Comments on Temporal Logics for Real-Time System Specification

CARLO A. FURIA

Politecnico di Milano

MATTEO PRADELLA

Consiglio Nazionale delle Ricerche

and

MATTEO ROSSI

Politecnico di Milano

The article “Temporal Logics for Real-Time System Specification” surveys some of the relevant literature dealing with the use of temporal logics for the specification of real-time systems. Unfortunately, it introduces some imprecisions that might create some confusion in the reader. While a certain degree of informality is certainly useful when addressing a broad audience, imprecisions can negatively impact the legibility of the exposition. We clarify some of its remarks on a few topics, in an effort to contribute to the usefulness of the survey for the reader.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Temporal logic*; D.2.1 [**Software Engineering**]: Requirements/Specifications—*Languages*

General Terms: Languages

Additional Key Words and Phrases: Temporal logic

ACM Reference Format:

Furia, C. A., Pradella, M., and Rossi, M. 2009. Comments on temporal logics for real-time system specification. ACM Comput. Surv. 41, 2, Article 13 (February 2009), 5 pages. DOI = 10.1145/1459352.1459358 http://doi.acm.org/10.1145/1459352. 1459358

The article “Temporal Logics for Real-Time System Specification” [Bellini et al. 2000] surveyed some of the relevant literature dealing with the use of temporal logics for the specification of real-time systems. Unfortunately, Bellini et al. [2000] introduced some imprecisions that might create some confusion in the reader. While a certain degree of informality is certainly useful when addressing a broad audience, imprecisions can negatively impact the legibility of the exposition. We clarify some of the remarks of

Authors' addresses: C. A. Furia and M. Rossi, Dipartimento di Elettronica e Informazione, Politecnico di Milano, p.zza L. da Vinci 32, 20133 Milan, Italy; email: {furia, rossi}@elet.polimi.it; M. Pradella, IEITT, Consiglio Nazionale delle Ricerche, via Ponzio 34/5, 20133 Milan, Italy; email: pradella@elet.polimi.it.

Bellini et al. [2000] on a few topics, in an effort to contribute to the usefulness of the survey for the reader.[1]

Completeness and Soundness. Section 2.1 of Bellini et al. [2000] introduced the definitions of *completeness* and *soundness* of a deductive system, which are essentially tautologies. Instead, a deductive system $\mathcal{F}$ is *sound* when "every theorem of $\mathcal{F}$ is valid" [Andrews 1992, page 80] and it is *complete* when "every valid well-formed formula of $\mathcal{F}$ is a theorem" [Andrews 1992, page 94]. In Section 3.8 of Bellini et al. [2000] it is said that "it is never possible to build a *complete* deductive system." This statement is clearly false, as there exist numerous complete deductive systems for several logic languages, such as propositional logic [Andrews 1992] or, as Bellini et al. [2000] itself suggested elsewhere, PTL (propositional temporal logic).

Expressiveness. In the literature, two different meanings are associated with the term *expressiveness*. The first one, common in much of the literature on temporal logics (e.g., Alur and Henzinger [1993]; Emerson [1990]), refers to the ability to describe a class of properties. For example, the property "A until B" is not expressible with a logic which only uses the unary modalities *eventually* and *always* [Kamp 1968]. The second meaning, sometimes used informally, refers to the ease and simplicity with which one can specify some behavior with a given formalism. For example, Pascal is more expressive than Assembly language since it abstracts many details away and allows programmers to express algorithms in a more compact form. This notion has been formalized in programming languages [Fischer 1993]. Unless otherwise indicated, we will use *expressiveness* in the first sense.

Metric on Time. Section 3.4 of Bellini et al. [2000] claimed that temporal operators are qualitative as the $\bigcirc$ *next* does not involve an exact measure, and therefore to add a metric for time bounded operators are needed, for example, $\Diamond_{\leq 5}A$, which means that A will be true within 5 time units. This is not always true, since time can be interpreted to be isomorphic to natural numbers: $\bigcirc$ is then durational. With this assumption one can define, for instance, $\Diamond_{\leq 5}A \equiv \bigcirc(A \vee \bigcirc(A \vee \bigcirc(A \vee \bigcirc(A \vee \bigcirc A))))$, which shows that it is possible to give an exact measure of the elapsing time.

Logic Executability. Section 3.9 of Bellini et al. [2000] presented three different definitions of executability of temporal logic specifications. The first is said to correspond "to that of decidability of the validity problem" [Bellini et al. 2000, page 25], as noted in Moszkowski [1986]; the second corresponds to *history checking* [Felder and Morzenti 1994]; the third consists of using the system specification itself as a prototype or implementation of the system, thus allowing the *on-line* generation of system outputs on the basis of present inputs and its internal state and past history. Bellini et al. [2000] also stated that "there exists [sic] only few executable temporal logics that can be used to build a system prototype according to meaning (iii) of executability" (page 25). This was further stressed in that article's Table 7, where the column "Logic executability" summarized this feature for the considered logics. Executability of each logic is classified under one of the following five labels: "N=No, (N)=no in the general case, Y=yes, (Y)=yes in some specific case, NA=not available". Only two of the logics carry a "Y"; four carry a "(Y)," and the rest (the vast majority) carry an "NA." However, most temporal logics can be restricted to a proper subset that is isomorphic to basic temporal logic with finite domains only (except for the temporal domain, which is usually assumed denumerable). This subset is reducible to PTL, which is executable (Gabbay [1987]

[1] The interested reader may find more details in Furia et al. [2008].

and in accordance with Table 7 of Bellini et al. [2000]). Therefore most of the logics tagged "NA" are in fact "executable in some specific case" (and hence "(Y)"), regardless of whether an actual implementation of the execution algorithm for the specific subset has been provided.

Also, while Bellini et al. [2000] mentioned the issue of the "computational complexity of the algorithms" to execute temporal logics, it did not point out that this is what distinguishes the third definition from the two previous ones, that is, the possibility of *implementing* an *efficient* algorithm for building a model for a logic specification [Fisher and Owens 1993].

Past and Future. Bellini et al. [2000] stated that, whenever a temporal logic does not explicitly provide past operators, it is impossible to express requirements about the past in that logic. However, it is a well-established result that PTL (interpreted over time models isomorphic to the natural numbers, as customary) with both past and future operators is (initially) equivalent in expressive power to PTL with future operators only [Gabbay et al. 1980, 1994; Gabbay 1987; Emerson 1990]. The presence of explicit past operators does *facilitate* the writing of formulas about the past, and enhance their conciseness, but it is not strictly necessary.

Other passages in Bellini et al. [2000] maintained that the availability of past operators is not necessary—as far as the expression of requirements about the past is concerned—whenever the past is bounded. This is also not true in general, as there exist temporal logics that are strictly more expressive when endowed with past operators even when they are interpreted over structures with a bounded past. In fact, more recently Hirshfeld and Rabinovich [2003] proved that PTL with *until* only is strictly less expressive than its variety with *until* and *since*, over the nonnegative reals. Another example is MTL with the qualitative *since* operator, which is more expressive than its future-only variety, even if structures with a bounded past are adopted [Prabhakar and D'Souza 2006].

A Running Example. Bellini et al. [2000] informally introduced a simple example of real-time specification (Figure 6 of Bellini et al. [2000]), which was then formally specified with each of the considered metric temporal logics. The example was that of a predicate E whose occurrence triggers predicates $startA$ and $endA$ within t_e time units, thus marking an interval in which A is true. Bellini et al. [2000] in some cases presented formulas that were claimed to be equivalent. For example, consider the two TRIO formulas for the example presented in Section 4.14 of Bellini et al. [2000] (similar considerations can be made for the MTL formulas of Section 4.15):

(1) $Alw(E \rightarrow \exists t((0 < t < t_e) \wedge Futr(endA, t) \wedge WithinF(startA, t)))$;
(2) $Alw(E \rightarrow WithinF(endA, t_e) \wedge \neg Until(\neg startA, endA))$.[2]

Formula (2) is stronger than Formula (1), since (2) forces the first occurrence of $endA$ after E to be preceded by an occurrence of $startA$, while (1) does not.

Point- Versus Interval-Based Logics. Section 3.3 of Bellini et al. [2000] stated that "Interval-based temporal logics are more expressive [than point-based logics], since they are capable of describing events in time intervals, and a single time instant is represented with a time interval of one." As previously discussed, there are two different meanings of *expressiveness*, but it is not apparent what is the intended one in the sentence above.

[2] In (2) we have swapped the arguments of the *Until* to conform with TRIO's usual syntax.

If *expressiveness* in the formal sense is meant, the above claim is not correct. For example, over discrete time, every finite interval can be represented by a finite union of discrete points. The MTL logic [Koymans 1990] (a point-based formalism according to the taxonomy of Bellini et al. [2000], Table 7) is more expressive than the interval-based TILCO logic [Mattolini and Nesi 2001], since only the former allows explicit quantification over time variables. The introduction of temporal logics based on intervals, rather than points, has been supported essentially by claims of simplification in writing specifications, not for reasons of expressiveness [Emerson 1990; Koymans 1992].

If Bellini et al. [2000] referred to *expressiveness* in its informal sense, the statement above is too vague. To compare meaningfully the informal expressiveness of formalisms one should first establish that they have "similar" formal expressiveness. Otherwise, the comparison is irrelevant because the classes of properties they are capable of representing are too different.

Implicit and Explicit Time. Section 3.6 of Bellini et al. [2000] stated that "The explicit specification of time allows the specification of expressions that have no sense in the time domain—e.g., the activation of a predicate when the time is even." On the contrary, a property such as "predicate P occurs at all time instants that are multiple of a constant n" is of interest in timed systems (consider, for instance, the behavior of a counter, or the clock signal of a synchronous integrated circuit), and the impossibility of expressing such a property in PTL [Wolper 1983] spawned a number of PTL extensions. Emerson [1990] Section 6.1.1 showed that a second-order extension of PTL, in which it is possible to quantify over propositions and where time is still implicit, allows one to express the property above.

REFERENCES

ALUR, R. AND HENZINGER, T. A. 1993. Real-time logics: Complexity and expressiveness. *Inform. Computat. 104*, 35–77.

ANDREWS, P. B. 1992. *An Introduction to Mathematical Logic and Type Theory*. Academic Press, New York, NY.

BELLINI, P., MATTOLINI, R., AND NESI, P. 2000. Temporal logics for real-time system specification. *ACM Comput. Surv. 32*, 1 (Mar.), 12–42.

EMERSON, E. A. 1990. Temporal and modal logic. In *Handbook of Theoretical Computer Science*. Elsevier, Amsterdam, The Netherlands, 996–1072.

FELDER, M. AND MORZENTI, A. 1994. Validating real-time systems by history-checking TRIO specifications. *ACM Trans. Softw. Eng. Method. 3*, 4 (Oct.), 308–339.

FISCHER, M. J. 1993. Lambda-calculus schemata. *Lisp Symbol. Computat. 6*, 3/4, 259–288.

FISHER, M. AND OWENS, R. 1993. An introduction to executable modal and temporal logics. In *Proceedings of the Workshop on Executable Modal and Temporal Logics*.

FURIA, C. A., PRADELLA, M., AND ROSSI, M. 2008. Comments on 'Temporal logics for real-time system specification.' Tech. rep. 2008.7. DEI, Politecnico di Milano, Milan, Italy.

GABBAY, D. M. 1987. The declarative past and imperative future. In *Proceeding of TLS'87*. Lecture Notes in Computer Science, vol. 398. Springer, Berlin, Germany, 409–448.

GABBAY, D. M., HODKINSON, I., AND REYNOLDS, M. 1994. *Temporal Logic (vol. 1): Mathematical Foundations and Computational Aspects*. Oxford University Press, Oxford, U.K.

GABBAY, D. M., PNUELI, A., SHELAH, S., AND STAVI, J. 1980. On the temporal basis of fairness. In *Proceedings of POPL'80*. 163–173.

HIRSHFELD, Y. AND RABINOVICH, A. M. 2003. Future temporal logic needs infinitely many modalities. *Inform. Computat. 187*, 2, 196–208.

KAMP, J. A. W. 1968. Tense logic and the theory of linear order. Ph.D. dissertation. University of California at Los Angeles, Los Angeles, CA.

KOYMANS, R. 1990. Specifying real-time properties with metric temporal logic. *Real-Time Syst. 2*, 4, 255–299.

KOYMANS, R. 1992. (Real) Time: A philosophical perspective. In *Real-Time: Theory in Practice*. Lecture Notes in Computer Science, vol. 600. Springer, Berlin, Germany, 353–370.

MATTOLINI, R. AND NESI, P. 2001. An interval logic for real-time system specification. *IEEE Trans. Softw. Eng. 27*, 3 (Mar.), 208–227.

MOSZKOWSKI, B. 1986. *Executing Temporal Logic Programs*. Cambridge University Press, Cambridge, U.K.

PRABHAKAR, P. AND D'SOUZA, D. 2006. On the expressiveness of MTL with past operators. In *Proceedings of FORMATS'06*. Lecture Notes in Computer Science, vol. 4202. Springer, Berlin, Germany, 322–336.

WOLPER, P. 1983. Temporal logic can be more expressive. *Inform. Contr. 56*, 1, 72–99.

Received September 2005; revised July 2007, April 2008; accepted April 2008