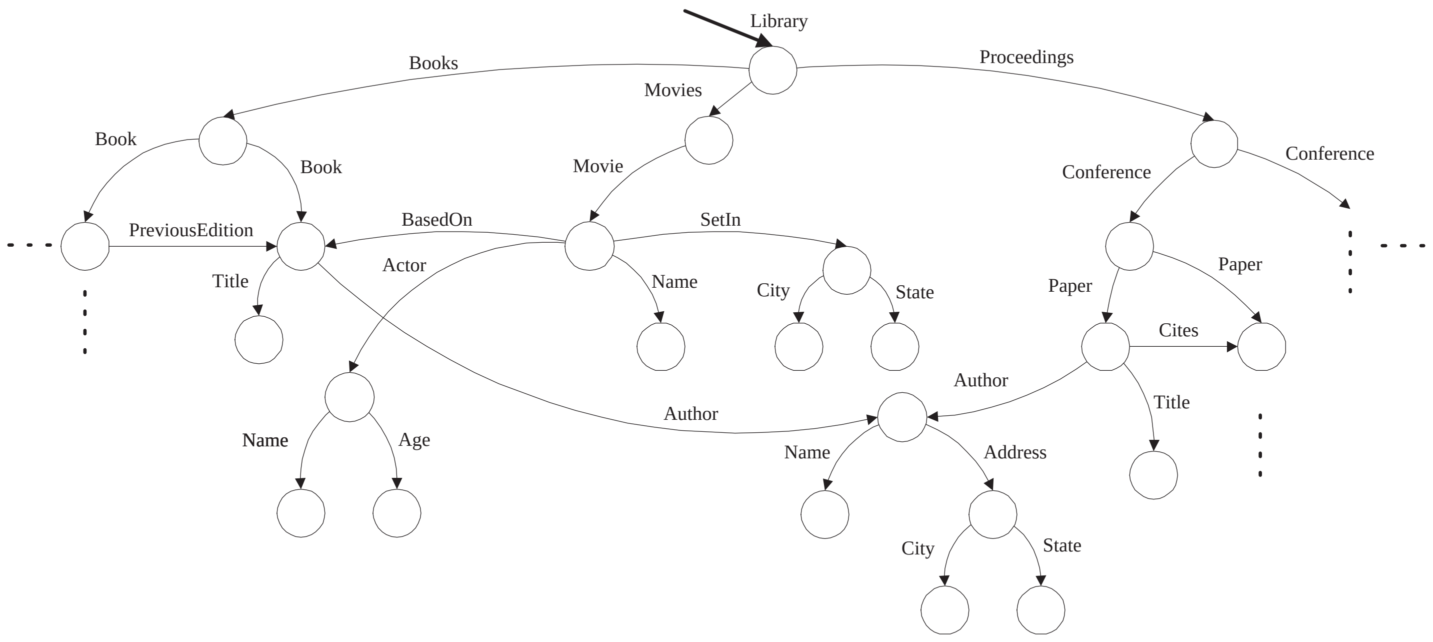
1 Introduction

Efficiently storing and querying *semistructured* data—data that need not adhere to a fixed schema—has emerged as an important topic in the database research community [Abi97, Bun97, Suc98]. Researchers have addressed the design of query languages for semistructured data, e.g., [AQM$^+$97, BDHS96, FFLS97], the implementation of prototype database management systems for semistructured data, e.g., [FFLS97, MAG$^+$97], and query optimization issues related to semistructured data and path expressions, e.g., [BDHS96, FLS98, FS98, GGT96, MW98]. With the recent emergence of *XML*, a proposed standard for exchanging information on the Web [LB97], and the remarkable similarity of XML to typical models for semistructured data, support for query languages for semistructured data—and the performance of such queries over large semistructured databases—is of increasing importance.

XML, as well as other semistructured data, can be stored in a database system by modeling the data as a labeled directed graph. Queries use *path expressions* to describe traversals through the labeled edges in the graph. The efficient evaluation of a query then hinges upon choosing efficient traversal schemes for the set of path expressions in the query [MW98]. A *regular* (or *generalized*) path expression uses regular expression operators to specify a (possibly recursive) path *pattern*. Regular path expressions are particularly useful when the structure of the data is irregular, changes often, or is not completely known to the user. Modeling semistructured data as labeled graphs and querying it using a language based on regular path expressions is common to most research in semistructured data, e.g., [BDHS96, FFLS97, MAG$^+$97].

Run-time evaluation of regular path expressions can be expensive. In this paper we explore improving efficiency by performing compile-time expansion of regular path expressions based on a structural summary (*DataGuide* [GW97]) of the current database. Compile-time expansion incurs the cost of exploring the structural summary and rewriting the query, but it can eliminate significant amounts of unnecessary database exploration at run-time. We have implemented our algorithms in the *Lore* DBMS for semistructured data [MAG$^+$97], and preliminary performance results confirming the benefits of the approach are reported.

Our work is similar in spirit, but not in details, to [FS98]. In [FS98], a cross-product is computed between a *graph schema*—a summary of the database that must be small and reside in memory—and a representation of the query. From this cross-product an expanded version of the query is


Library
Books
Movies
Proceedings
Book
Book
Movie
Conference
Conference
PreviousEdition
BasedOn
SetIn
Actor
Title
Name
City
State
Paper
Paper
Cites
Author
Author
Title
Name
Age
Name
Address
City
State