A Solver for Monadic Second Order Logic of Graphs of Bounded Tree-width

Radu IOSIF (VERIMAG, Grenoble)
mailto:Radu.Iosif@univ-grenoble-alpes.fr

1 Background

Graphs are important for many areas of computing such as static analysis [7], databases with knowledge representation [1] and concurrency [4]. A well-established language for specifying graph properties is Monadic Second Order Logic (MSO), where quantification is over vertices only, or both vertices and edges, and sets thereof [3].

The tree-width of a graph is a positive integer measuring the distance between the graph and a tree. For instance, trees have tree-width one, series-parallel graphs (i.e., circuits with one input and one output that can be either cascaded or overlaid) have tree-width two, whereas $n \times n$ square grids have tree-width n, for any $n \geq 1$. The tree-width is a cornerstone of algorithmic tractability. For instance, many NP-complete graph problems such as Hamiltonicity and 3-Coloring become PTIME, when restricted to inputs whose tree-width is uniformly bounded by a constant, see, e.g., [5, Chapter 11].

Having a uniform bound on the tree-width of the graphs in class sets a sharp frontier between the decidability[1] and undecidability of Monadic Second Order (MSO) theories. A result of Courcelle [2] proves that MSO is decidable over bounded tree-width structures, by reduction to the emptiness problem of tree automata. A dual result of Seese [8] proves that each class of structures with a decidable MSO theory necessarily has bounded treewidth. Since MSO is the yardstick of graph specification logics, these results show that *tree-width bounded* classes of structures are tantamount to the existence of decision procedures for important classes of properties.

2 Challenges and Goal

Despite being an important tool for reasoning about graphs, there is currently no implementation for a solver that takes as input an MSO formula and an integer k and decides whether there exists a graph of tree-width at most k that is a model of the given formula.

A naïve implementation would encode the input problem as the satisfiability of an MSO formula interpreted over trees, following the proof of Courcelle's Theorem (see, e.g., [5, Chapter 11] for a comprehensive presentation of this proof), for which efficient implementations exist (see, e.g., [6]).

However, a limitation often experienced by automata-based decision procedures is the state-explosion problem. Indeed, the size of the automata constructed from the MSO formula becomes huge quickly exhausting the available memory before an answer can be given.

The goal of this internship is to first try the direct implementation that follows Courcelle's proof, identity its bottlenecks and, second, devises solutions by generating more compact MSO formulæ over trees, that postpone (or even avoid) state explosion, in some cases.

The internship consists of both theoretical and implementation work.

3 How to Apply

Send your CV and a letter of intent to: mailto:Radu.Iosif@univ-grenoble-alpes.fr

[1] The existence of an algorithm that answers whether a given input formula has a model.

References

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, 2000.

[2] B. Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[3] B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2012.

[4] P. Degano, R. D. Nicola, and J. Meseguer, editors. *Concurrency, Graphs and Models, Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday*, volume 5065 of *Lecture Notes in Computer Science*. Springer, 2008.

[5] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

[6] J. Henriksen, J. Jensen, M. Jørgensen, N. Klarlund, B. Paige, T. Rauhe, and A. Sandholm. Mona: Monadic second-order logic in practice. In *Tools and Algorithms for the Construction and Analysis of Systems, First International Workshop, TACAS '95, LNCS 1019*, 1995.

[7] N. D. Jones and S. S. Muchnick. A flexible approach to interprocedural data flow analysis and programs with recursive data structures. In *Proceedings of the 9th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '82, page 66–74, New York, NY, USA, 1982. Association for Computing Machinery.

[8] D. Seese. The structure of the models of decidable monadic theories of graphs. *Annals of Pure and Applied Logic*, 53(2):169–195, 1991.