A Database Approach to Solve the Tree Containment Problem in Phylogenetic Networks

Sarah J. Berkemer, Pierre Bourhis, Philippe Gambette, Lionel Seinturier, Marion Tommasi

A Database Approach to Solve the Tree Containment Problem in Phylogenetic Networks

Sarah J. Berkemer[1] Pierre Bourhis[2] Philippe Gambette[3]
Lionel Seinturier[4] Marion Tommasi[5]

[1]École Polytechnique
[2]CNRS
[3]LIGM, Université Gustave Eiffel, CNRS
[4]Université de Lille
[5]Inria

2023-11-30

Classification AMS 2020: 68N17, 68P15, 05C85, 68R10, 68W10,

Keywords: phylogenetic networks, tree containment, databases, SAT, Datalog

1 Introduction

The TREE CONTAINMENT problem consists in deciding whether a phylogenetic tree is displayed by a phylogenetic network. It is NP-hard in general [5], although polynomial time algorithms were found for restricted cases when constraints are given on the structure of the phylogenetic network [4]. Considering this problem as a special case of directed subgraph homeomorphism [2] allows us to design several new algorithmic approaches to address it efficiently in practice, using the pebble game algorithm introduced to solve this problem. After proposing a SAT formulation of the problem, we turn to database theory, by first showing how to design a Datalog program to solve the problem. We then show how to optimize this approach to solve the problem faster in practice.

2 TREE CONTAINMENT and SUBGRAPH HOMEOMORPHISM

The TREE CONTAINMENT problem consists in deciding, given a rooted binary phylogenetic tree T, with leaves bijectively labeled by elements from a set X, and a rooted binary phylogenetic network N, with leaves bijectively labeled by elements of the same set X, if there exists a sequence of vertex and arc deletions which, when applied to N, result in obtaining T.

The SUBGRAPH HOMEOMORPHISM problem consists in deciding, given two directed graphs P and G, and a one-to-one mapping m of the nodes of P into the nodes of G, if P is homeomorphic to a subgraph of G, that is deciding if there exists a mapping m' from the arcs of P to pairwise internal-node-disjoint paths in G such that each arc (t, h) is mapped to a directed path from $m(t)$ to $m(h)$, such that all those paths are internal-vertex-disjoint (i. e. their only common vertices are their extremities). The connection between TREE CONTAINMENT and SUBGRAPH HOMEOMORPHISM, which was suggested in the PhD thesis of Juan Carles Pons Mayol [9], can be formally expressed as follows.

Theorem 2.1. *Considering that the size of the phylogenetic tree in* TREE CONTAINMENT *is constant, and that the size of the pattern P in* SUBGRAPH HOMEOMORPHISM *is constant,* TREE CONTAINMENT *reduces to* SUBGRAPH HOMEOMORPHISM *in polynomial time.*

Proof. Given an instance of TREE CONTAINMENT, that is a binary phylogenetic tree T with a constant number k of leaves and a binary phylogenetic network N with k leaves (with the same labels as T's) and with n vertices, T is contained in N if and only if there exists a mapping m from the root of T to the root of N, from the leaves of T to the leaves of N with the same labels, and from the other vertices of T to vertices of N such that the SUBGRAPH HOMEOMORPHISM problem has a positive answer for T, N and this mapping m. Therefore, this instance of TREE CONTAINMENT has a positive answer if and only if one of the $O(m^k)$ mappings of vertices of T into vertices of N, combined with the relevant mappings of the roots and of the labeled leaves, has a positive answer for the SUBGRAPH HOMEOMORPHISM problem. If k is a constant, this can be tested in polynomial time using the pebble algorithm for directed acyclic graphs described in Section 4 of [2]. □

3 A SAT formulation of TREE CONTAINMENT

We recall that the SAT problem consists in deciding whether a boolean formula is satisfiable. It is NP-complete, even when the boolean formula is in conjunctive normal form with $k \geq 3$ literals by clause (3-SAT) [6]. However, it can be solved efficiently in practice, for conjunctions of thousands of variables and millions of clauses, by solvers like Sat4j [8]. It may therefore be useful to reduce the TREE CONTAINMENT problem to a SAT problem, or more precisely to 5-SAT, where the boolean formula is a conjunction of disjunctions having at most 5 literals.

Theorem 3.1. *The* TREE CONTAINMENT *problem can be reduced to* 5-SAT *in polynomial time.*

We provide an algorithm which, given a phylogenetic tree T and a phylogenetic network N with n vertices as input, builds an instance $C(N, T)$ of the 5-SAT problem, in time $O(n^3)$. The general idea is to build this formula using variables x_{a_T,a_N}, where (a_T, a_N) is a pair of arcs of the tree and the network respectively, such that x_{a_T,a_N} has value *true* if $m(a_N) = a_T$, *false* otherwise. The fact that a_N is mapped to a_T has consequences on the possible mappings of arcs just below a_N, which lead to building clauses on the variables depending on local configurations of arcs below a_N and a_T.

Finally, it can be proven that the disjunction of all clauses built in this way is true if and only if T is contained in N.

4 A naive Datalog formulation of TREE CONTAINMENT

The TREE CONTAINMENT problem can be expressed with Datalog, a database query language based on the logic programming paradigm [1]. A first naive implementation reduces our problem into a Linear Datalog program and consists in splitting the program into two parts: first, rules to generate all the possible mappings and second, rules to check if a particular mapping is a homeomorphism.

5 Optimizations for the Datalog reformulations

The naive Datalog approach does not work in particular because it generates too many configurations which are not relevant. We improve our approach in three manners: the main one is to improve the Datalog program in order to reduce the number of configurations generated through the game. Instead of running the pebble game algorithm of [2] on all possible mappings of vertices of the tree T into vertices of the network N, as explained in the proof of Theorem 2.1, it is possible to start with known mappings of vertices, that is either the mapping of the roots of T and N (top-down approach) or the mapping of the leaves of T and N (bottom-up approach) and continue building the mappings gradually using the pebble game algorithm, stopping as soon as a rule of the pebble game algorithm does not provide a correct mapping. This results in generating only partial mappings of vertices which are candidates to be extended to mappings of all vertices of T with a positive outcome of the pebble game algorithm, which is faster in practice than generating all possible mappings of vertices before running the pebble game algorithm on them. We also note that it is possible to group some pebble moves of the pebble game algorithm when a pebble has to continue moving and has no alternatives between two different arcs for its next move.

Secondly, we introduce different techniques to explore the configurations i.e. the strategies to deduce the facts in a Linear Datalog program evaluation: we notice that the classical approach in Datalog is not optimal for our problem but a depth-first search seems to provide the best results.

Finally, we propose a new approach to parallelize the evaluation of Linear Datalog programs based on a Map-Reduce paradigm. Our tests show that for our problem this optimisation has an impact, by reducing by a factor of 4 the computation time for some instances.

6 Implementations

We implemented the reduction to 5-SAT in a Python script called `sat-tc.py`, with an external call to the Sat4j solver [8]. A direct implementation of our Datalog approach

was done using Python 3.5.2. It is available at `https://gitlab.inria.fr/Spirals/logical-approach-for-tree-containment`.

References

[1] Stefano Ceri, Georg Gottlob, Letizia Tanca What you always wanted to know about Datalog (and never dared to ask). *IEEE Transactions on Knowledge and Data Engineering*, 1(1), 146-166, 1989.

[2] Steven Fortune, John Hopcroft, James Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2), 111–121, 1980.

[3] Daniel Huson, Celine Scornavacca. Dendroscope 3: an interactive tool for rooted phylogenetic trees and networks. *Systematic Biology*, 61(6), 1061-1067, 2012.

[4] Leo van Iersel, Charles Semple, Mike Steel. Locating a tree in a phylogenetic network. *Information Processing Letters*, 110(23), 1037-1043, 2010.

[5] Iyad A. Kanj, Luay Nakhleh, Cuong Than, Ge Xia. Seeing the trees and their branches in the network is hard. *Theoretical Computer Science*, 401(1-3), 153-164, 2008.

[6] Richard M. Karp. *Reducibility Among Combinatorial Problems*. In Raymond E. Miller, James W. Thatcher, *Complexity of Computer Computations*, Plenum, 85-103, 1972.

[7] Hans Keur. *Creating phylogenetic networks from clusters*. Msc Thesis, Delft University of Technology, 2023.

[8] Daniel Le Berre, Anne Parrain. The Sat4j library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation*, 7(2-3), 59-64, 2010.

[9] Juan Carles Pons Mayol. *Reconstruction Problems for LGT Networks*. Doctoral Thesis, Universitat de les Illes Balears, p. 36, 2016.