AUT Journal of Mathematics and Computing

AUT J. Math. Com., 3(2) (2022) 147-151
DOI: 10.22060/AJMC.2022.21716.1104

Original Article

Breaking intractability of spanning caterpillar tree problem: A logical approach

Masoud Khosravani*[a]

[a]*Algoreen Software Ltd. Auckland, New Zealand*

ABSTRACT: In this paper we pursue a logical approach to prove that the optimisation problem of finding a spanning caterpillar tree in a graph has polynomial algorithm for bounded tree width graphs. A caterpillar (tree) is a tree with the property that if one removes all its leaves only a path is left. To this end we use Courcelle's theorem and we show how one can present the spanning caterpillar tree problem by using monadic-second order logical expression. The value of this approach reflected better by the fact that finding a spanning caterpillar in a graph is an NP-complete problem [9].

Review History:
Received:02 August 2022
Accepted:25 August 2022
Available Online:01 September 2022

Keywords:
Caterpillar trees
Monadic second order logic
Optimization
Parametrized complexity

AMS Subject Classification (2010):
05C85

(Dedicated to Professor S. Mehdi Tashakkori Hashemi)

1. Introduction

Whenever one wishes to model relations or dependencies among objects, the first model that comes to mind is a graph. That is why graphs are ubiquitous in computer science and related fields. By making such a model many questions can be addressed using the language of graph theory. For example:

- What is the largest set of objects with no mutual relationships? In graph theory term we are looking for a maximum independent set.
- How can one partition the objects into disjoint sets such that each set contains non-relating objects? A proper vertex colouring gives the answer to this question.
- How large is a set of objects in which every pair of objects are related? Here we wish to find a maximum clique.

In some applications one may wish to find a specific substructure within a graph. For example one may want to find a spanning tree to check the connectivity. In the Hamiltonian path problem we are looking for a path within a graph that covers all nodes in the graph. There are other applications where weights are assigned to links and/or nodes, to represent cost or traversal time. In this group of problems we look for a substructure with the

*Corresponding author.
E-mail addresses: masoud@algoreen.com, khosravani.m@gmail.com

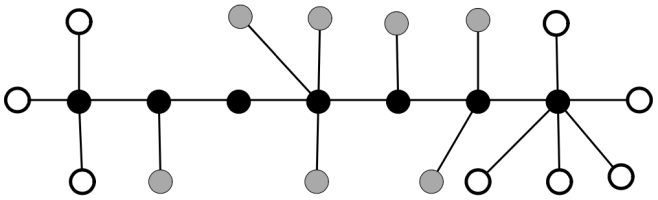

Figure 1: Figure Sample

optimal cost. The travelling salesman problem is a well studied example of such problems, where the goal is to find a spanning cycle with the minimum travel cost. Many of those graph problems are computationally intractable. That is, no one knows a polynomial-time algorithm for solving any of them. So researchers are using different techniques to find some solutions that are as good as possible. Here we focus on one of those intractable problem on finding spanning caterpillar within a graph.

Here we mention a few applications of finding spanning caterpillars within a graph. In many problems where a path plays a main role, a caterpillar tree can be considered as an alternative. So their applications are not restricted to the following ones.

1. In network design, we may wish to find a cost effective linearly arranged backbone to place our communication routers. Here the backbone is the spine of a caterpillar and leaves are the sites that are connected to the backbone. Generally the cost of backbone links is different with the cost of links that connect a site to a router.
2. In a facility transportation problem, where the task of distributing facilities is divided among one global but costly distributor and some local and cheap ones. Here, the global distributor follows the spine path to deliver facilities to warehouses and the local ones use the leaf edges to distribute them among end users. The goal is to find a transportation route that has the minimum overall cost.
3. In chemical graph theory caterpillars are considered as a model for benzenoid hydrocarbon molecules; see [6].
4. Caterpillars also appear in designing algorithms for solving RNA alignment and comparing evolutionary trees; see [10] and [4].

As another application of the MSCP, Tan and Zhang [12] used it to solve some problems concerning the Consecutive Ones problem. For more applications in combinatorics and mathematics (in general) we refer the reader to [11]. In what follows we first introduce the more rigorous definition of caterpillar tree and related topic, then we explain what is a Monadic Second Order logic expression is and finally we show the main result of the paper.

2. Definitions

2.1. Spaning Caterpillar Trees and Related Problems

Let $G = (V, E)$ be a graph. By a caterpillar (sub-)tree of G we mean a tree that reduces to a path by deleting all its leaves. We refer to the remaining path as the spine of the caterpillar. The edges of a caterpillar H can be partitioned to two sets, the spine edges, $B(H)$, and the leaf edges, $L(H)$, see Figure 1. One common problem may be to find spanning caterpillar in a graph. Another problem is to find Minimum Spanning Caterpillar Problem (MSCP). Where one wants to find a caterpillar with the minimum cost that contains all nodes. As mentioned MSCP is NP-complete for general graphs [9].

2.2. Bounded Tree Width Graphs

We define a tree decomposition of a graph $G = (V, E)$ by a tree T that its nodes $X_1, \ldots, X_n$ each is a subset of $V(G)$ so we refer to them as subset nodes of T to distinguish them from nodes of G, satisfying the following properties (the term node is used to refer to a vertex of T to avoid confusion with vertices of G):

1. The union of all sets X_i equals V, i.e. each node of G is contained in at least one subset node.
2. If X_i and X_j both contain a vertex v of G, then all nodes X_k of T in the (unique) path between X_i and X_j contain v as well.
3. For every edge (v, w) in the graph G, there is a subset node X_i that contains both v and w. That is, when vertices are adjacent in the graph then the corresponding subtrees have a node in common, too.

The width of a tree decomposition is the size of its largest set X_i minus one. The treewidth $tw(G)$ of a graph G is the minimum width among all possible tree decompositions of G. For more information on tree-width graph see [5].

2.3. Monadic Second Order Logic

It is known that graph structures that are expressible by Monadic Second Order Logic (MSOL) [3] [2]are recognizable in linear time on bounded tree-width graphs. The same is true for those optimization problems that can be defined by Extended Monadic Second Order Logic (EMSOL); see the survey of Hlineny et al. [8]. In the next section we introduce a formula that expresses spanning caterpillar problem in the language of monadic second order logic. Let us first define the first order formulas. For a more comprehensive introduction to logic refer to Enderton [7]. Formulas in the first order logic are made from

1. a set of variables,
2. a set of constant symbols,
3. functional symbols (with arities),
4. relational symbols (with arities),
5. two logical quantifiers $\forall$ and $\exists$, and
6. the set of connective symbols $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$.

The set of terms consists of expressions that are made from constants and variable symbols by applying (zero or more times) of functions. Note that this definition also consider all constants and variable symbols as terms. Now we define atomic formulas by using terms and relations. An atomic formula is an expression $R(t_1, \ldots, t_n)$, where R is an n-ary relation and $t_1, \ldots, t_n$ are terms. Finally we build the set of formulas by using the connective symbols $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ and the quantifier symbols. The language of the first order logic consists of formulas in which quantifiers are applied to variables. In the second order logic we define two more sets of variables, function variables and relation variables.

3. Monadic Second Order Logic Formulation

In this section we introduce the Monadic Second Order Logic (MSO) expression of the spanning caterpillar problem. In an MSO formula the quantifiers are applied to variables or the set of variables. Here we are interested in the MSO logic that is defined on graphs, where the set variables are defined over nodes and edges of a graph. By the following theorem Courcelle [3] shows that all problems that are definable in MSO Logic can be solved in polynomial time on the class of graphs with bounded tree-width.

Theorem 3.1. *Let P be a graph problem that has an MSO sentence ϕ. Also let $G = (V, E)$ be a graph of tree-width at most k where k is a fixed value. Then there is an Algorithm that solves P on input G in time $f(\|\phi\|, w)n \in O(n)$, where $\|\phi\|$ denotes the length of the MSO formula ϕ.*

Arnborg, Lagergren and Sees [1] extended the Courcelle's result by introducing Extended Monadic Second Order (EMSO) logic. They show that optimization problems which are definable as an EMSO formula have a polynomial solution on graphs of bounded tree-width. There are properties that cannot be formulated by the first order logic, but they can be expressed in MSO logic. For example, connectivity of a graph can be formulated in MSO logic while it is not possible to express it as a first order sentence. So in that sense, we can say that MSO logic is more powerful than the first order logic.

An MSO formula in the language of graphs is made from atomic formulas $E(x, y)$ and the expression $x = y$ by using the connective and the quantifier symbols. The atomic formula $E(x, y)$ denotes adjacency of x and y. The formula $x = y$ expresses equality relation between x and y. Also the formula $X(x)$ denotes that variable x (a node in a graph) belongs to X (a subset of the set of nodes). In what follows we use lowercase letters such as x, y, z to indicate individual variables and capital letters like X, Y, Z to denote set variables. Note that a caterpillar is a tree in which there is a path P such that each node v, either belongs to P or is attached to P. Also we define a tree as a connected graph that has no cycle, see [36]. Now we are ready to express the existence of a caterpillar in MSO logic. To reduce the length of formulas we use the following conventions for using quantifiers and nodes set variables and edge set variables (the same conventions are applied when replacing $\forall$ by $\exists$):

1. instead of $\forall x(X(x))$ we write $\forall x \in X$,
2. we denote $\forall x X(x) \vee \forall y X(y)$ by $\forall x, y \in X$,
3. we show $\forall Y(Y \subseteq X)$ by $\forall Y \subset X$.
4. we write $E(x, y) \in F$ to show $E(x, y) \wedge x, y \in F$

In the following formulas we assume that X is a nodes set variable and F is an edges set variable. Also we assume that the subgraph relation is valid for (X, F), say $\text{sub}(X, F) = \forall E(x, y) \in F(x \in X \wedge y \in X)$ is always true.

Lemma 3.2. *There is an MSO formula that expresses that (X, F) is a connected subgraph*

Proof. It is easy to show that disconnectivity can be expressed by the following formula

$$\exists X(\exists x, x \in X \wedge \exists y, y \notin X \wedge \forall x, y(E(x,y) \rightarrow (x \in X \leftrightarrow y \in Y)))$$

and then we just need to negate the expression as follows.

$$\alpha(X,F) = \neg\Big(\exists X(\exists x, x \in X \wedge \exists y, y \notin X \wedge \forall x, y(E(x,y) \rightarrow (x \in X \leftrightarrow y \in Y)))\Big)$$

Lemma 3.3. *That a subgraph (X,F) is acyclic has MSO expression.*

Proof.

$$\beta(X,F) = \neg\exists Y \subseteq X\Big(\exists x \in Y \wedge \forall x(x \in Y \rightarrow \exists y_1 \in X \exists y_2 \in X(y_1 \neq y_2 \wedge E(x,y_1) \in F \wedge E(x,y_2) \in F \wedge y_1 \in Y \wedge y_2 \in Y))\Big),$$

Lemma 3.4. *On can represent by MSO that a node x has degree one in subgraph (X,F)*

Proof.

$$\gamma(X,F,x) = \exists y \in X\Big(E(x,y) \in F \wedge (\forall z \in X(E(x,z) \in F \rightarrow y = z)\Big),$$

Lemma 3.5. *Node x has degree two in (X,F)*

Proof.

$$\theta(X,F,x) = \exists y,z \in X\Big((y \neq z) \wedge E(x,y) \in F \wedge E(x,z) \in F \wedge \big(\forall t \in X(E(x,t) \in F \rightarrow (t = y \vee t = z))\big)\Big),$$

Lemma 3.6. *Subgraph (X,F) represent a path in a graph.*

Proof.

$$\eta(X,F) = \alpha(X,F) \wedge \beta(X,F) \wedge \Big(\forall x \in X(\gamma(X,F,x) \vee \theta(X,F,x)\Big)),$$

Lemma 3.7. *Now we have enough materials to express that subgraph (X,F) is a caterpillar.*

Proof. Note that be Courcelle's theorem and since the following expression is a logical formula for the fact that a graph is a caterpillar if:

1. it is connected, as expressed by $\alpha(X,F)$, and
2. it is acyclic, as expressed by $\beta(X,F)$, and
3. it is a path, as expressed by $\forall x \in X(\gamma(X,F,x) \vee \theta(X,F,x)$, i.e. each vertex is of degree one or two, and
4. the vertices of the graph is divided to leaves and spine vertices, that are presented by Y and $X - Y$ in the following formule
$$\exists Y \subseteq X \wedge \exists H \subseteq F \wedge \eta(Y,H) \wedge (\forall x \in X(\exists y \in Y \wedge E(x,y) \in F))$$

Finally, one may combine those properties by the combination of logical and, as follows

$$\mathrm{cat}(X,F) = \alpha(X,F) \wedge \beta(X,F) \wedge \Big(\exists Y \subseteq X \wedge \exists H \subseteq F \wedge \eta(Y,H) \wedge (\forall x \in X(\exists y \in Y \wedge E(x,y) \in F))\Big).$$

Theorem 3.8. *For Spanning Caterpillar Problem there is an algorithm that solves it in polynomial time on graph with bounded tree-width.*

Proof. The proof follows from the following MSO expression. The first part of logical conjunction indicates that the subgraph is spanning subgraph and the second part refer to the fact that the subgraph in indeed a caterpillar tree.

$$\mathrm{spanCat} = \exists X \exists F \forall x\Big(x \in X \wedge \mathrm{cat}(X,F)\Big).$$

And now applying Courcelle's main theorem completes the proof.

4. Conclusion

Using MSO to show that a graph has polynomial solution for a parametrized input, opens a new way to break intractability by automatically generating formula. Since there has been long research history in computer aided proof in other branched of Mathematics and Computer Science, such an automated generated proof may be re-deployed to show more NP-hard problems have parametrized algorithms.

Acknowledgement

The author wishes to thank Professor Micheal J. Dinneen from University of Auckland for his suggestion to follow this approach and his helpful comments. Parts of this research were done while the author was affiliated with the Department of Computer Science, University of Auckland.

References

[1] S. Arnborg, J. Lagergren, and D. Seese, *Easy problems for tree-decomposable graphs*, J. Algorithms, 12 (1991), p. 308–340.

[2] B. Courcelle, *Graph rewriting: An algebraic and logic approach*, in Formal Models and Semantics, Elsevier, 1990, pp. 193–242.

[3] ——, *The monadic second-order logic of graphs. i. recognizable sets of finite graphs*, Information and computation, 85 (1990), pp. 12–75.

[4] M. Csűrös, J. A. Holey, and I. B. Rogozin, *In search of lost introns*, Bioinformatics, 23 (2007), pp. i87–i96.

[5] R. G. Downey and M. R. Fellows, *Parameterized complexity*, Springer Science & Business Media, 2012.

[6] S. El-Basil, *Caterpillar (gutman) trees in chemical graph theory*, Advances in the theory of Benzenoid hydrocarbons, (1990), pp. 273–289.

[7] H. B. Enderton, *A mathematical introduction to logic*, Elsevier, 2001.

[8] P. Hliněný, S.-i. Oum, D. Seese, and G. Gottlob, *Width parameters beyond tree-width and their applications*, The computer journal, 51 (2008), pp. 326–362.

[9] M. Khosravani, *Searching for optimal caterpillars in general and bounded treewidth graphs*, PhD thesis, ResearchSpace, Auckland, 2011.

[10] A. Lozano, R. Y. Pinter, O. Rokhlenko, G. Valiente, and M. Ziv-Ukelson, *Seeded tree alignment*, IEEE/ACM transactions on Computational Biology and Bioinformatics, 5 (2008), pp. 503–513.

[11] M. Numan, A. Nawaz, A. Aslam, and S. I. Butt, *Hosoya polynomial for subdivided caterpillar graphs*, Combinatorial Chemistry & High Throughput Screening, 25 (2022), pp. 554–559.

[12] J. Tan and L. Zhang, *The consecutive ones submatrix problem for sparse matrices*, Algorithmica, 48 (2007), pp. 287–299.

Please cite this article using:

Masoud Khosravani, Breaking intractability of spanning caterpillar tree problem: A logical approach, AUT J. Math. Com., 3(2) (2022) 147-151
DOI: 10.22060/AJMC.2022.21716.1104