Unité Mixte de Recherche UGA - CNRS - Grenoble INP
Bâtiment IMAG
Université Grenoble Alpes
700, avenue centrale
38401 Saint Martin d'Hères
France
tel : +33 4 57 42 22 42
fax : +33 4 57 42 22 22
http://www-verimag.imag.fr/

Grenoble, 19 février 2020

Report on the thesis entitled

*Automata in Decision Procedures and Performance Analysis*

submitted by Mr. Tomas FIEDOR

**Motivation and Background** Automata-based techniques provide an important basis for understanding and application of formal verification to real-life computer systems. From a theoretical point of view, the connection between logic and automata theory provides deep model-theoretic insights related to the decidability and computational complexity of fragments of second-order monadic logic (MSOL), such as the weak calculus of one (WS1S) or countably many (WS$\omega$S) successor functions. In practice, finite automata offer push-button theorem proving mechanisms based on checking the emptiness of an automaton built inductively on the structure of a formula, which is possible via a nondeterministic logarithmic-space algorithm (respectively, polynomial-time algorithm for tree automata). The bottleneck of this method is the size of the automaton, which depends on the quantifier depth of the input formula, in a non-elementary recursive fashion (as a matter of fact, the satisfiability problems of WS1S and other related logics are Tower-complete).

This negative result opens new directions for research. Since the lower bound applies to a very large class of problems, it is important to know what is the average complexity of checking satisfiability of formulæ stemming from practical verification conditions. A follow-up question is then how to optimize automata-based solvers in order to perform well on practical examples. Research in this area has many applications in various branches of hardware and software verification, because many verification problems, such as checking memory-safety of sequential programs or checking deadlock-freedom of parametric concurrent systems ultimately reduce to checking satisfiability of formulæ written in various fragments of MSOL.

**Technical Contributions** The thesis is organized into three chapters. Chapter 1 describes a new satisfiability checking procedure for WS1S based on a refinement of the technique of antichains that avoids the exponential blowup caused by an explicit construction of the automaton from a formula. Chapter 2 takes this line of work further and describes a symbolic lazy evaluation technique for the same problem. Finally, Chapter 3 moves on to a different topic and describes a resource bounds analysis for pointer-manipulating programs, based on a translation of heap update statements into a updates of a finite set of integer norms.

**Nested Antichains** The main problem with the automata-based approach to logic is the worst-case exponential blowup caused by the quantifier alternation, which requires determinisation (automata corresponding to existentially quantified formulaæ are nondeterministic) prior to complementation. An interesting line of research consists in representing the set of states of

a deterministic automaton obtained by subset construction using only its minimal or maximal elements, which form an antichain. In this thesis, this idea is pushed further to considering closed WS1S formulæ with arbitrary alternation depths and building automata whose sets of states are nested sets of states thereof. Provided that certain operations on automata, such as union, intersection and quotienting, can be performed directly on this representation, much space can be saved using a suitable recursive subsumption relation, that only explores the elements of an antichain, in the nested set inclusion partial order.

This approach has been implemented in a tool called DWINA, which is compared with several existing tools (primarily with MONA, the reference tool in this area), showing encouraging results, but also room for improvement. In particular, it seems that important space savings can be achieved in exchange of more time consuming primitives for generation of predecessor states (right quotienting) and subsumption checking.

**Lazy Evaluation** The second approach to checking satisfiability of WS1S formulæ presented in this thesis combines automata-based with proof-theoretic techniques, such as tableaux and cyclic proofs. The main idea is to postpone the construction of automata for as much as possible, by expanding the input formula into a symbolic language term which is checked for emptiness using lazy tableau-like techniques. Interestingly, the evaluation of fixpoint iterations needed for the backward computation of the predecessors of final states can be done by unfolding a recursive fixpoint expression and a stronger (under-approximated) subsumption check can be done via term isomorphism, much like wih a cyclic induction proof.

Explicit automata generation can thus be postponed to the leaves of the language term, which avoids much of the overhead. However, the method does not benefit from automata minimization techniques (as it is the case of MONA) and must rely on further optimizations, such as miniscoping (called anti-prenexing in the thesis). The method has been implemented in a tool called GASTON, which shows steady improvement with respect to the existing state of the art tools, such as MONA.

**Resource Bounds Analysis** The final chapter of the thesis deals with a completely different topic, namely the analysis of the memory usage and worst-case execution time of a program with low-level memory handling primitives, such as allocation and pointer updates. The idea is to track the maximal length among all paths between the heap locations referred by two designated pointer variables, or alternatively, between a pointer variable and a condition on the data at the destination cell. This framework greatly extends previous quantitative analyses of pointer programs, essentially geared towards proving their termination, with analyses of the space and time complexity of programs with subtle interplay between the data structures in the heap and the values of the integer variables.

The method starts with a set of norms (integer variables) and statically translates the input source code into an integer program that only changes the values of the norms. An interesting feature of the translation is that new norms are added on-the-fly when tracking additional paths may refine the analysis. The technique has been implemented in a prototype tool called RANGER, which uses existing tools, such as FORESTER, for shape analysis and LOOPUS for computing the upper bounds of the given norms. Experimental evaluation and comparison with existing tools, such as APROVE and COSTA, show promising results.

**Thesis Evaluation** The thesis reports on nontrivial and novel results and provides interesting contributions to the area of verification of computer systems. The presentation is structured and

the manuscript is readable by researchers familiar with the topics addressed, with comprehensive explanations and several worked-out examples. However, the manuscript could have been improved in several ways, listed below :

- An overview of antichain-based checking of language inclusion (or universality) would have been beneficial for understanding the refinements of the method described in Chapter 1. Based on a generic antichain worklist algorithm, one could present the specialized version using nested sets, which is only informally described in Section 3.2.1. Moreover, a presentation of the algorithm in the form of pseudocode could then lead to a general proof of correctness of the method (though the results seem correct, based on experience with similar techniques). In particular, Lemmas 3.5 and 3.6 are identical copies of Lemmas 3.3 and 3.4. I would suggest simply removing Lemmas 3.3 and 3.4 in a revised version.
- An overall lazy evaluation algorithm could have been formalized and proved in Chapter 2. In particular, the soundness of subsumption using term isomorphism requires proper formalization : a tableau-like proof of (non-)emptiness is sound only if every cycle (following direct successors and subsumption edges) contains a fixpoint unfolding (called a progress point, in the language of cyclic proofs), which guarantees that the cycle cannot be iterated ad infinitum. Although I do not doubt the correctness of the result, I believe that proving correctness properly would be a neat improvement of the thesis.
- Since the lazy evaluation method is promising and shows good improvement over purely automata-based methods, such as MONA, how difficult is to extend this method to trees (WS$\omega$S) and even graphs of bounded treewidth ? In particular, since there are no implemented solvers for MSOL over graphs of bounded treewidth, having such a tool could be beneficial, given the large number of applications (graph rewriting, Separation Logic, etc.)
- The resource bounds analysis in Chapter 3 is nicely presented, at the aid of examples and detailed explanations of the semantics of norm updates, however there is no general proof of soundness that would (at least on paper) ensure the correctness of the result. Although such small-step operational semantic proofs are quite time consuming, they are also a good way to catch subtle errors in the design of the verifier.

The results presented in this thesis have been published in 3 conferences (TACAS'15, TACAS'17, VMCAI'18) and 1 journal (Acta Informatica). The contents of the thesis has thus been published in a sufficient way. Summing up, I find that the PhD thesis of Tomas Fiedor meets the requirements of the proceedings leading to PhD title conferment. Taking into account the amount of work invested in the development of nontrivial and clearly novel techniques that address classical problems in verification, as well as the implementation of prototype tools that can be reused by other researchers, *I recommend the thesis to be accepted for defence and upon its successful completion, Tomas Fiedor to be assigned a PhD degree.*

Radu IOSIF

Researcher of the French National Research Center (CNRS),
VERIMAG laboratory