Abstract of Ph.D. Thesis

Automata for Branching and Layered Structures

Gabriele Puppis

Department of Mathematics and Computer Science
University of Udine, Italy
puppis@dimi.uniud.it

The aim of the thesis is to exploit different classes of (sequential and tree) automata for modeling and reasoning on infinite complex systems. The leit-motif underlying the results provided in the thesis is that, once one identifies a finite set of properties to be tested, any infinite complex system can be often reduced to a simpler one, which satisfies the same properties and presents strong regularities (e.g., periodicity) in its structure. As a consequence, checking properties of such a system can be done by considering only a finite number of components.

As for sequential automata, the most simple example is the notion of single-string automata, which recognizes a single infinite word. Such a notion of automaton makes it possible to represent and reason on any infinite ultimately periodic word by means of a finite number of states and transitions. The notion can then be refined by introducing counters in order to compactly represent repeated occurrences of the same substring in an infinite word. Thus, algorithms working on such a kind of representation exploit suitable abstractions of the configuration space of the automaton (these abstractions must take into account the indistinguishability of the valuations of the counters with respect to the transition function).

A similar approach can be adopted when dealing with the acceptance problem for tree automata. In such a case, abstractions are defined according to a suitable indistinguishability relation on colored trees with respect to the involved tree automaton. By exploiting such a notion of indistinguishability, one can reduce several instances of the model checking problem for monadic second-order logics (interpreted over non-regular trees) to equivalent instances of the acceptance problem for tree automata over regular trees, which are known to be decidable.

The thesis is organized in three chapters: Chapter 1 provides some background knowledge and basic notations; Chapter 2 focuses on different classes of sequential automata in the context of time granularity representation and management; Chapter 3 deals with tree automata and their acceptance problem. The contents of Chapter 2 and Chapter 3 are presented in the sequel, together with a short discussion of the most significant achievements.

Chapter 2 - Sequential automata and time granularities

Sequential automata turn out to be useful when one wants to represent temporal information and to deal with periodic phenomena. These tasks are widely recognized as relevant ones in a variety of application areas ranging from temporal database design and inter-operability to data conversion and data mining, to the specification and verification of reactive systems, to the synthesis, execution, and monitoring of timed workflow systems, to temporal constraint representation and reasoning, and to temporal abstraction. One of the most effective attempts at dealing with these problems takes advantage of the notion of time granularity.

Different time granularities can be used to specify the occurrence times of different classes of events. For instance, temporal characterizations of a flight departure, a business appointment, and a birthdate are usually given in terms of minutes, hours, and days, respectively. Moreover, when a computation involves pieces of information expressed at different time granularities, the system needs the ability of properly relating time granularities (this is usually the case, for instance, for query processing in federated database systems). Such an integration presupposes the formalization of the notion of granularity and the analysis of the relationships between different time granularities. According to a commonly accepted perspective, any time granularity can be viewed as the partitioning of a temporal domain in groups of elements, where each group is perceived as an indivisible unit (a granule). Most granularities of practical interest are modeled as infinite sequences of time granules, which present a repeating pattern and, possibly, temporal gaps within and between granules. A representation formalism can then use these granules to provide facts, actions or events with a temporal qualification, at the appropriate abstraction level. Even though conceptually clean, this point of view does not address the problem of providing infinite granularities with a finite (and compact) representation to make it possible to deal with them in an effective (and efficient) way. To be computationally acceptable, any formal system for time granularity should indeed satisfy the following requirements:

- **Suitable to algorithmic manipulation.** The formalism must provide infinite granularities with a finite representation. Furthermore, data structures, which are used to actually store information, should ease access to and manipulation of time granularities.
- **Powerful**. The set of all possible time granularities is not countable. Consequently, every representation formalism is bound to be incomplete. The class of granularities captured by the formalism should be expressive enough to be of practical interest.
- **Compact.** The formalism should exploit regularities exhibited by the considered granularities to make their representation as compact as possible.

Chapter 2 of the thesis presents suitable formalisms for the representation and the management of time granularities by exploiting the notion of sequential automaton, possibly extended with counters. In developing these formalisms, the attention is focused on the crucial problems of equivalence (to decide whether two given representations define the same time granularities), granule conversion (to relate granules of a given granularity to those of another one), granularity comparison (to decide whether there exist pairs of elements from two given sets of time granularities that satisfy a designated relation), and optimization (to compute compact and/or tractable representations of time granularities). Some real-world applications of automaton-based representations of time-granularities are also given. Chapter 2 is organized as follows.

Section 2.1 reviews the most important contributions in the field, among all algebraic frameworks (e.g., collection expressions [16], slice expressions [20], Calendar Algebra [21]), logical approaches (e.g., Calendar Logic [22], propositional linear temporal logic for time granularities [8], integer periodicity constraints for time granularities [12]), and string-based and automaton-based formalisms (e.g., granspecs [28], single-string automata [11]).

In Section 2.2 time granularities are defined as labeled partitions of a subset of the underlying temporal domain, which contain countably many non-overlapping equivalence classes (i.e., granules). In this sense, the definition of time granularity is similar to the one proposed by Bettini et al. [3]. Other similar notions of time granularity are discussed and compared with the chosen one. In this section, we also introduce some standard relationships between time granularities (e.g., partition, group, sub-granularity, aligned refinement, etc.).

Section 2.3 describes in details the algebraic formalism of Calendar Algebra, which allows one to represent a broad class of granularities in a natural and compact way. More specifically, Calendar Algebra uses symbolic operations, called calendar operations, which capture significant relationships between time granularities. Calendar operations, once they are applied to some known granularities, form symbolic expressions that represent new granularities in terms of existing ones. As an example, the granularity `Week` can be generated by applying the operation $Group_7$ to the granularity `Day`. Since these operations reflect the ways people define new granularities from existing ones, Calendar Algebra turns out to be a natural and intuitive way to represent user-defined granularities. Moreover, granularity representations are typically built, directly or indirectly, from a fixed single bottom granularity (e.g., `Second`, `Hour`, or `Day`, depending on the accuracy required in the application context).

Section 2.4 introduces Wijsen's string-based approach to time granularity. In this section we also describe an efficient (linear-time) algorithm that solves the equivalence problem for string-based representations of time granularities. We then compare the expressiveness of the string-based formalism with that of Calendar Algebra, by providing suitable algorithms that map Calendar Algebra expressions to equivalent string-based representations. By taking advantage of such a correspondence and by exploiting the proposed solution to the equivalence problem for string-based representations, one can easily solve the problem of deciding whether two given expressions of Calendar Algebra are equivalent or not (as a matter of fact, a strong limitation of the formalism of Calendar Algebra is that it focuses on expressiveness issues mainly, and almost completely ignores some basic problems of obvious theoretical and practical importance). Moreover, from this perspective, the string-based approach (as well as the automaton-based one, which is intimately related to it) can be viewed as a 'low-level' framework into which 'high-level' and 'user-friendly' formalisms like Calendar Algebra can be mapped.

Sections 2.5, 2.6, and 2.7 explore the automaton-based approach to time granularity in full detail. Section 2.5 reviews the notion of (sequential) single-string automaton, which was originally proposed by Dal Lago and Montanari and was used to recognize single ultimately periodic words. We also consider extensions of automata with counters ranging over discrete domains, in order to compactly encode redundancies of temporal structures. Precisely, we exploit the possibility of activating different transitions from the same (control) state and we rule them through guards envisaging the values of the counters. Such an idea is somehow related to the notion of timed automaton [1], but, differently from that notion, no synchronization between counters is performed (instead, suitable operators are used to explicitly update the values of the counters when a transition is activated). We then show that single-string automata (possibly extended with counters and multiple transitions) are as expressive as the formalism of Calendar Algebra and Wijsen's string-based models.

Section 2.6 introduces a refined notion of single-string automaton with counters, called restricted labeled single-string automaton (RLA for short). Such a refined notion of automaton is obtained by enforcing suitable restrictions on the structure of the transition functions, which are then exploited to devise efficient algorithms for basic problems involving RLA-based representations of time granularities. More precisely, we describe some deterministic polynomial-time algorithms that solve granule conversion problems and a non-deterministic polynomial-time algorithm that solves the (non-)equivalence problem. The latter solution is based on a reduction of the non-equivalence problem to a number-theoretic problem, namely, the problem of testing the satisfiability of linear diophantine equations, where variables are constrained by lower and upper bounds. We also consider optimization problems for RLA-based representations of time granularities, by identifying two possible ways of optimizing such representations. According to the first one, optimizing means computing the smallest (i.e., size-optimal) representation of a given time granularity; according to the second one, optimizing means computing the most tractable (i.e., complexity-optimal) representation of a given granularity, that is, the one on which crucial algorithms (e.g., conversion algorithms) run fastest. These two criteria have been showed to be not equivalent and to yield non-unique solutions. At the end of Section 2.6, we provide suitable polynomial-time algorithms that receive in input a string-based representation of a time granularity and compute an equivalent complexity-/size-optimal automaton-based representation of it.

Section 2.7 focuses on the problem of representing and reasoning on (possibly infinite) sets of periodical time granularities, rather than single time granularities. We first give a characterization of the class of Büchi-recognizable (i.e., regular) ω-languages consisting of ultimately periodic words only. Then, by exploiting such a characterization, we define a proper subclass of Büchi automata, called ultimately periodic automata (UPA), which recognize exactly the regular ω-languages of ultimately periodic words. As a matter of fact, UPA allow one to encode single granularities, sets of granularities which have the same repeating pattern and different prefixes, and sets of granularities characterized by a finite set of non-equivalent repeating patterns (a formal notion of equivalence for repeating patterns is given in the thesis), as well as any possible combination of them. In this section we also describe efficient (deterministic and non-deterministic) solutions to several basic problems involving sets of time granularities, precisely: the emptiness problem (that is to decide whether a given set of time granularities is empty), the membership problem (that is to decide whether a given time granularity belongs to a given set of time granularities), the equivalence problem (that is to decide whether two given representations define the same set of time granularities), the size-optimization problem (that is to compute compact representations of a given set of time granularities), and the comparison of granularities (that is, given two sets of time granularities $\mathcal{G}$ and $\mathcal{H}$, to decide whether there exist $G \in \mathcal{G}$ and $H \in \mathcal{H}$ such that $G \sim H$, where $\sim$ is one of the commonly-used relations between time granularities, namely, partition, group, sub-granularity, aligned refinement, etc.).

Chapter 3 - Tree automata and MSO logics

Tree automata usually come into play in the automatic verification of properties of infinite-state systems. A natural approach to this problem is to model a system as a directed graph, whose vertices (resp., edges) represent system configurations (resp., transitions). An expected property of the system is then expressed by a logical formula, which can be satisfied or not by the corresponding graph, thought of as a relational (Kripke) structure. Thus, the verification problem reduces to the model checking problem, namely, the problem of deciding the truth of a given formula interpreted over a fixed relational structure (possibly expanded with valuations for free variables).

Monadic second-order (MSO) logic has been commonly used as a specification language, because it is powerful enough to express relevant properties of graph structures such as reachability and confluency properties. It can be viewed as an extension of first-order logic with set variables, namely, variables that have to be instantiated by sets of elements. Even though the model checking problem for MSO logic is highly undecidable, several approaches to such a problem have been proposed in the literature. As a matter of fact, the model checking problem for MSO logic can be often reduced to the acceptance problem for suitable classes of automata (e.g., Büchi automata over linear structures, Rabin automata over tree-shaped structures) [26]. As an example, Büchi's Theorem [2] can be exploited to decide the MSO theory of $(\mathbb{N}, <)$ (i.e., the natural numbers equipped with the usual ordering) and the MSO theories of some expansions of $(\mathbb{N}, <)$ by unary predicates. More precisely, one can reduce the model checking problem for MSO logic interpreted over a given relational structure $(\mathbb{N}, <, P)$ to the acceptance problem for Büchi automata over $(\mathbb{N}, <, P)$. The latter problem consists in determining, for any given Büchi automaton M, whether M accepts the infinite word that characterizes $(\mathbb{N}, <, P)$. Elgot and Rabin gave a positive answer to this problem for various relevant predicates $P \subseteq \mathbb{N}$, e.g., the factorial one [13]. Intuitively, their approach consists in defining a transformation of a given infinite word u into another infinite word u' and a transformation of a Büchi automaton M into another automaton M' in such a way that M accepts u iff M' accepts u'. If u' happens to be ultimately periodic, then the latter condition (and hence the original question) can be checked effectively. In [5] Elgot and Rabin's method has been generalized to deal with the class of profinitely ultimately periodic words, whose acceptance problem can be traced back to the case of ultimately periodic words.

Chapter 3 of the thesis basically shows that an approach similar to Elgot and Rabin's one can be followed in order to deal with expanded tree structures. Here the role of ultimately periodic words is taken by regular vertex-colored trees and the notion of factorization for infinite words is accordingly generalized to branching structures. More precisely, it is well known that the model-checking problem for MSO logic interpreted over a deterministic vertex-colored tree T is reducible to the acceptance problem of T for Rabin tree automata, namely, to the problem of checking, for any Rabin tree automaton M, whether M accepts T. The latter problem is easily shown to be decidable if T is a regular tree. By exploiting a suitable notion of tree equivalence (indistinguishability) with respect to tree automata, we show that the acceptance problem is decidable over a large class of non-regular trees as well. We also prove that such a class is closed with respect to several natural operations on trees and it includes meaningful relational structures inside and outside the Caucal hierarchy [6]. Finally, we consider the model checking problem for the chain fragment of MSO logic interpreted over the so-called multi-layered temporal structures, which are tree-shaped structures suited for modeling and reasoning about temporal relations at different 'grain levels'. Chapter 3 is organized as follows.

Section 3.1 preliminarily recalls some background knowledge in the field of automatic verification of infinite-state systems, in particular, Büchi's and Rabin's theorems [2, 23], Elgot and Rabin's contraction method [13], the approach introduced by Carton and Thomas exploiting the notion of profinitely ultimately periodic word [5], Muchnik's theorem [24, 10, 27], and the so-called Caucal hierarchy [6].

Section 3.2 reviews some commonly-used operations on (vertex-colored) trees, such as finite-state recolorings, second-order tree substitutions, tree morphisms, and tree transductions (the latter ones are transformations computed by deterministic top-down tree transducers). It also discusses some variants of finite-state recolorings and tree transductions, where the facilities of bounded and rational lookahead are introduced (see, for instance, [14, 7]). Moreover, in this section we compare the above mentioned operations and we provide characterizations for some of them. As an example, we show that any tree transduction is equivalent to a suitable composition of a regular tree morphism, a finite-state recoloring, and another regular tree morphism. A similar result holds if tree transductions and finite-state recolorings are equipped with the facility of bounded or rational lookahead.

Section 3.3 describes in full detail the automaton-based approach to establish the decidability of MSO theories of expanded tree structures. First, by taking advantage of well-known results from automata theory, we reduce the model checking problem for MSO logic to the acceptance problem for Rabin (or, equivalently, Muller) tree automata. Such a problem is proved to be decidable in the case of regular colored trees. Then, we extend the decidability result to several non-regular trees by exploiting a suitable notion of indistinguishability of trees with respect to tree automata. More precisely, we define an equivalence (indistinguishability) relation between trees, which is parametrized with respect to a given Muller tree automaton M. Its equivalence classes are called M-types and can be finitely represented as collections of features of some (partial) runs of M. Then, given a tree T, we decompose it into regular pieces (called factors) and we substitute each of them with the corresponding M-type (i.e., an equivalence class), thus obtaining a tree-shaped structure R, called retraction of T for M. We finally show that the problem of deciding whether the automaton M accepts the tree T can be reduced to an acceptance problem for (a bisimilar copy of) the retraction R, which is expectedly 'simpler' than the original acceptance problem (for instance, R could be a regular tree). By exploiting such a method, one can reduce, in a uniform way, instances of the model checking problem (over non-regular trees) to equivalent instances of the acceptance problem for tree automata over regular trees. Moreover, the proposed approach is somehow related to Shelah's composition method [25], which directly exploits indistiguishability of relational structures with respect to MSO formulas. However, differently from our approach, Shelah's composition method finds it difficult to manage different valuations of a given variable over distinct copies of the same factor. This problem arises, in particular, when one needs to reduce the model checking problem over a branching structure to the model checking problem over a linear one (for instance, see Example 3.3.21 in the thesis).

Section 3.4 shows that the the automaton-based approach to the decidability of MSO logic works effectively for a large class of complex (possibly non-regular) trees, which we call reducible trees. In particular, we prove a 'compatibility' property for the notion of M-type with respect to second-order tree substitutions; then, we prove closure properties for the class of regular trees with respect to several natural transformations of trees (for instance, finite-state recolorings with bounded lookahead, regular tree morphisms, etc.); finally, we generalize closure properties to the class of reducible trees. These results, besides showing the robustness of the class of reducible trees, provide a neat framework to reason on retractions of trees and to easily transfer decidability results.

Section 3.5 compares the effectiveness of our approach with that of other frameworks proposed in the literature. Precisely, we show that the class of reducible trees includes several interesting structures, such as the unfoldings of the deterministic context-free graphs [19], the algebraic trees [9], the tree generators for the levels of the Caucal hierarchy [4], and all the deterministic trees in the Caucal hierarchy obtained by alternating inverse *finite* mappings and unfoldings [6]. Moreover, we consider the model checking problem for MSO logics interpreted over the so-called multi-layered temporal structures [17, 18]. As an example, the k-refinable downward unbounded ω-layered structure can be viewed as an infinite ordered sequence of infinite k-ary trees, while the k-refinable upward unbounded ω-layered structure can be seen as a complete k-ary infinite tree generated from the leaves or, equivalently, as an infinite ordered sequence of finite increasing k-ary trees. The MSO theories of multi-layered structures have been shown to be expressive enough to capture meaningful temporal properties of reactive systems (such as 'P holds at all time points k^i, with $i \geq 0$' or 'P holds densely over a given time interval') and moreover decidable. In order to generalize previous results in the literature, we introduce a new notion of multi-layered structure, called k-refinable totally unbounded ω-layered structure, and we show that the MSO theory of such a kind of structure (extended with a suitable coloring predicate) subsumes the MSO theories of both downward unbounded and upward unbounded ω-layered structures. Then, by exploiting the automaton-based approach described in Section 3.3, we establish the decidability of the MSO theories of totally unbounded ω-layered structures. Finally, we provide new decidability results for the chain fragment of MSO logic interpreted over the totally unbounded ω-layered structures expanded with either the equi-level or the equi-column predicates (the equi-level predicate allows one to check whether two given elements belong to the same layer of the structure, while the equi-column predicate allows one to check whether two given elements are at the same distance from the origin of the layer they belong to). These results subsume previous achievements in the literature about multi-layered structures and they solve an open problem posed in [15], namely, establishing the decidability of the theory of the chain logic over the downward unbounded ω-layered structures expanded with the equi-column predicate.

References

[1] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[2] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proceedings of the International Congress for Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

[3] C. Bettini, S. Jajodia, and X.S. Wang. *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer, July 2000.

[4] A. Carayol and S. Wöhrle. The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata. In *Proceedings of the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 2914 of *LNCS*, pages 112–123. Springer, 2003.

[5] O. Carton and W. Thomas. The monadic theory of morphic infinite words and generalizations. *Information and Computation*, 176(1):51–65, 2002.

[6] D. Caucal. On infinite terms having a decidable monadic theory. In *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science*, volume 2420 of *LNCS*, pages 165–176. Springer, 2002.

[7] T. Colcombet and C. Löding. On the expressiveness of deterministic transducers over infinite trees. In *Proceedings of the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2996 of *LNCS*, pages 428–439. Springer, 2004.

[8] C. Combi, M. Franceschet, and A. Peron. Representing and reasoning about temporal granularities. *Journal of Logic and Computation*, 14:51–77, 2004.

[9] B. Courcelle. The monadic second-order logic of graphs II: Infinite graphs of bounded tree width. *Mathematical Systems Theory*, 21:187–221, 1989.

[10] B. Courcelle and I. Walukiewicz. Monadic second-order logic, graph coverings, and unfoldings of transition systems. *Annals of Pure and Applied Logic*, 92:35–62, 1998.

[11] U. Dal Lago and A. Montanari. Calendars, time granularities, and automata. In *Proceedings of the 7th International Symposium on Spatial and Temporal Databases (SSTD)*, volume 2121 of *LNCS*, pages 279–298. Springer, 2001.

[12] S. Demri. LTL over integer periodicity constraints. In I. Walukiewicz, editor, *Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (FOSSACS)*, volume 2987 of *LNCS*, pages 121–135. Springer, 2004.

[13] C.C. Elgot and M.O. Rabin. Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. *Journal of Symbolic Logic*, 31(2):169–181, 1966.

[14] J. Engelfriet. Top-down tree transducers with regular look-ahead. *Mathematical Systems Theory*, 10:289–303, 1977.

[15] M. Franceschet, A. Montanari, A. Peron, and G. Sciavicco. Definability and decidability of binary predicates for time granularity. In *Proceedings of the 10th International Symposium on Temporal Representation and Reasoning (TIME) and 4th International Conference on Temporal Logic (ICTL)*, pages 192–202. IEEE Computer Society Press, 2003.

[16] B. Leban, D. McDonald, and D. Foster. A representation for collections of temporal intervals. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, volume 1, pages 367–371. AAAI Press, 1986.

[17] A. Montanari. *Metric and Layered Temporal Logic for Time Granularity*. ILLC Dissertation Series 1996-02. Institute for Logic, Language and Computation, University of Amsterdam, 1996.

[18] A. Montanari, A. Peron, and A. Policriti. Theories of ω-layered metric temporal structures: Expressiveness and decidability. *Logic Journal of IGPL*, 7(1):79–102, 1999.

[19] D. Muller and P. Schupp. The theory of ends, pushdown automata, and second-order logics. *Theoretical Computer Science*, 37:51–75, 1985.

[20] M. Niezette and J. Stevenne. An efficient symbolic representation of periodic time. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 161–168, Baltimore, MD, 1992. ACM Press.

[21] P. Ning, S. Jajodia, and X.S. Wang. An algebraic representation of calendars. *Annals of Mathematics and Artificial Intelligence*, 36:5–38, 2002.

[22] H.J. Ohlbach and D.M. Gabbay. Calendar logic. *Journal of Applied Non-Classical Logics*, 8(4):291–324, 1999.

[23] M.O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[24] A.L. Semenov. Decidability of monadic theories. In *Proceedings of the Mathematical Foundations of Computer Science (MFCS)*, volume 176 of *LNCS*, pages 162–175. Springer, 1984.

[25] S. Shelah. The monadic theory of order. *Annals of Mathematics*, 102:379–419, 1975.

[26] W. Thomas. Languages, automata, and logic. In G. Rozemberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.

[27] I. Walukiewicz. Monadic second-order logic on tree-like structures. *Theoretical Computer Science*, 275:311–346, 2002.

[28] J. Wijsen. A string-based model for infinite granularities. In C. Bettini and A. Montanari, editors, *Proceedings of the AAAI Workshop on Spatial and Temporal Granularities*, pages 9–16. AAAI Press, 2000.