Indexing Structure for Graph-Structured Data

Stanislav Bartoň and Pavel Zezula

Faculty of Informatics, Masaryk University
Botanická 68a, 60200 Brno
Czech Republic
{barton, zezula}@fi.muni.cz

Abstract. An own design of an indexing structure for general graph structured data called ρ-index that allows an effective processing of special path queries is presented. These special queries represent for example a search for all paths lying between two arbitrary vertices limited to a certain path length. The ρ-index is a multilevel balanced tree structure where each node is created with a certain graph transformation and described by modified adjacency matrix. Hence, ρ-index indexes all the paths to a predefined length l inclusive. The search algorithm is then able to find all the paths shorter than or having the length l and some of the paths longer then the predefined l lying between any two vertices in the indexed graph. The designed search algorithm exploits a special graph structure, a transcription graph, to compute the result using the ρ-index. We also present an experimental evaluation of the process of creating the ρ-index on graphs with different sizes and also a complexity evaluation of the search algorithm that uses the ρ-index.

10.1 Introduction

In the context of the Semantic Web, ρ-operators are proposed in [5] as a mean to explore complex relationships [20] between entities. The problem of searching for the complex relationships can be modeled as the process of searching paths in a graph where various entities represent vertices and edges the direct relationships between them. In case of the semantic web the resources or classes and edges the properties between them. The notion of complex relationships can be also identified in bibliographic digital libraries, where entities are publications and the relationship can represent references or direct citations between them.

As proposed in [5], we recognize two kinds of complex relationships. The first one is represented by a path lying between two inspected vertices. Speaking in terms of publications this means that one publication indirectly cites or references the other publication – a chain of publications can be built so that one cites another. The second type of complex relationship is a connection between two inspected vertices. This symbolizes a fact that the two inspected publications indirectly cite one common publication, see Figure 10.1 for an example of this kind of complex relationship.
The knowledge about complex relationships among publications can be used for example for ranking the result of the search for publications using the complex relationship discovery among entities present in the result and then sorting them according to that information. Another use case can be an automated recommendation of publications based on the preferred set of publications by searching for close connections between the publications from the preferred set. Intuitively, the complex relationship discovery has sense in any other field of interest that incorporates graph structured data. For that reason, this chapter introduces an indexing technique called the $\rho$-index that enables efficient discovery of all complex relationships between any two inspected entities in large collections of arbitrary graph structured data.

This chapter is then structured as follows, Section 10.2 presents related work in the field of indexing graph structured data, Section 10.3 is a brief insight into the design of the proposed indexing structure. Section 10.4 introduces a search algorithm that is used to discover all paths between any two vertices in the indexed graph using $\rho$-index. Consequently, the experimental evaluation of the designed indexing structure and the search algorithm is in Section 10.5. Finally, this chapter is concluded and some directions of the future work are proposed in Section 10.6.

10.2 Related Work

The problem of answering various graph queries has two possible solutions. One is through an algorithmic on the fly query answering and the other one is preprocessing some indexing structure that would ease the computational complexity of the query processing.

Firstly we discuss one of the on the fly algorithmic approaches which is Tarjan’s algorithmic solution to a single source path expression problem from [18][19]