From MITL to Timed Automata*

Oded Maler[1], Dejan Nickovic[1], and Amir Pnueli[2,3]

[1] Verimag, 2 Av. de Vignate, 38610 Gières, France
{Dejan.Nickovic, Oded.Maler}@imag.fr
[2] Weizmann Institute of Science, Rehovot 76100, Israel
[3] New York University, 251 Mercer St. New York, NY 10012, USA
Amir.Pnueli@cs.nyu.edu

Abstract. We show how to transform formulae written in the real-time temporal logic MITL into timed automata that recognize their satisfying models. This compositional construction is much simpler than previously known and can be easily implemented.

Prediction is very difficult, especially about the future.
Niels Bohr

1 Introduction

Decision procedures for model-checking of temporal logic formulae [MP91, MP95] play a central role in algorithmic verification [CGP99]. Such procedures are based, in the linear-time context, on deriving from a formula φ an automaton-like device that accepts exactly sequences of states or events that satisfy it [VW86]. For discrete-time models, used for functional verification of software or synchronous hardware, the logical situation is rather mature. Logics like LTL (linear-time temporal logic) or CTL (computation-tree logic) are commonly accepted and incorporated into verification tools. LTL admits a variety of efficient algorithms for translating a formula into an equivalent automaton [GPVW95, SB00, GO01, KP05] and it even underlies the industrial standard PSL [KCV04, HFE04].

When considering *timed* models and specification formalisms whose semantics involves the time domain $\mathbb{R}_+$ rather than $\mathbb{N}$, the situation is somewhat less satisfactory [A04]. Many variants of real-time logics [Koy90, AH92a, Hen98, HR04] as well as timed regular expressions [ACM02] have been proposed but the correspondence between simply-defined logics and variants of timed automata (automata with auxiliary clock variables [AD94]) is not as simple and canonical as for the untimed case, partly, of course, due to the additional complexity of the timed model. Consequently, existing verification tools for timed automata rarely use temporal properties other than safety. One of the most popular dense-time extensions of LTL is the logic MITL introduced in [AFH96] as a restriction of the logic MTL [Koy90]. The principal modality of MITL is the *timed until* $\mathcal{U}_I$ where I is some non-singular interval. A

* This work was partially supported by the European Community project IST-2003-507219 PROSYD (Property-based System Design).

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 274–289, 2006.

formula $p\,\mathcal{U}_{[a,b]}\,q$ is satisfied by a model at any time instant t that admits q at some $t' \in [t+a, t+b]$, and where p holds continuously from t to t'. The decidability of MITL was established in [AFH96] and it was, together with MTL, subject to further investigations [AH92b, RSH98, HRS98, OW05]. However, the automaton construction in [AFH96] is very complicated (11 pages) and, to the best of our knowledge, has never been implemented. The only logic that has been integrated into a real-time model-checking tool was the timed version of CTL, TCTL [HNSY94], used in the tool KRONOS [Y97].[4]

In this paper we remedy this situation somewhat by presenting a simple construction of timed automata that accept exactly models of MITL formulae. The construction is based on two ideas, the first being the *modular construction* of property testers for untimed formulae [KP05] and the other is the observation, similar to the one already made in [AFH96], that the evolution over time of the satisfiability of a formula of the form $p\,\mathcal{U}_{[a,b]}\,q$ is of bounded variability, regardless of the variability of p and q.

The rest of the paper is organized as follows. In Section 2 we illustrate the modular construction of testers for (untimed) LTL formulae. The logic MITL is presented in Section 3 together with its semantic domain (Boolean signals) and timed automata. The main result, the construction of property testers for MITL, is presented in Section 4, followed by a brief discussion of the differences between the version of MITL that we use and the one presented in [AFH96].

2 Temporal Testers for LTL

In this section we discuss some of the problems associated with the construction of automata that accept models of LTL formulae, and describe the modular procedure of [KP05] which we later extend for MITL. We feel that, independently of its timed generalization, this construction, based on composition of *acausal sequential transducers*, improves our understanding of temporal logic and can serve as a unifying framework for both verification and monitoring. We assume familiarity with basic notions of LTL, whose syntax is defined according to the grammar

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathcal{U} \varphi_2$$

where p belongs to a set $P = \{p_1, \ldots, p_n\}$ of propositions. LTL is interpreted over n-dimensional Boolean ω-sequences of the form $\xi : \mathbb{N} \to \mathbb{B}^n$. We abuse p to denote the projection of the sequence ξ on p. The semantics of LTL formulae is typically given via a recursive definition of the relation $(\xi, t) \models \varphi$ indicating that the sequence ξ satisfies φ at position t. The satisfaction of a compound formula $op(\varphi_1, \varphi_2)$, where *op* is a temporal or propositional operator, by a sequence ξ is an *op*-dependent function of the satisfaction of the sub formulae φ_1 and φ_2 by ξ. Functionally speaking, the satisfaction of φ by arbitrary sequences can be viewed as a *characteristic function* χ^φ which maps sequences over $\mathbb{B}^n$ into Boolean sequences such that $\beta = \chi^\varphi(\xi)$

[4] An efficient emptiness checking algorithm for timed Büchi automata has been proposed and implemented in [TYB05] but without an upstream translation from a logical formalism.