Pattern Matching in Directed Graphs

Jianghai Fu

Department of Computer Science, University of Waterloo
Waterloo, ON, Canada, N2L 3G1
Email: jfu@neumann.uwaterloo.ca

Abstract. Pattern matching in directed graphs is a natural extension of pattern matching in trees and has many applications to different areas. In this paper, we study several pattern matching problems in ordered labeled directed graphs. For the rooted directed graph pattern matching problem, we present an efficient algorithm which runs in time and space $O(|E(P)| \times |V(T)| + |E(T)|)$, where $|E(P)|$, $|V(T)|$ and $|E(T)|$ are the number of edges in the pattern graph $P$, the number of nodes in the target graph $T$ and the number of edges in the pattern graph $T$, respectively. It is by far the fastest algorithm for this problem. This algorithm can also solve the directed graph pattern matching problem without increasing time or space complexity. Our solution to this problem outperforms the best existing method by Katzenelson, Pinter and Schenfeld by a factor of at least $|V(P)|$. We also present an algorithm for the directed graph topological embedding problem which runs in time $O(|V(P)| \times |E(T)| + |E(P)|)$ and space $O(|V(P)| \times |V(T)| + |E(P)| + |E(T)|)$, where $|V(P)|$ is the number of nodes in the pattern graph $P$. To our knowledge, this algorithm is the first one for this problem.

1 Introduction

Pattern matching in trees has been successful in a number of application areas. However, because of the lack of mechanism in trees to express recursive structures, trees are not suitable for representing some complex objects in which explicit expressions of recursive relations are useful. As a consequence, there has been increasing demand in recent years that pattern matching be extended to more general graphs [2, 3, 5, 6, 7, 10, 11, 12].

A directed graph is a natural choice for expressing recursive relations. Pattern matching in directed graphs has been used by several research groups for different purposes in different areas. One example is Katzenelson, Pinter and Schenfeld’s type-checking system [11] in which type expressions are represented by type-graphs which are ordered labeled rooted directed graphs, where a rooted directed graph is a directed graph in which there is a single node designated as the root from which there is at least one edge emanating. A key step in their system is to identify all equivalent subgraphs in a type-graph so that redundant parts in the type-graph can be eliminated. Using the technique of pattern matching, Katzenelson, Pinter and Schenfeld [11] describe a method which can identify all equivalent subgraphs in a type-graph $G$ in time $O(|V(G)|^2 \times |E(G)|)$. 
Pattern matching in directed graphs is also used in Holm's system, in which graph concepts are used in semantics descriptions of functional languages [10]. In Holm's system, recursively typed languages are represented by directed graphs. The task of pattern matching can be described as follows: the patterns describe a class of objects and are represented using directed graphs. When checking a recursive type, the recursive type value is matched against the patterns, i.e., the algorithm checks if the type value is an instance of a pattern. The matching process is implemented in a simple graph reduction machine which supports primitive graph operations and provides a base for language implementations.

Directed graphs are well suited for representing regular tree expressions [2]. Aiken and Murphy [2] discuss the implementation of regular tree expressions which are used in type inference and program analysis algorithms [3, 8]. The most commonly used operation among all fundamental operations in the implementation is the operation for testing inclusion relations which can be implemented using a pattern matching technique.

Pattern matching is a crucial component of term rewriting systems. Directed graphs are well suited for expressing the infinite terms with a finite number of distinct subterms in a cyclic term graph rewriting system [5, 6]. Motivated by the need for providing satisfactory interpretation for cyclic term graph rewriting, Corradini [5] discuss the extension of the classical theory of term rewriting systems to infinite and partial terms. A theoretical treatment of pattern matching in directed graphs has an important impact on the research of infinite term rewriting systems.

Although pattern matching in directed graphs has been used by individual research groups, a thorough study of the problems has not been carried out. This paper is a step towards an extensive investigation of various pattern matching problems in directed graphs and the development of techniques for these problems. Efficient solutions to these problems not only are of theoretical interest in their own right, but also improve the performance of many systems such as listed above.

In this paper, we first define in Section 2 the notation and terminology that are used throughout this paper and the problems with which we are concerned. Then in Section 3, we review the previous results for some problems. In Sections 4.1 and 4.2 we present an efficient algorithm for the ordered labeled rooted directed graph pattern matching problem which runs in time and space \( O(|E(P)| \times |V(T)| + |E(T)|) \). Extension of this algorithm to the ordered labeled directed graph pattern matching problem is also discussed. In Section 5 we present an efficient algorithm for the ordered labeled directed graph topological embedding problem which runs in time \( O(|V(P)| \times |E(T)| + |E(P)| + |E(T)|) \) and space \( O(|V(P)| \times |V(T)| + |E(P)| + |E(T)|) \). Our solution to the ordered labeled directed graph pattern matching problem is faster than the best method obtained by adapting Katzenelson, Pinter and Schenfeld's type equivalence testing algorithm [11] by a factor of at least \(|V(P)|\). Section 6 gives the conclusion.