Core algorithms to search in biological structured data

V. Bonnici[1], R. Giugno[2]✉, A. Pulvirenti[2], D. Shasha[3], A. Ferro[2]

[1]Dept. Computer Science, University of Verona, Italy
[2]Dept. Clinical and Molecular Biomedicine, University of Catania, Italy
[3]Courant Institute of Mathematical Sciences, New York University, United States

Motivations

The graph is a data structure to represent biological data ranging from molecules and proteins to biological networks and metabolic pathways. Working on those data involves manly applying graph isomorphism algorithms. Those algorithms are computationally hard and their efficiency may depend upon the input graphs. We are building a library, SubGraphLib, of the most popular searching algorithms and benchmarks highlighting drawbacks, advantages, and best performance input cases for each method. A novel approach to find all occurrences of a query subgraph in a target graph is also proposed. This new method applies a search strategy which significantly reduces the search space without using any complex pruning rule. Results show a significant reduction of the running time with respect to other methods together with a scalable memory requirement.

Methods

The best known algorithms to solve the subgraph isomorphism problem are the ones proposed by Ullmann [1] and by Cordella et al. [2] (VF2), which make use of backtracking algorithms in conjunction with some filtering rules to prune branches of the search space represented as a tree. The nodes of the tree denote pairs of matched vertices of the query and the target graphs, respectively. During the visit, the isomorphism conditions are applied to verify the partial matches. The algorithm in [1] modeled the graph isomorphism problem also as a constraint satisfaction problem (CSP). A CSP is defined by a set of variables and a set of constraints among them. To each variable a set of possible values, called domain, is associated. The solution of a given CSP problem is an assignment of values to all variables such that all constraints are satisfied. More recently, Solnon [3] published a method, LAD, for propagating global neighborhood constraints together with a generalized arc consistency. Ullmann [4] proposed a new method, FocusSearch, based on bitvector representation of domains, to deal with parallel operations. In FocusSearch, domain reduction is not applied until convergence is achieved. The search phase is preceded by two steps based on vertex invariants and local AllDifferent constraints [3,4]. Search strategy is established by a static instantiation sequence based on the number of future branches. Our newly proposed algorithm, called CoreGraph, is based on a new search strategy which builds a static instantiation sequence of the query node. CoreGraph does not deal with complex filtering rule or domains. The basic idea for the construction of the search sequence is to maximize the number of branches to preceding nodes in the sequence. The sequence is recursively generated by adding those neighbors maximizing a score function. The score of each candidate node is assigned taking into account its degree, the number of its edges leading to nodes in the sequence and to their neighbors. Notice that, CoreGraph applies those filtering rules only to the query graph. Concerning the target graphs, the only information CoreGraph uses for pruning is node degree. Finally, since the search strategy does not give priority to more dense parts of the target graphs it results efficient in a large variety of query and target graphs.

Results

SubGraphLib contains the original implementation of VF2, LAD, and CoreGraph and a new implementation of FocusSearch in C++ (which is originally distributed in modula2). All algorithms have been compared on benchmarks such as synthetic unlabeled graphs, molecules, and biological networks. CoreGraph and FocusSearch in all cases outperform the other algorithms in terms of execution time. In most benchmarks, CoreGraph outperforms also FocusSearch. FocusSearch results particularly efficient on regular graphs having a mesh structure. However, since FocusSearch uses initial domains to avoid label comparisons, the memory requirements do not scale with respect to graphs size. On the

other hand, CoreGraph maintains a low memory profile.

References

1. Ullmann, J. R. 1976. An algorithm for subgraph isomorphism. J. ACM 23, 1, 31-42.
2. Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. 2004. A (sub)graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 10, 1367-1372.
3. Solnon, C. 2010. AllDifferent-based Filtering for subgraph isomorphism. Artif. Intell. 174, 12-13, 850-864.
4. Ullmann, J. R.. 2011. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism. J. Exp. Algorithmics15, Article 1.6 February 2011.