The Way to Go: Multi-Level Temporal Logics

Angelo Montanari, Adriano Peron, and Alberto Policriti

Abstract

In this paper we briefly survey the main contributions of our research on time granularity and outline some directions for current and future researches. The original motivation of our research was the design of a temporal logic embedding the notion of time granularity, suitable for the specification of complex real-time systems, whose components evolve according to different time units. However, there are significant similarities between the problems we encountered in pursuing our goal, and those addressed by current research on combining logics, theories, and structures. Furthermore, exploiting interesting connections between multi-level temporal logics and automata theory that we recently established, a complementary point of view on time granularity arises: time granularity can be viewed not only as an important feature of a representation language, but as well as a formal tool to investigate expressiveness and decidability properties of temporal theories. Finally, as a by-product of our work, we defined a uniform framework for time and states that "reconciles" the tense logic and the logic of program perspectives.

Contents

The Way to Go: Multi-Level Temporal Logics

It will be clear from all that I have said what my own view of temporal structure is. There is a hierarchy of various temporal structures, corresponding to different 'grain sizes' that we want to study. One can have intervals plus points, or days/hours/seconds. In this richer ontology, one wants to develop a variety of temporal structures, plus a new feature*: their systematic interconnections.*

— *Points on Time*, Johan van Benthem

1 Motivations

As pointed out in [vB95], the ability of providing and relating temporal representations at different 'grain levels' of the same reality is widely recognized as an important research theme for temporal logic and a major requirement for many applications, including formal specifications of real-time systems, temporal databases, and data mining. Despite such a widespread recognition of its relevance, there is a lack of a systematic framework for time granularity.

The original motivation of our research was the design of a temporal logic embedding the notion of time granularity, suitable for the specification of complex real-time systems, whose components evolve according to different time units. However, it is worth noting that there are significant similarities between the problems we encountered in pursuing our goal, and those addressed by current research on combining logics, theories, and structures. Furthermore, we recently established interesting connections between multi-level temporal logics and automata theory that suggests a complementary point of view on time granularity: besides an important feature of a representation language, time granularity can be viewed as a formal tool to investigate expressiveness and decidability properties of temporal theories.

1.1 The specification of granular real-time systems

Logic-based methods for representing and reasoning about temporal information have proved to be highly beneficial in the area of formal specifications. Timing properties play a major role in the specification of reactive and concurrent software systems that operate in real-time, which are among the most critical software systems. They constrain the interactions between different components of the system as well as between the system and its environment, and minor changes in the precise timing of interactions may lead to radically different behaviors.

Temporal logic has been successfully used for modeling and analyzing the behavior of reactive and concurrent systems, e.g. [MP95]. It supports semantic model checking, which can be used to verify the consistency of specifications, and to check positive and negative examples of system behavior against specifications; it also supports pure syntactic deduction, which may be used to prove properties of systems. Unfortunately, most common specification languages are inadequate for real-time applications: they cannot deal with temporal properties in a simple and satisfactory way, because they lack an explicit and quantitative representation of time. A few remarkable exceptions do exist. They are extensions of Petri Nets or metric variants of temporal logic, which support direct and quantitative specifications of temporal properties and relevant validation activities.

There are, however, systems whose temporal specification is far from being simple even with timed Petri Nets or metric temporal logic. Consider the wide-ranging class of real-time systems whose components have dynamic behaviours regulated by very different—even by orders of magnitude—time constants (hereafter *granular real-time systems*). As an example, a pondage power station consists of a reservoir, with filling and emptying times of days or weeks, generator units, possibly changing state in a few seconds, and electronic control devices, evolving in milliseconds or even less. A complete specification of the power station must include the description of these components and of their interactions. A natural description of the temporal evolution of the reservoir state will probably use days: "During rainy weeks, the level of the reservoir increases 1 meter a day", while the description of the control devices behaviour may use microseconds: "When an alarm comes from the level sensors, send an acknowledge signal in 50 microseconds". We say that systems of such a type have *different time granularities*. It is somewhat unnatural, and sometimes impossible, to compel the specifier of these systems to use a unique time granularity, microseconds in the previous example, to describe the behaviour of all the components. For instance, the specifier of the requirements for a pondage power plant should not be compelled to write sentences like "the filling of the reservoir must be completed within n microseconds". A good language must allow the specifier to easily describe all simple and intuitively clear facts. A major issue of specification languages is indeed the naturalness of the notation. Hence, a specification language for granular real-time systems must support *different time granularities*.

1.2 The combining logics perspective

Even though the original motivation of our work on time granularity was the design of a temporal logic suitable for the specification of granular real-time systems, there are significant similarities between the problems it addresses and those dealt with by the current research on logics that model changing contexts and perspectives. Indeed, even if it has been developed in a temporal framework, our proposal actually outlines the basic features of a general *logic of granularity*. In this respect, it can be seen as a generalization of the well-known Rescher and Garson's topological logic to layered structures. Moreover, it presents interesting connections with the logics of contexts recently developed in the area of knowledge representation, where modalities are used to shift variables, domains, and interpretation functions from one context to another. More generally, the design of these types of logics is emerging as a relevant research topic in the broader area of combination of logics, theories, and structures, at the intersection of logic with artificial intelligence, computer science, and computational linguistics.

In our work, we devised suitable combination techniques both to define temporal logics for time granularity and to prove their logical properties, such as decidability. Furthermore, we expect them to help us in finding a direct proof of the completeness of the proposed axiomatization (cf. Section 2.1). We did not directly address the problem of proving the completeness of the axiomatization. However, we proved that the theories of significant classes of metric and layered temporal structures for time granularity are decidable. Axiomatic completeness follows as a by-product of decidability, even though the axioms are not produced explicitly: since such theories are decidable, one can list all their theorems and thus axiomatic completeness trivially follows. As for the completeness of the proposed axiomatization, there are at least two possible approaches to such a problem. On the one hand, one can adopt the direct approach of building a canonical model for the proposed logic. Even though there seem to be no specific technical problems to solve, the process of canonical model construction is undoubtedly very demanding in view of the size and complexity of the axiom system. On the other hand, one can follow the approach outlined by Finger and Gabbay in [FG96], viewing the multi-level temporal logic for time granularity as the combination of a number of differently-grained metric temporal logics, and determining what constraints such a combination must satisfy to guarantee the transference of the completeness results from the component metric temporal logics to the combined one. This second approach seems the most promising one with respect to the problem of mastering the complexity of the axiomatization.

1.3 A complementary point of view on time granularity

Recent research suggests a complementary point of view on time granularity: besides an important feature of a representation language, time granularity can be viewed as a formal tool to investigate the expressibility of meaningful timing properties, such as density and exponential grow/decay, as well as the expressiveness and decidability of temporal theories [MPP99a]. In this respect, the number of layers (single vs. multiple, finite vs. infinite) of the underlying temporal structure, as well as the nature of their interconnections, play a major role: certain timing properties can be expressed using a single layer; others using a finite number of layers; others only exploiting an infinite number of layers. In particular, finitely-layered metric temporal logics can be used to specify timing properties of granular real-time systems composed by a finite number of differently-grained temporal components, which have been fixed once and for all (closed/rigid systems). Furthermore, if provided with a rich enough layered structure, they suffice to deal with conditions like "p holds at all even times of a given temporal domain" that cannot be expressed using flat propositional temporal logics [Em90]. On the contrary, ω-layered metric temporal logics are needed to specify granular real-time systems that can dynamically change their structure, e.g., by adding temporal components of possibly different grain sizes (open/flexible systems). As a matter of fact, ω-layered metric temporal logics also allow one to express relevant properties of infinite sequences of states over a single temporal domain that cannot be captured by using flat or finitely-layered temporal logics. This is the case, for instance, of conditions like "p holds at all time points 2^i, for all natural numbers i, of a given temporal domain". This

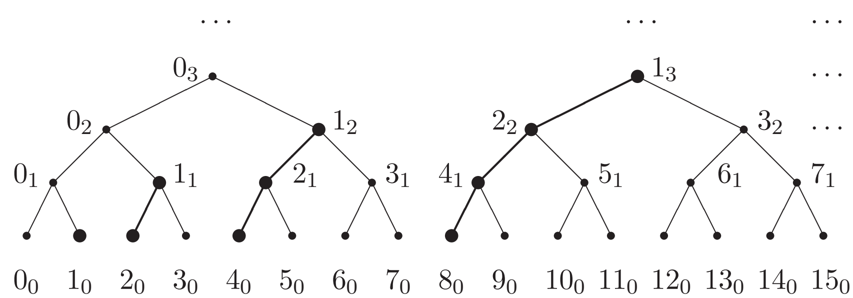

Figure 1: The set of powers of two (big circles and thick lines).

condition can be expressed in a monadic second-order logic interpreted over ω-layered metric temporal structures, such that each time point belonging to a given layer, with the exception of the finest layer, can be decomposed into 2 time points of the immediately finer layer (*upward unbouded 2-refinable metric and layered temporal structures*). An example of such structures is given in Figure 1, where time points 2^i are identified by means of big circles. It is easy to see that, for any given layer, time points 2^i are all and only those points that belong to the left downward closure of the right children of the time points belonging to the set $\{0_i : i \geq 0\}$.

1.4 Reconciling tense logics and logics of programs

Logic and computer science communities have traditionally followed a different approach to the problem of representing and reasoning about time and states (both approaches actually date back to Prior's different grades of tense-logical involvement). Research in philosophy, linguistics, and mathematical logic resulted in a family of (metric) tense logics that take *time* as a primitive notion and define (*timed*) *states* as sets of atomic propositions which are true at given instants, e.g. [Bur84]. In the last year, a few papers demonstrated the possibility of successfully exploiting metric (possibly layered) tense logics in computer science, e.g. [Mon96, MdR97]. On the other hand, most research in computer science concentrated on the so–called temporal logics of programs, which have been largely used to specify and verify reactive and concurrent systems, e.g. [Em90]). In order to deal with real-time systems, such logics have been provided with a metric of time, e.g. [AH93]. The resulting temporal logics, called *real-time logics*, take *state* as a primitive notion, and define *time* as an attribute of states. More precisely, given an ordered set of states $\mathcal{S}$ and an ordered set of time points $\mathcal{T}$, real-time logics are characterized by a weakly monotonic function $\rho : \mathcal{S} \to \mathcal{T}$ that associates a time instant with each state. As it is clear, it may happen that there exist pairs of states $s_i, s_j \in \mathcal{S}$ such that $s_i < s_j$ and $\rho(s_i) = \rho(s_j)$ (temporally indistinguishable states), or $\rho(s_{i+1}) > \rho(s_i) + 1$ (temporal gaps between states). Metric temporal logics endowed with an infinite number of layers (ω-layered), in which each time point belonging to a given layer can be decomposed into k time points of the immediately finer one (k-refinable), provide a unifying framework within which the two approaches can be reconciled [MPP99b]. The embedding of timed state sequences into upward

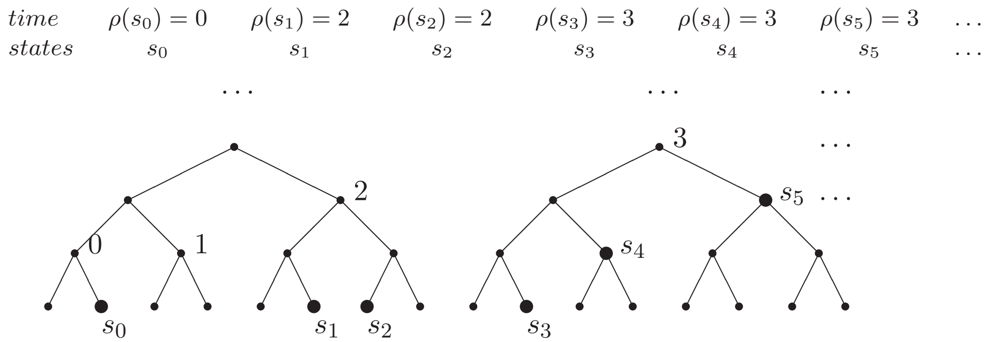

Figure 2: A timed state sequence and its embedding into an upward unbounded temporal structure.

unbounded layered structures is sketched in Figure 2, where both the states and their associated times are elements of the domain, and the mapping ρ of states into times is modeled by the standard *ancestor* relation over trees.

It is immediate to see that the proposed embedding allows us to deal with temporal indistinguishability and temporal gaps. The basic idea is that temporal indistinguishability and temporal gaps are due to the lack of the ability to express properties at the right level of granularity: distinct states, having the same associated time, can always be ordered at the right level of granularity; similarly, time gaps represent intervals in which a state cannot be specified at a finer level of granularity. With reference to the example of Figure 2, different states are associated with different descendants of the corresponding time elements of an upward unbounded temporal structure. States s_1 and s_2, as well as states s_3, s_4, and s_5, which are temporally indistinguishable in the given timed state sequence, share the same time ancestor; time 1, which is devoid of state descendants, models the temporal gap between states s_0 and s_1.

Notice that a finite number of layers is not sufficient to capture timed state sequences: it is not possible to fix a priori any bound on the granularity that a domain must have to allow one to temporally order a given pair of states, and thus we need to have an infinite number of temporal domains at our disposal.

2 The past

In the following, we briefly survey the main contributions of our research on time granularity. The main issues to be confronted when formalizing a temporal logic (for time granularity) are: (i) *expressiveness (definability) and axiomatization*, (ii) *decidability*, and (iii) *executability*.

2.1 Expressiveness and axiomatization

In [MdR97], we defined (a suitable extension of) *metric temporal logic* which provides a uniform framework in which both qualitative and quantitative timing properties of real-time systems can be expressed by means of a parametrized operator of (relative) temporal realization. We explore *completeness issues* of metric temporal logic (*MTL* for short). We do this by starting with a very basic system, and we build on it either by adding axioms or by enriching the underlying structures. We view *MTLs* as two-sorted logics having both formulae and parameters; formulae are evaluated at time instants, while parameters take values in an (ordered) abelian group of temporal displacements. We first define a minimal *MTL* that can be seen as the metric counterpart of minimal tense logic, and we provide it with a sound and complete axiomatization. Next, we characterize the class of two-sorted frames with a linearly ordered temporal domain. Then, we extend our systems with the ability to mix temporal and displacement formulae to make their logical machinery sufficiently powerful. Finally, we hint at the possibility of using the proposed two-sorted framework for characterizing a variety of *MTLs* simply by changing the requirements on the algebraic and/or temporal components.

In [CCMP93], we developed a *metric and layered temporal logic* (*MLTL* for short), extending *MTL* with time granularity, and showed how it can be used for specifying granular real-time systems. *MLTL* replaces the flat temporal domain of *MTLs* with a temporal universe consisting of a set of differently-grained temporal domains. Such a temporal universe identifies the relevant temporal domains and defines the relations between instants belonging to different domains. To qualify formulae with respect to the temporal universe, *MLTL* is provided with an operator of *contextualization* that identifies the domains a given formula refers to. Within each temporal domain, it is then possible to talk about truth and falsehood of formulae at different time instants by means of a *displacement* operator. Finally, a *projection* operator is added to constrain the relationships between formulae associated with differently-grained domains.

In [Mon96], we defined syntax, semantics, and (sound) axiomatization of *MLTL*. The language for *MLTL* is a three-sorted temporal language extending the (two-sorted) language for *MTL* with a *context* sort. The axiomatization of validity in the language of *MLTL* is obtained by adding to the axioms and rules of *MTL* a number of axiom schemata governing the behaviour of the contextual and projection operators as well as the relations between these operators and the displacement one.

2.2 Decidability

In order to guarantee the usefulness of *MLTLs* as formal tools, it is necessary to show some basic decidability properties. We obtained decision procedures for testing satisfiability (and validity) of *MLTL*-formulae by following an automata-theoretic approach. We reduced the satisfiability problem for theories of metric and layered temporal structures to the emptiness problem for automata on infinite objects.

The connection between automata theory and logic has been opened by Büchi, McNaughton, and Rabin (see [Tho90] for a general survey). Büchi showed that the collection of models (over natural numbers) satisfying a formula of the monadic second-order theory of one successor $MSO[<]$ is a ω-regular lan-

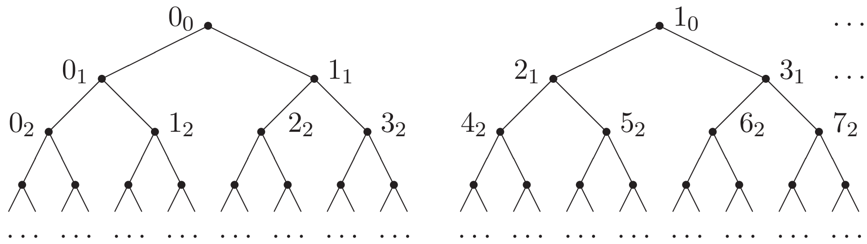

Figure 3: A downward unbounded (2-refinable) structure.

guage (i.e. it is accepted by a Büchi automaton) and vice versa. Rabin extended that result showing a similar correspondence between the second-order monadic theory of k-successors and the languages of regular k-ary infinite trees. The decidability of these theories turned out to be a powerful result to which a large number of other decision problems can be reduced (particularly, in the field of temporal logic). In fact, the theory of one (resp. k) successor can be considered as a "universal process logic" for linear (resp. branching) computations, as far as programs with a finite state space are considered.

In [MP96, MPP99a], we defined some *decidable theories of metric and layered temporal structures*. The decidability problem for the pure metric (non-granular) fragment has been originally addressed by Alur and Henzinger in [AH93]. They showed that, under suitable assumptions about the temporal domain and the associated operations, the satisfiability problems for real-time logics extending propositional linear temporal logics with metric features are decidable. These problems can indeed be reduced to the corresponding problems for $MSO[<]$. In [MP96], we presented a first extension of their results, aiming at dealing with time granularity. Such an extension allows one to treat situations in which a finite number of coarsenings/refinements of the temporal domain is sufficient. The key idea to deal with the resulting finitely-layered metric temporal structures is to reformulate the decidability problem into an equivalent one relative to the finest metric component (layer). We first formally defined the theory of finitely-layered metric temporal structures, and the associated second-order language. Then, we provided a computable function which translates each sentence of such a language into a logically equivalent sentence of the language underlying the theory $MSO[<]$. The translation was actually performed in two steps: we first embedded finitely-layered metric temporal structures into (flat) metric temporal structures; then, we reduced metric temporal structures to $MSO[<]$ structures. Hence, in both the original work by Alur and Henzinger and the above mentioned extension to finitely layered temporal structures, the basic tool for proving decidability properties is the theory $MSO[<]$ and the basic engine is Büchi theorem.

In [MPP99a], we dealt with the more general case in which the underlying temporal structure consists of infinitely many temporal layers (ω-layered, k-refinable, metric temporal structures). We introduced a second-order language for ω-layered k-refinable metric temporal structures $MSO[<, \downarrow_0, \ldots, \downarrow_{k-1}]$, and showed how to interpret it over different classes of structures. We first considered the problem of deciding infinitely refinable structures, called *downward* *unbounded layered structures* (cf. Figure 3). Following a traditional stream, we proved that the theory of downward unbounded layered structures can be embedded in the classical monadic second-order theory of k-successors. Next, we focused on the case of temporal structures in which there is a finest temporal domain together with an infinite number of coarser and coarser domains, called *upward unbounded layered structures* (cf. Figure 1). Such structures are too expressive to be embedded in $MSO[<]$. Indeed, the monadic second-order theory of upward unbounded layered structures can be proved to be equivalent to $MSO[<]$ (properly) extended with a suitable function $flip$ over natural numbers. In [MP99], the resulting decidable theory $MSO[<, flip]$ has been proved to be the counterpart (in the style of Büchi Theorem) of the class of ω-languages which is accepted by binary tree systolic automata, which strictly includes the class of ω-regular languages.

2.3 Deductive mechanisms

Most inference systems for modal and temporal logic are defined in the style of sequent or tableaux calculi. As an alternative, a number of translation methods for modal and temporal logic into classical first-order logic have been proposed in the literature. Such methods allow one to use automated theorem provers for Predicate Calculus to implement modal and temporal theorem provers; furthermore, they have the advantage of being independent of the particular logic under consideration: a single theorem prover may be used for any translatable logic.

In [DMP95], we proposed a novel translation method for modal logic that maps modal formulae into set-theoretic terms, thus making it possible to successfully exploit the automated theorem-proving machinery for first-order set theories to implement derivability in modal logic. The basic idea is to represent any Kripke frame as a set, with the accessibility relation modeled using the membership relation $\in$. The method can be easily generalized to polymodal logics (such a generalization can be seen as a completely symmetric set-theoretic version of Thomason's technique to reduce frame validity in tense logic to that in modal logic). In [MP97], we showed how to adapt the set-theoretic translation for polymodal logics to support derivability in *MTL*. We first reformulated *MTL* as a sort of *Propositional Dynamic Logic* (*PDL*), where programs have been replaced by displacements. Unlike *PDL*, *MTL* does not encompass any operation corresponding to the *PDL* program (term) $?\phi$, which is mapped into an accessibility relation $R_{?\phi}$ whose definition depends on the considered model. This allows us to express the semantics of *MTL* in terms of standard frames instead of standard models (a standard model simply being a model based on a standard frame). Moreover, *MTL* has not infinitary operations, like the operation $(.)^*$ of *PDL*, but it has a richer finitary structure. Then, we interpreted the resulting *PDL*-like *MTL* as a polymodal logic with an infinite number of accessibility relations, each one corresponding to a different temporal displacement. Finally, we defined a suitable modification of the set-theoretic translation for finite polymodal logics to handle such an infinite number of accessibility relations.

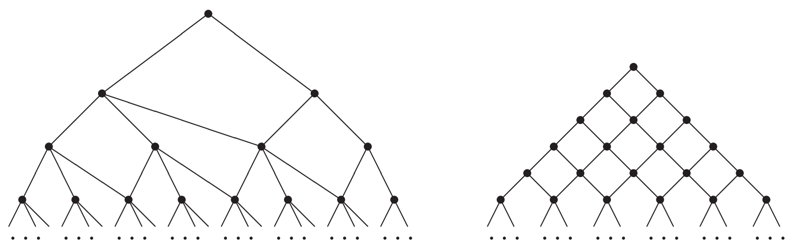

Figure 4: Structures for systolic Y-tree (a) and trellis (b) automata.

3 The present

3.1 Elementarily decidable temporal logics of time granularity

In [GPSS80], Gabbay et al. showed that propositional linear temporal logic (*PLTL*) can be viewed as an expressively complete and elementarily decidable modal temporal logic counterpart of $MFO[<]$ (the first-order fragment of $MSO[<]$). What we are looking for is a suitable extension of *PLTL* capable to play the same rôle with respect to $MFO[<, \downarrow_0, ..., \downarrow_{k-1}]$, interpreted over upward unbounded layered structures. To give a glimpse of the involved problems, we just list the main steps through which we plan to establish our result. First of all, a suitable finite base for the modal language must be identified and, to this end, the standard *until* operator must be replaced by a *qualified until* suitable for expressing truth in bounded intervals. Moreover, in order to correctly translate general formulae of the language $MFO[<, \downarrow_0, ..., \downarrow_{k-1}]$, a normal form theorem for formulae of such language must be proved. A major stumbling block at this point is to render the nesting of first-order quantifiers through the limited nesting capabilities of the modal language.

3.2 Weakening and strengthening the layered structure

We are investigating the possibility of weakening and strengthening the interconnections between the differently-grained domains of ω-layered metric structures. As far as weakening is concerned, a requirement that can be released is the one that constrains each time point to have exactly k children uniformly in each layer. For example, keeping the number of children constant within each layer, but allowing it to change across layers, would make it possible to deal with temporal structures including the domains of days, hours, and minutes. Let us define the *refinement degree* of a given layer as the number of children of each point belonging to the layer. It is not difficult to show that decidability is preserved if the map that associates a refinement degree with each layer is constant, except for a finite number of layers. A more general case to be considered is that of a map which is (ultimately) periodic.

As for the quest of more expressive structures, let us recall the well-known problem of the weeks/months refinement relationship: weeks stand naturally at a finer level of granularity than months, but there are months which do not consist of a whole number of weeks, that is, there are weeks which overlap the boundary between two consecutive months. As upward unbounded layered structures correspond to systolic automata over k-ary (infinite) trees, it seems promising to model relationships such as the weeks/months one by means of richer structures corresponding to systolic automata over Y-*trees* and *trellises* (cf. Figure 4). In [MP98b], initial results on the logical counterparts of ω-languages accepted by Y-tree and trellis automata are established. Such counterparts are proper extensions of $MSO[<, flip]$.

4 The future

As for the expressiveness of the proposed logics for time granularity, it should be noticed that lifting the same predicate from one level to another is not the only issue. As pointed out in [vB98], it is often easy to find connections between mathematical ordering structures (such a precedence or inclusion) going from one level to another. But as for predicates with factual content, the general case seems to be that each level has its own natural predicates and that we should study their connections. The proposed axiomatization constrains the interpretation domain to remain unchanged under temporal displacement and projection. Such constraints must be relaxed to deal with the general case.

As for the strengthening of the layered structure, observe that the infinitely layered structures considered so-far are either upward or downward unbounded. A natural generalization of them are structures which are both upward and downward unbounded. We are also considering the possibility of dealing with non-homogeneous structures, e.g., structures encompassing both discrete and dense domains. Do the results obtained in the upward and downward cases generalize to these classes of structures? Is the machinery employed (systolic and Rabin tree automata) sufficient?

As for the automata-theoretic side, it is well-known that $MFO[<]$ corresponds to star-free regular (ω-)languages. In [MP98a], the counterpart of $MFO[<, flip]$ with respect to finite models has been provided. We are looking for such a counterpart in the case of ω-languages.

References

[AH93] R. Alur and T.A. Henzinger. Real-time logics: complexity and expressiveness. *Information and Computation*, 104:35–77, 1993.

[vB95] J. van Benthem. Temporal logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. IV*, pages 241–350. Oxford University Press, 1995.

[vB98] J. van Benthem. Points on time. *Electronic News Journal on Reasoning about Actions and Change*, 2, 1998.

[Bur84] J.P. Burgess. Basic tense logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Vol. II*, pages 89–133. D. Reidel Pub. Comp., Dordrecht-Holland, 1984.

[CCMP93] E. Ciapessoni, E. Corsetti, A. Montanari, and P. San Pietro. Embedding time granularity in a logical specification language for synchronous real-time systems. *Science of Computer Programming*, 20(1-2):141–171, 1993.

[DMP95] G. D'Agostino, A. Montanari, and A. Policriti. A set-theoretic translation method for polymodal logics. *Journal of Automated Reasoning*, 15:317–337, 1995.

[Em90] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. B*, pages 995–1072. Elsevier Science Publishers B.V., 1990.

[FG96] M. Finger and D. Gabbay. Combining temporal logic systems. *Notre Dame Journal of Formal Logic*, 37:204–232, 1996.

[GPSS80] D. Gabbay, A. Pnueli, S. Shelah, and Stavi J. On the temporal analysis of fairness. In *Proc. 7th ACM Symposium on Principles of Programming Languages*, pages 163–173, 1980.

[MP95] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems (Safety).* Springer, 1995.

[MdR97] A. Montanari and M. de Rijke. Two-sorted metric temporal logic. *Theoretical Computer Science*, 183:187–214, 1997.

[Mon96] 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.

[MP96] A. Montanari and A. Policriti. Decidability results for metric and layered temporal logics. *Notre Dame Journal of Formal Logic*, 37:260–282, 1996.

[MP97] A. Montanari and A. Policriti. Executing metric temporal logic. In C. Brzoska and M. Fisher, editors, *Proc. of the IJCAI'97 Workshop on Programming in Temporal and Non Classical Logics*, 1997.

[MPP99a] A. Montanari, A. Peron, and A. Policriti. Theories of ω-layered metric temporal structures: expressiveness and decidability. *The Logic Journal of IGPL*, 7(1), 1999.

[MPP99b] A. Montanari, A. Peron, and A. Policriti. The taming (timing) of the states. *Submitted for Publication*, 1999.

[MP99] A. Monti and A. Peron. Systolic tree ω-languages: the operational and the logical view. *Theoretical Computer Science*, 1999. To appear.

[MP98a] A. Monti and A. Peron. A logical characterization of systolic languages. Proc. of the International Symposium on Theoretical Aspects of Computer Science (STACS'98), LNCS 1373, pp. 466–476, 1998.

[MP98b] A. Monti and A. Peron. Towards a logical characterization of systolic ω-languages. Research Report RR1798, Dipartimento di Matematica e Informatica, Università di Udine, 1998.

[Tho90] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. B*, pages 133–191. Elsevier Science Publishers, 1990.