Dov M. Gabbay, Ian Hodkinson, and Mark Reynolds, *Temporal Logic. Mathematical Foundations and Computational Aspects*, Volume 1, Oxford Science Foundations, Oxford, Clarendon Press, 1994, xii + 653 pp.

Reviewed by

VLADIK KREINOVICH

Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
email: vladik@cs.utep.edu

1. Why temporal logic?

What is temporal logic? The term "temporal logic" means "logic that describes knowledge about time".

Is it mathematics for mathematics' sake, or do we really need a special logic to describe time? The above description of temporal logic does not explain whether we need such a logic at all. Often a reader who encounters a technical paper on temporal logic with lots of technical results and no motivations in it, can get the idea that temporal logic is pure mathematics, generalization for generalization's sake that mathematicians are known to do a lot.

Well, there may be some papers on temporal logic whose authors do exactly this, but one of the main (and successful) goals of the book is to explain that many results from temporal logic are indeed useful.

At first glance, first order logic is quite sufficient, and no special logic is needed to describe temporal knowledge. A century of using traditional logic has shown that first-order logic is, in principle, sufficient to describe practically any part of human knowledge. In particular, we can use first order logic to describe practically every statement about time. For example, the truth of the temporal statement "B will be true" at a moment of time t can be described as $\exists s(s > t \ \& \ B_{|s})$ (where $B_{|s}$ means that B is true at the moment s). This description is what we professors teach the students in classes on logic and Artificial Intelligence. So why is a special logic necessary to describe temporal knowledge? This question is raised (and answered) in the very first chapter of the book.

First of all, if nothing else, temporal logic is a useful shorthand for describing knowledge about time. The first hint that first order logic may not be the best way to describe temporal knowledge comes from the fact that even reasonably simple phrases, quite understandable to any native speaker, are translated into clumsy and unreadable formulas of first order logic. In fact, they can be so unreadable that asking a student to translate a given statement into plain English becomes a problem requiring lots of effort (and worth quite a few points on the test). From the viewpoint of readability, it is definitely desirable to use special notations for the typical temporal constructions, such as *FA* for "*A* will be true", *PA* "*A* was true", etc.

Shorthand is not yet a new logic. We can, of course, call the logic, enriched by allowing these abbreviations, *temporal logic*. But if we simply view these abbreviations as shorthand for longer formulas of first order logic, we are, in actuality, not yet introducing a new logic.

These "shorthand" denotations do lead to a new logic after all: they help to uncover new decidable classes by reformulating first order statements in a propositional form. It turns out that these new notations can help us much more than simply giving us a useful shorthand writing.

This help is of the same type that lies behind other logics that can also be, in principle, reduced to first order logic, such as model logics, logics of knowledge and belief, etc. Namely, in traditional logic there is a big gap between *propositional* logic, that is *decidable* (i.e., for which we can always check whether a given statement is always true), and a *predicate* logic for which no deciding procedure exists (i.e., for which no universally applicable checking algorithm exists).

If our knowledge is described in terms of propositional formulas, and we have a query that is also expressed as a formula of propositional logic, then we can automatically check:

- whether the query follows from our knowledge (so that the answer to this query is "true"), or

- its negation follows from the knowledge (in this case, the answer to this query is "no"), or

- neither the query, not its negation follow from the knowledge; in this case, the answer to the query is "unknown".

If our knowledge is described by formulas of first order logic, with quantifiers, then no such general algorithm is possible. This does not mean that the situation is hopeless:

- we can use an automatic theorem prover to try and prove either the query or its negation;

- We can translate the knowledge and the query into Prolog and try to use a Prolog compiler; etc.

These methods are often successful, but in general, no algorithm is possible to check whether the answer to the given query should be "true", "false", or "unknown".

The translation mentioned above translates *propositional* formulas of temporal logic (i.e., formulas that do not use quantifiers) into formulas of predicate logic *that do* use quantifiers. Our hope (and help) comes from the fact that although the resulting formulas do use quantifiers, they use them in a special way. So, maybe, if we restrict ourselves only to the resulting *special* formulas of first order logic, we would be able to find a deciding algorithm.

This hope turns out to be justified: there are many decidable temporal logics (i.e., decidable fragments of first order logic).

Even undecidable fragments of temporal logic are often useful. Moreover, even when the resulting logical fragment is not decidable, it may still be advantageous to use the shorthand notations of temporal logic rather than the resulting first order formulas: there often exist heuristics specifically tailored to the resulting fragment of first order logic that work much better on this fragment that any known general first order logic heuristics. For example:

- when we represent a simple temporal knowledge base in Prolog using the shorthand temporal notations, the resulting logic program can be quite treatable by a given Prolog implementation, while

- the translation with explicit time leads to properties with additional arity and to additional rules that not only make Prolog work longer, but often push the resulting logic program outside the scope of the existing implementation.

Temporal logic is even more useful than it may seem at first glance. The difference between simple shorthand knowledge and first-order translation becomes even more drastic if we take into consideration the fact that to make the translation adequate, we also need to describe the first order

theory that describes the moments of time. In some problems, it is sufficient to describe this theory as linear order, but it is often important to know, e.g., that this order is *dense* (i.e., that for every two moment of time $t < s$, there exists a third moment of time x that lies in between the given two: $t < x < s$). We may also want to consider not a simple linear sequence of events, but all possible alternatives of possible decisions; in this case, we want *branching* time.

2. The main problems of temporal logic.

The main problem: in brief. In view of the justification provided above, the main problem of temporal logic is: *to find out what temporal connectives we can add to the existing propositional logic in order to keep it decidable* (or, at least, to make it easier to decide).

The main problems: a more detailed description. We have just explained the main idea of temporal logic. When we describe this main problem in detail, we will see that this *main problem* can be described as a sequence of several important problems. So, in the more detailed description, we will talk about main problems of temporal logic. Let us start this description:

- First, we want to describe what type of time we want (linear, branching, etc.).

- Second, we must specify with what logic we want to start.

The simplest choice is to start with the classical propositional logic. However, other choices are also necessary.

  - First, as we have mentioned above, the same need for simplification leads to other non-classical logics such as modal logic, etc.

  - Also, in view of the above, we may want to add some predicate material to the basic logic that we want to "temporalize".

- Third, we must describe exactly temporal connectives in which we are interested.

  - In some cases, we can simply write down the short list of connectives having real interest.

  - In other cases, we are interested in so many possible temporal connectives that it is practically impossible to independently analyze

and implement all of them. In this case, the problem becomes: how to describe the *basic* ones, in terms of which we can describe all the others (and is it possible to find such basic connectives at all)?

• Fourth, we must find out whether the resulting temporal logic is decidable.

• Finally, we must describe how to decide these statements without "unfolding" them back into the first order logic. For this, we at least need to *axiomatize* the resulting logic, i.e., describe a set of axioms from which each true statement can be derived. If such a representation is known, then we will be able to apply automatic theorem provers.

These problems are discussed in different chapters of the book. Let us briefly describe these problems one by one.

3. What model of time should we use?

Main possible models of time. The main possible models of time are described in Chapter 2:

• If we want to describe the actual sequence of events, then we must consider linear time, in which every two events are ordered.

• If, on the other hand, we are interested in *control* or *planning*, and we want to consider several possible alternatives, then we must consider *branching time*, in which a moment of time describes not only the physical time, but also what alternative we have chosen. If the choice was made at a moment 0, then, dependent on the choice, we will have different moments of this "model" time that correspond, e.g., to the physical time $t = 1$. These moments of "model time" are not ordered, so we do not have a linear order any more.

For each of these two choices, we also have the choice between continuous and discrete time:

• If we are interested in the state of the real world, then normally, we need *continuous* time that can take any real value.

• If, however, we are describing the dynamics of a discrete system (e.g., the computer), especially a system that has the internal digital clock controlling its operations, then we do not need to know the state at *every* possible moment of time, because no changes occur between

the ticks of this clock. We only need to consider moments of time that corresponding to the first, second, third, etc., ticks. In other words, we only need to consider *integer-valued* moments of time.

• When we describe a periodic process with some period T, then again, we do not need to use all moments of time, because the states of the process at t and $t + T$ are absolutely identical. For such systems, we must consider *cyclic* time.

Possibility of other models of time. These are the major possibilities, but they do not exhaust all possible time models: in other real-life situations, we have to consider more complicated models of time.

For example, usually, we have only a partial knowledge about the timing of an event: we do not know its exact moment of time t, but we know the approximate timing $\tilde{t}$ and the accuracy Δ of this approximation. From this information, we can only conclude that the (unknown) moment of time t belongs to the *interval* $[t^-, t^+]$, where $t^- = \tilde{t} - \Delta$ and $t^+ = \tilde{t} + \Delta$. In this case, the ordering of events becomes only a *partial* ordering: Namely:

• If two events are dated by real *numbers* t and s, then either $t \le s$ (t precedes or is simultaneous to s), or $s \le t$ (s precedes or is simultaneous to t).

• If, on the other hand, two events are characterized by *intervals* $[t^-, t^+]$ and $[s^-, s^+]$, then we have *three* possibilities:

  • the first event definitely precedes or is simultaneous to the second one ($t^+ \le s^-$);

  • the second event definitely precedes (or is simultaneous to) the first event ($s^+ \le t^-$);

  • we do not know which events was first: e.g., if we know that both events occured around time $\tilde{t} = 1$, and $\Delta = 0.5$, then $[t^-, t^+] = [s^-, s^+] = [0.5, 1.5]$ and we do not know which event was first.

This *interval* ordering was first considered in [Wiener *1914*] and [Wiener *1921*]; the serious use of interval ordering for knowledge representation started with Allen's paper [Allen *1983*] (for the latest developments, see,

e.g., [Allen *1991*]). Interval ordering is just one example of possible *partial* orderings of time.

4. What logics should we temporalize?

We have already mentioned that the authors' main effort is devoted to temporalizing (traditional) propositional logic. Often, however, propositional logic is not sufficient to describe the desired quantity. In such situations, more complicated logics are used, such as modal logics, etc.

Metalogics. In some cases, it is important to distinguish between the logic itself and the metalogic used to make statements about this logic. The resulting pair of logic and metalogic can be viewed as a two-tier logic. Temporalization of such logics is described in Chapter 5.

Logics with propositional quantifiers. One of the main applications of temporal logics is reasoning about programs. Many programs use *recursive* constructions in their definitions, and recursions in the way they work. For example, in a programming language we often have a recursive definition of a datatype, a formula, etc. It is known that such recursive definitions are not always describable by propositional (or even sometimes first order) logic; they require additional constructions such as quantifiers over propositions or the explicit *(smallest) fixed point* construction. These constructions are widely used in the semantics of logic programming (there, fixed points are a standard tool, and quantifiers over propositions are used in the propositional case of the so-called *circumscription*).

For such logics, temporalization problems are analyzed in Chapter 8. In general, the resulting temporal logics are undecidable, but many of them are, nevertheless, useful.

Temporalizing the general first order logic is an ideal objective, but (so far) practically hopeless. Ideally, we would like to use the full first order logic when talking about time. However, it turns out that when we add discrete time to first order logic, the resulting theory becomes so rich that arithmetic can be described in it (see Chapter 4). There are two main consequences of this richness:

- From the *logical* viewpoint, the resulting theory becomes not only undecidable, but incomplete as well.

- From the *practical* viewpoint, the fact that we can represent an arbitrary arithmetic formula makes automatic theorem proving practically hopeless.

5. In what temporal connectives are we interested?

The simplest connectives. In addition to the above-described F and P, there are several other important temporal connectives, such as:

- NA ("A is true *now*"); this statement is true at a moment t iff A is true for all moments of time from some open interval (t^-, t^+) that contains t.

- $U(A, B)$ ("B will be true until A is true"); this statement is true at a moment t iff $\exists s\ (s > t\ \&\ A_{|s}\ \&\ \forall u\ (t < u < s \rightarrow B_{|u}))$.

- $S(A, B)$ ("B has been true since A was true"); this statement is true at a moment t iff $\exists s\ (s < t\ \&\ A_{|s}\ \&\ \forall u\ (s < u < t \rightarrow B_{|u}))$.

For continuous time, these definitions capture (to a certain extent) the intuitive meanings of the terms "until" and "since". For discrete time (or, in general, for time "with gaps"), it may happen that there is no moment of time s that separates, say, A from B; in this case, more complicated definitions are needed. Corresponding connectives have been proposed; they are called *Stavi connectives* (by the name of their inventor) and denoted by U' and S'.

More complicated connectives. Phrases from natural language can describe very complicated temporal relations. A natural question is: do we need to design a new temporal logic connective for each such relation, or it is possible to express all possible temporal connectives in terms of the few selected ones (just like in classical propositional logic, where we can express an arbitrary logical relation in terms of, say, conjunction and negation)? In other words, is there a small *expressively complete* set of connectives?

It turns out (see Chapters 9–13) that:

- For linear continuous time, the connectives "until" and "since" (U and S) described above are expressively complete.

- For general linearly ordered time, we need to add Stavi's versions of "until" and "since" to get an expressively complete set.

- In general, there exist logics for which no finite set of temporal connectives is expressionally complete: e.g., the temporal logic that describes cyclic time.

The smallest number of connectives in an expressionally complete set is called the *Henkin dimension* (or simply *H-dimension*, for short) of a temporal logic. A simple syntactic upper bound for *H*-dimension is known: if we can express the corresponding first order theory of time moments by using only k different variables in each formula, then *H*-dimension cannot exceed K. A warning: this syntactic criterion does not provide us with a final solution to the problem, because H-dimension may be actually smaller than k.

Multi-dimensional connectives. The above described temporal connectives F, P, etc., are *one-dimensional* in the sense that each of the resulting statements FA, etc., is true or false depending on a single time value t. Not all possible connectives are one-dimensional: some statements from the natural language require at least two different moments of time to describe. For example, to check whether a statement "A was true during a certain time interval" is true, we must fix *two* moments of time: the endpoints of that interval.

In some cases, we can represent the same knowledge using only one-dimensional properties: e.g., the above description can be reduced to the knowledge of A_t for different moments of time t. However, in other cases, such a reduction is impossible: e.g., in planning, we have to consider several possible future trajectories, and therefore, consider statements of the type: "by moment t, it was still possible to achieve the given goal by the year s". Such *truly multi-dimensional* logics are analyzed in Chapter 7.

It is interesting to mention that if we allow multi-dimensional connectives, then for *some* (but not for *all*) temporal logics, it will be possible to find an expressionally complete set of temporal connectives: e.g., such a set becomes possible for the theory of cyclic time.

6. When is the resulting temporal logic decidable?

General results. In Chapter 14, a general method of temporalizing logics is described, and general theorems are proven, that under certain reasonable conditions:

- if the original logic was decidable, then the resulting temporalized logic is decidable as well;
- if the original logic was complete, then the resulting temporalized logic is also complete;

- temporalization is a *conservative* extension in the sense that a statement from the original logic is provable in the temporalization iff it was originally provable.

Temporalization corresponds, crudely speaking, to adding all possible one-dimensional temporal connectives. To describe two-dimensional connectives, we can apply temporalization to the temporalized logic; if we apply temporalization once again, we get 3-D connectives, etc.

Specific results. The results of applying these techniques to main temporal logics are described in Chapter 15.

Main tool for proving decidability. One of the main tools for proving decidability of temporalized logic is decidability of weak second order theory.

7. How to axiomatize a temporal logic?

Different axiomatization techniques are described in Chapter 3. A general theory that describes these techniques and their results is described in Chapter 6.

Conclusion.

This book is a wonderful introduction to the area, rich in technical details, and at the same time, rich in motivations. Many specific problems of temporal reasoning are promised in the second volume that is still to come, e.g., *interval temporal logic* in which the main notion is not "a property is true at a given moment of time", but rather "a property is true for all moments of time from a given time interval". Together, these two volumes will be a must for a logician who has ever been interested in time.

References.

The volume is equipped with a comprehensive bibliography. The only references that we included here are about *interval time,* because we mention it in our review, and this part of temporal logic is not covered in Volume 1.

ALLEN, J. F. 1983. *Maintaining knowledge about temporal intervals,* Communications of the ACM **26** (No. 11), 832–843; reprinted in: Peter G.

Raeth (editor), *Expert systems: A software methodology for modern applications* (Los Alamitos, CA, IEEE Computer Society Press, 1990), 248–259.

ALLEN, J. F., H. A. KAUTZ, R. N. PELAVIN, and J. D. TENENBERG. 1991. *Reasoning about plans*, San Mateo, CA, Morgan Kaufmann.

WIENER, N. 1914. *A contribution to the theory of relative position*, Proc. Cambridge Philos. Soc. **17**, 441–449.

—. 1921. *A new theory of measurement: a study in the logic of mathematics*, Proceedings of the London Mathematical Society **19**, 181–205.