Integrating Discrete- and Continuous-Time Metric Temporal Logics Through Sampling

Carlo A. Furia and Matteo Rossi

Dipartimento di Elettronica e Informazione, Politecnico di Milano
32, Piazza Leonardo da Vinci, 20133, Milano, Italy
{furia, rossi}@elet.polimi.it

Abstract. Real-time systems usually encompass parts that are best described by a continuous-time model, such as physical processes under control, together with other components that are more naturally formalized by a discrete-time model, such as digital computing modules. Describing such systems in a unified framework based on metric temporal logic requires to integrate formulas which are interpreted over discrete and continuous time.

In this paper, we tackle this problem with reference to the metric temporal logic TRIO, that admits both a discrete-time and a continuous-time semantics. We identify sufficient conditions under which TRIO formulas have a consistent truth value when moving from continuous-time to discrete-time interpretations, or vice versa. These conditions basically involve the restriction to a proper subset of the TRIO language and a requirement on the finite variability over time of the basic items in the specification formulas. We demonstrate the approach with an example of specification and verification.

Keywords: formal methods, real-time, integration, discretization, metric temporal logic, discrete time, continuous time, dense time.

1 Introduction and Motivation

The application of formal methods to the description and analysis of large and heterogenous systems inevitably requires being able to model different parts of the system using disparate notations and languages. In fact, it is usually the case that different modules are naturally described using diverse formal techniques, each one tailored to the specific nature of that component. Indeed, the past decades have seen the birth and proliferation of a plethora of different formal languages and techniques, each one usually focused on the description of a certain kind of systems and hinged on a specific approach. On the one hand, this proliferation is a good thing, as it allows the user to choose the notation and methodology that is best suited for her needs and that matches her intuition. However, this is also inevitably a hurdle to the true scalability in the application of formal techniques, since we end up having heterogeneous descriptions of large systems, where different modules, described using distinct notations, have no definite global semantics when put together. Therefore, we need to find ways to

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 215–229, 2006.

integrate dissimilar models into a global description which can then be analyzed, so that heterogeneity of notation.

A particularly relevant instance of the above general problem is encountered when describing real-time systems, which require a quantitative modeling of time. Commonly, such systems are composed of some parts representing physical environmental processes and some others being digital computing modules. The former ones have to model physical quantities that vary continuously over time, whereas the latter ones are digital components that are updated periodically at every (discrete) clock tick. Hence, a natural way to model the physical processes is by assuming a *continuous-time* model, and using a formalism with a compliant semantics, whereas digital components would be best described using a *discrete-time* model, and by adopting a formalism in accordance. Thus, the need to integrate continuous-time formalisms with discrete-time formalisms, which is the object of the present paper.

In particular, let us consider the framework of descriptive specifications based on (metric) temporal logic. Some temporal logic languages have semantics for both a continuous-time model and a discrete-time one: each formula of the language can be interpreted in one of the two classes of models. TRIO is an example of these logics [5], the one we are considering in this paper; MTL [14] is another well-known instance.[1] The discrete-time semantics and the continuous-time one, however, are unrelated in general, in that the same formula unpredictably changes its models when passing from one semantics to another. On the contrary, integration requires different formulas to describe parts of the *same* system, thus referring to unique underlying models.

To this end, we introduce the notion of *sampling invariance* of a specification formula. Informally, we say that a temporal logic formula is sampling invariant when its discrete-time models coincide with the samplings of all its continuous-time models (modulo some additional technical requirements). The *sampling* of a continuous-time model is a discrete-time model obtained by observing the continuous-time model at periodic instants of time. The justification for the notion of sampling invariance stems from how real systems are made. In fact, in a typical system the discrete-time part (e.g., a controller) is connected to the (probed) environment by a sampler, which communicates measurements of some physical quantities to the controller at some periodic time rate (see Figure 1). The discrete-time behaviors that the controller sees are samplings of the continuous-time behaviors that occur in the system under control. Our notion of sampling invariance captures abstractly this fact in relating a continuous-time formula to a discrete-time one, thus mirroring what happens in a real system.

Once we have a sampling invariant specification, we can integrate discrete-time and continuous-time parts, thus being able, among other things, to resort to verification in a discrete-time model, which often benefits from more automated approaches, while still being able to describe naturally physical processes in a continuous-time model.

[1] We note that all the results drawn in this paper about TRIO can be applied to MTL with little effort.