Chapter 14
A Backtracking Algorithmic Toolbox for Solving the Subgraph Isomorphism Problem

Jurij Mihelič
Faculty of Computer and Information Science, University of Ljubljana, Slovenia

Uroš Čibej
Faculty of Computer and Information Science, University of Ljubljana, Slovenia

Luka Fürst
Faculty of Computer and Information Science, University of Ljubljana, Slovenia

ABSTRACT

The subgraph isomorphism problem asks whether a given graph is a subgraph of another graph. It is one of the most general NP-complete problems since many other problems (e.g., Hamiltonian cycle, clique, independent set, etc.) have a natural reduction to subgraph isomorphism. Furthermore, there is a variety of practical applications where graph pattern matching is the core problem. Developing efficient algorithms and solvers for this problem thus enables good solutions to a variety of different practical problems. In this chapter, the authors present and experimentally explore various algorithmic refinements and code optimizations for improving the performance of subgraph isomorphism solvers. In particular, they focus on algorithms that are based on the backtracking approach and constraint satisfaction programming. They gather experiences from many state-of-the-art algorithms as well as from their engagement in this field. Lessons learned from engineering such a solver can be utilized in many other fields where backtracking is a prominent approach for solving a particular problem.

DOI: 10.4018/978-1-7998-7156-9.ch014

INTRODUCTION

Collecting, organizing, and analysing large amounts of data has emerged as a promising new area of research. Such data often require a structural description, for which graphs are commonly used. Graphs are an essential form of data organization and are emerging in the most surprising applications from chemistry (Agrafiotis et al., 2011; Balaban, 1985; Barnard, 1993), neurology (He & Evans, 2010), social sciences (Carrington, Scott, & Wasserman, 2005), linguistics (Krahmer, Van Erk, & Verleg, 2003), social network analysis (Fan, 2012), and computer vision (Liu & Lee, 2001). Moreover, in these domains graph databases are replacing the traditional relational databases (Batra & Tyagi, 2012). For an extensive review of graph matching for pattern recognition see (Conte, Foggia, Sansone, & Vento, 2004).

From the viewpoint of pattern analysis and recognition, matching graphs and subgraphs is one of the most important tasks in graph processing. The problem of finding instances of a given graph in a larger graph is called *subgraph isomorphism problem*. Given two graphs, namely, a pattern graph and a target graph, the problem is defined as finding a subgraph (corresponding to the pattern graph) in the target graph. Due to the ubiquity of this problem, it has been studied extensively, both from a purely theoretical point of view as well as from a more practical point of view.

A plethora of algorithms for solving the problem exactly exist in the literature. The most widely used and well-known subgraph isomorphism algorithms are Ullmann's algorithm (Ullmann, 1976; Čibej & Mihelič, 2015), VF2 (Cordella, Foggia, Sansone, & Vento, 1999), VF3 (Carletti, Foggia, Saggese, & Vento, 2017), RI (Bonnici, Giugno, Pulvirenti, Shasha, & Ferro, 2013a), Glasgow subgraph solver (McCreesh & Prosser, 2015a), SubSea (Lipets, Vanetik, & Gudes, 2009), FocusSearch (Ullmann, 2010), LAD (Solnon, 2010). Most of them are based on the *backtracking* approach which explores the search tree of possible solutions. As the problem is NP-hard, all such algorithms exhibit exponential running time in the worst case. Nevertheless, in many practical cases, the problem can be quite efficiently solved if advanced techniques of pruning the search tree are employed.

In this paper, we focus on such techniques (intended for backtracking algorithms) which may provide a significant performance boost to algorithms especially if used in combination with others. The techniques are mostly based on heuristic approaches as well as on optimization and tuning of algorithms. They often originate from the field of constraint satisfaction problems and represent a basic approach in the field of algorithm engineering (Muller-Hannemann & Schirra, 2010; McGeoch, 2012). Our approach is systematic (Blagojević et al., 2016) and we strive to give an overview of different issues encountered when developing and engineering efficient backtracking algorithms. The collected experiences can serve as a toolbox for further improvement of algorithms for the subgraph isomorphism problem, as well as guidelines on how to solve similar problems.

In the rest of the chapter, we first formally define the subgraph isomorphism problem and present the main framework of the backtracking approach to solving it. Afterward, we present the backtracking refinement techniques for reduction of both time and space complexity. In the last part, we focus on various search orders, which are one of the most important speedup factors. For each technique, we give a description followed by its experimental evaluation.