SCALABLE REACHABILITY INDEXING FOR VERY LARGE GRAPHS

By

Hilmi Yildirim

An Abstract of a Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Mohammed J. Zaki, Thesis Adviser
Mark Goldberg, Member
Malik Magdon-Ismail, Member
William Wallace, Member
Elliot Anshelevich, Member

Rensselaer Polytechnic Institute
Troy, New York

August 2011
(For Graduation December 2011)

ABSTRACT

Answering reachability queries in graphs is an important problem. With the development of high-throughput data acquisition techniques and the advances in the areas of semantic web and social networks, we have abundance of enormous graph-structured data on which different queries are asked. One of the fundamental queries, a reachability query, asks whether there exists a path between any two given nodes. This can map to the question of whether one researcher has been influenced by another in a citation network; whether a protein inhibits or activates another one indirectly in a protein interaction network; whether a protein is broken down to a specific molecule in a metabolic pathway graphs; or whether a concept is subsumed by part of another in an ontology. Aside from these direct correspondences with real-life questions, they can constitute building blocks for complicated queries in various databases. Therefore, there is a crucial need for mechanisms that expedite querying in graph databases.

Existing methods for reachability trade-off indexing time and space versus query time performance. However, the biggest limitation of existing methods is that they do not scale to very large real-world graphs. They are also vulnerable to increasing edge densities. Another limitation of the existing methods is that they barely, if at all, support dynamic updates. This is primarily due to the complex nature of the problem a single edge addition or deletion can potentially affect the reachability of all pairs of nodes in the graph. Most of the previous work has focused on dynamically maintaining the transitive closure of a graph, which has the obvious $O(n^2)$ worst-case bound, where n is the number of nodes. Moreover, most of the static indexes cannot be directly generalized to the dynamic case. This is because these indexes trade-off the computationally intensive preprocessing/index construction stage to minimize the index size and querying time. For dynamic graphs, the efficiency of the update operations is another aspect which needs to be optimized. However, the costly index construction typically precludes fast updates. It is interesting to note that a simple approach consisting of depth-first search (DFS)

can handle graph updates in $O(1)$ time and queries in $O(n+m)$ time, where m is the number of edges. For sparse graphs $m = O(n)$ so that query time is $O(n)$ for most large real-world graphs. Any dynamic index will be effective only if it can amortize the update costs over many reachability queries.

In this thesis, we present two approaches for addressing the problems of scalable reachability indexing for both static and dynamic graphs. More specifically, we introduce two indexing schemes, namely GRAIL and DAGGER. GRAIL is a simple yet scalable reachability index that is based on the idea of randomized interval labeling, and that can effectively handle very large graphs. Based on an extensive set of experiments, we show that while more sophisticated methods work better on small graphs, GRAIL is the only index that can scale to millions of nodes and edges. GRAIL has linear indexing time and space, and the query time ranges from constant time to being linear in the graph order and size.

Our second contribution is a scalable, light-weight reachability index for dynamic graphs called DAGGER which has linear (in the order of the graph) index size and index construction time, and reasonably fast update and query times. DAGGER is based on the idea of maintaining randomized interval labels for the nodes of the underlying acyclic graph (DAG) of the input graph. Therefore DAGGER yields an efficient algorithm for maintaining the strongly connected components of the evolving graph, which is of independent interest. We demonstrate the efficiency and effectiveness of DAGGER in large dynamic real-world networks such as Wikipedia graph and citation networks as well as synthetic dynamic graphs.

In the future, we plan to improve the query time of DAGGER by maximizing the quality of the index while keeping the updates fast. We also plan to extend GRAIL and DAGGER for other variants of reachability problem such as constrained reachability and shortest path queries.