Application of temporal logics

Ihor Volokhovych, MMAI-1

Temporal logic is a branch of formal logic which is of particular interest for the specification and verification of concurrent systems including communication services and protocols. This tutorial presents the principles of temporal logic and explains how it can be applied.

In logic, temporal logic is any system of rules and symbolism for representing, and reasoning about, propositions qualified in terms of time (for example, "I am always hungry", "I will eventually be hungry", or "I will be hungry until I eat something"). It is sometimes also used to refer to tense logic, a modal logic-based system of temporal logic introduced by Arthur Prior in the late 1950s, with important contributions by Hans Kamp. It has been further developed by computer scientists, notably Amir Pnueli, and logicians.

Temporal logic has found an important application in formal verification, where it is used to state requirements of hardware or software systems. For instance, one may wish to say that whenever a request is made, access to a resource is eventually granted, but it is never granted to two requestors simultaneously. Such a statement can conveniently be expressed in a temporal logic.

Motivation

Consider the statement "I am hungry". Though its meaning is constant in time, the statement's truth value can vary in time. Sometimes it is true, and sometimes false, but never simultaneously true and false. In temporal logic, a statement can have a truth value that varies in time—in contrast with an atemporal logic, which applies only to statements whose truth values are constant in time. This treatment of truth-value over time differentiates temporal logic from computational verb logic.

Temporal logic always has the ability to reason about a timeline. So-called linear "time logics" are restricted to this type of reasoning. Branching logic, however, can reason about multiple timelines. This presupposes an environment that may act unpredictably. To continue the example, in a branching logic we may state that "there is a possibility that I will stay hungry forever", and that "there is a possibility that eventually I am no longer hungry". If we do not know whether or not I will ever be fed, these statements can both be true.

History

Although Aristotle's logic is almost entirely concerned with the theory of categorical syllogism, there are passages in his work that are now seen as anticipations of temporal logic, and may imply an early, partially developed form of first-order temporal modal binary logic. Aristotle was particularly concerned with the problem of future contingents, where he could not accept that the principle of bivalence applies to statements about future events, i.e. that we can presently decide if a statement about a future event is true or false, such as "there will be a sea battle tomorrow".

There was little development for millennia, Charles Sanders Peirce noted in the 19th century. Time has usually been considered by logicians to be what is called 'extralogical' matter. I have never shared this opinion. But I have thought that logic had not yet reached the state of development at which the introduction of temporal modifications of its forms would not result in great confusion; and I am much of that way of thinking yet.

Surprisingly for Pierce, the first system of temporal logic was constructed, as far as we know, in the first half of the 20th century. Although Arthur Prior is widely known as a founder of temporal logic, the first formalization of such logic was provided in 1947 by Polish logician, Jerzy Łoś. In his work Podstawy Analizy Metodologicznej Kanonów Milla (The Foundations of a Methodological Analysis of Mill’s Methods) he presented a formalization of Mill's canons. In Łoś.' approach emphasis was placed on the time factor. Thus, to reach his goal, he had to create a logic which could provide means for formalization of temporal functions. The logic could be seen as a byproduct of Łoś' main aim, albeit it was the first positional logic which, as a framework, was used later for Łoś' inventions in epistemic logic. The logic itself has syntax very different from Prior's tense logic, which uses modal operators. Language of Łoś' logic rather uses a specific for positional logic, realization operator that binds the expression with the specific context in which its truth-value is considered. In Łoś' work this considered context was only temporal, thus expressions were binded with specific moments or intervals of time.

In the following years, research of temporal logic by Arthur Prior began. He was concerned with the philosophical implications of free will and predestination. According to his wife, he first considered formalizing temporal logic in 1953. Results of his research were firstly presented at the conference in Wellington in 1954. The system Prior presented was similar syntactically to Łoś' logic, although not until 1955 he explicitly referred to Łoś' work, in the last section of Appendix 1 in Prior's Formal Logic.

Prior gave lectures on the topic at the University of Oxford in 1955–6, and in 1957 published a book, Time and Modality, in which he introduced a propositional modal logic with two temporal connectives (modal operators), F and P, corresponding to "sometime in the future" and "sometime in the past". In this early work, Prior considered time to be linear. In 1958 however, he received a letter from Saul Kripke, who pointed out that this assumption is perhaps unwarranted. In a development that foreshadowed a similar one in computer science, Prior took this under advisement, and developed two theories of branching time, which he called "Ockhamist" and "Peircean".[clarification needed] Between 1958 and 1965 Prior also corresponded with Charles Leonard Hamblin, and a number of early developments in the field can be traced to this correspondence, for example Hamblin implications. Prior published his most mature work on the topic, the book Past, Present, and Future in 1967. He died two years later.

Along with tense logic, Prior constructed a few systems of positional logic, which inherited main ideas from Łoś.[6] Work in positional temporal logics was continued by Nicholas Rescher in the 60's and 70's. In such works as Note on Chronological Logic (1966), On the Logic of Chronological Propositions (1968), Topological Logic (1968), Temporal Logic (1971) he researched connections between Łoś' and Prior's systems. Moreover he proved that Prior's tense operators could be defined using the realization operator in specific positional logics.[6] Rescher in his work also created more general systems of positional logics. Although the first ones were constructed for purely temporal uses, he proposed a term of topological logics that were meant to contain a realization operator but had no specific temporal axioms - like the clock axiom.

The binary temporal operators Since and Until were introduced by Hans Kamp in his 1968 Ph.D. thesis, which also contains an important result relating temporal logic to first-order logic—a result now known as Kamp's theorem.

Two early contenders in formal verifications were linear temporal logic, a linear time logic by Amir Pnueli, and computation tree logic, a branching time logic by Mordechai Ben-Ari, Zohar Manna and Amir Pnueli. An almost equivalent formalism to CTL was suggested around the same time by E. M. Clarke and E. A. Emerson. The fact that the second logic can be decided more efficiently than the first does not reflect on branching and linear logics in general, as has sometimes been argued. Rather, Emerson and Lei show that any linear logic can be extended to a branching logic that can be decided with the same complexity.

Application

Applications of Temporal Logic include its use as a formalism for clarifying philosophical issues about time, as a framework within which to define the semantics of temporal expressions in natural language, as a language for encoding temporal knowledge in artificial intelligence, and as a tool for specification, formal analysis, and verification of the executions of computer programs and systems.

Temporal logic is a field whose development has been heavily driven by philosophical considerations. At the same time, the logical formalisms and technical systems developed over the years have found application in various different disciplines, ranging from computer science, artificial intelligence, and linguistics, to natural, cognitive, and social sciences. In this section, we briefly discuss some pertinent applications of temporal logics in computer science, artificial intelligence, and linguistics.

Temporal logics in Computer Science

The idea to apply temporal reasoning to the analysis of deterministic and stochastic transition systems was already present in the theory of processes and events in Rescher and Urquhart (1971, Chapter XIV). However, it was with the seminal paper of Pnueli (1977) that temporal logic became really prominent in computer science. Pnueli proposed the application of temporal logics to the specification and verification of reactive and concurrent programs and systems. In order to ensure correct behavior of a reactive program, in which computations are non-terminating (e.g. an operating system), it is necessary to formally specify and verify the acceptable infinite executions of that program. In addition, to ensure correctness of a concurrent program, where two or more processors are working in parallel, it is necessary to formally specify and verify their interaction and synchronization.

Key properties of infinite computations that can be captured by temporal patterns are liveness, safety, and fairness (see Manna and Pnueli 1992):

- *Liveness* properties or *eventualities* involve temporal patterns of the forms **Fp,q→Fp**, or **G(q→Fp)** which ensure that if a specific precondition (**q**) is initially satisfied, then a desirable state (satisfying **p**) will eventually be reached in the course of the computation. Examples are "If a message is sent, it will eventually be delivered" and "Whenever a printing job is activated, it will eventually be completed".
- Safety or invariance properties involve temporal patterns of the forms **Gp, q→Gp**, or **G(q→Gp)**, which ensure that if a specific precondition (**q**) is initially satisfied, then undesirable states (violating the safety condition **p**) will never occur. Examples are: "No more than one process will be in its

critical section at any moment of time", "A resource will never be used by two or more processes simultaneously", or, to give more practical examples: "The traffic lights will never show green in both directions", "A train will never pass a red semaphore".

- Fairness properties involve combinations of temporal patterns of the forms **GFp** ("infinitely often **p**") or **FGp** ("eventually always **p**"). Intuitively, fairness requires that whenever several processes that share resources are run concurrently, they must be treated 'fairly' by the operating system, scheduler, etc. A typical fairness requirement says that if a process is persistent enough in sending a request (e.g. keeps sending it over and over again), its request will eventually be granted.

An infinite computation is formally represented by a model of the linear time temporal logic LTL. Non-deterministic systems are modeled by branching time structures. Thus, both LTL and the computation tree logic CTL and CTL* have been very important for specification and verification of reactive and concurrent systems.

The following example combines liveness and safety properties of a single computation: "Whenever a state of alert is reached, the alarm is activated and remains activated until a safe state is eventually reached". This property is expressible in LTL as

$$\mathbf{G(alert \rightarrow (alarm\ U\ safe)).}$$

Another example, referring to all computations in the system, is: "If the process σ is eventually enabled on some computation starting from the current state, then on every computation starting there, whenever σ is enabled, it will remain enabled until the process τ is disabled". This property can be formalized in CTL* as

$$\Diamond \mathbf{F\ enabled\ \sigma \rightarrow \Box G(enabled\ \sigma \rightarrow (enabled\ \sigma\ U\ disabled\ \tau)).}$$

A variation of LTL with useful applications for specifying and reasoning about concurrent systems is Lamport's (1994) temporal logic of actions TLA. Other applications of temporal logics in computer science include: temporal databases, real-time processes and systems, hardware verification, etc. Further information on such applications can be found in e.g. Pnueli (1977); Emerson and Clarke (1982); Moszkowski (1983); Galton (1987); Emerson (1990); Alur and Henzinger (1992); Lamport (1994); Vardi and Wolper (1994); Bolc and Szalas (1995); Gabbay et al. (2000); Baier and Katoen (2008); Kröger and Merz (2008); Fisher (2011); Demri et al. (2016).

Temporal logics in Artificial Intelligence

Artificial Intelligence (AI) is one of the major areas of application of temporal logics. Relating temporal reasoning to AI was suggested already in the early philosophical discussion on AI by McCarthy and Hayes (1969), the theory of processes and events in Rescher and Urquhart (1971, Chapter XIV), and the period-based theories of Hamblin (1972); see Øhrstrøm and Hasle (1995) for an overview of these early developments. In the 1980s, temporal representation and reasoning gradually became an increasingly prominent theme in AI with several influential works, including McDermott's temporal logic for reasoning about processes and plans (McDermott 1982); Allen's general theory of action and time (Allen 1984); the Event Calculus of Kowalski and Sergot (1986); the reified temporal logic by Shoham (1987); the logic of time representation by Ladkin (1987); and the work on temporal database management by Dean and McDermott (1987). Galton (1987) provides a systematic account of these and other important developments in that period; see also Vila (1994) and Pani and Bhattacharjee (2001) for comprehensive reviews. Influential works in the 1990s include the introduction of interval-based temporal logics by Halpern and Shoham (1991) and by Allen and Ferguson (1994), with representation of actions and events; the Situation Calculus of Pinto and Reiter (1995); and Lamport's Action Theory (Lamport 1994), etc.

Further important developments relating temporal reasoning and AI since then include: temporal reasoning in natural language, temporal ontologies, temporal databases and constraint solving, temporal planning, executable temporal logics, spatial-temporal reasoning, temporal reasoning in agent-based systems, etc. The field has gradually grown so rich and broad — as witnessed by the 20-chapter handbook edited by Fisher et al. (2005) — that it is impossible to even briefly survey it here, so we only bookmark a few of the main issues of philosophical relevance that have been in the focus of temporal reasoning and logics in AI.

- The prevailing logical approach to temporal representation and reasoning in AI, especially in the 1980s-1990s, has traditionally been based on temporalized variations of first-order logic rather than on Prior-style temporal logic. The approach is best illustrated by the so-called *method of temporal arguments* (McCarthy and Hayes 1969; Shoham 1987; Vila 1994). According to this method, the temporal dimension is captured by augmenting propositions and predicates with 'time stamp' arguments, as for example "Publish(A. Prior, *Time and Modality*, 1957)". An alternative, yet closely related approach, is the one of *reified temporal logics* (McDermott 1982; Allen 1984; Shoham 1987; see Ma and Knight 2001 for a survey). This approach makes use of reifying meta-predicates, such as 'TRUE' and 'FALSE', but also 'HOLDS', 'OCCURS', 'BEFORE', 'AFTER', interval relations such as 'MEETS', 'OVERLAPS', etc., which are applied to propositions of some standard logical language (e.g. classical first-order logic). An example of a reified expression is "OCCUR(Born(A. Prior), 1914)". Associated with theories of time are theories of *temporal incidence* (cf. Vila 2005). Still, the prominence of the modal logic based approach has always been strong, and it has more recently resurged, e.g. in the context of agent-based temporal reasoning (cf. Fisher and Wooldridge 2005).
- Theories of temporal reasoning in AI distinguish between *fluents*, which are propositions describing states of the world that may change over time, and *events*, representing what happens in the world and causes changes between states. Philosophical issues arising here concern the nature of fluents and events, the meaning of instantaneous events, the distinction between homogeneous states and inhomogeneous events, the *dividing instant problem*, the *frame problem*, etc. For further discussion, see Shoham (1987); Galton (1990); and Vila (2005). See also the related discussion on reasoning about action and change in Section 4 of the entry on logic and artificial intelligence.
- Both fluents and events can be considered in discrete or continuous time and they can be instantaneous or durational. This keeps the debate on

instant-based versus *interval-based* formal models of time alive, with various theories following and comparing both approaches, including van Benthem (1983); Allen (1983); Allen and Hayes (1989); Allen and Ferguson (1994); Galton (1995); Vila (2005); etc.

For further reading and discussion on temporal reasoning and logics in AI, see Vila (1994); Galton (1995); the comprehensive handbook Fisher et al. (2005); and the more concise handbook chapter Fisher (2008).