Theoretical Computer Science 625 (2016) 85–124

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Timed recursive state machines: Expressiveness and complexity ☆

Massimo Benerecetti *, Adriano Peron

Department of Electrical Engineering and Information Technologies, Università di Napoli "Federico II", Italy

ARTICLE INFO

Article history:
Received 5 August 2014
Received in revised form 7 January 2016
Accepted 15 February 2016
Communicated by P. Aziz Abdulla

Keywords:
Formal languages
Formal models of computing
Timed automata
Real-time systems
Pushdown systems
Recursive state machines

ABSTRACT

The paper proposes a temporal extension of Recursive State Machines (RSMs), called Timed RSMs (TRSMs), which consists of an indexed collection of Timed Automata, called components. Each component can invoke other components in a potentially recursive manner. Besides being able to model procedure calls and recursion, TRSMs are equipped with the ability to suspend the evolution of time within a component when another component is invoked and to recover it when control is given back at return time. This mechanism is realized by storing clock valuations into an implicit stack at invocation time and restoring them upon return. Indeed, TRSMs can be related to an extension of Pushdown Timed Automata, called EPTAs, where an additional stack, coupled with the standard control stack, is used to store temporal valuations of clocks. The expressiveness and computational properties of the resulting model are analyzed, showing that it can be used to recognize timed languages exhibiting context-free properties not only in the untimed "control" part, but also in the associated temporal dimension. The reachability problem for both TRSMs and EPTAs is investigated, showing that the problem is undecidable in the general case. However, the problem becomes decidable for two meaningful subclasses, called I-TRSM and L-TRSM, obtained by suitably constraining the set of clocks to reset at invocation time and to restore at return time. The considered subclasses are compared with the corresponding EPTAs subclasses through bisimulation of their timed LTSs. The complexity of the reachability problem for L-TRSM and I-TRSM is proved to be **EXPTIME**-complete and **PSPACE**-complete, respectively. Moreover, we prove that such classes strictly enhance the expressive power of Timed Automata and of Pushdown Timed Automata, forming a proper hierarchy.

1. Introduction

The formalism of *Recursive State Machines* (RSMs) has been introduced in [5] to model control flow in typical sequential imperative programming languages with recursive procedure calls. A RSM is an indexed collection of finite state machines, called components. Components enhance the power of ordinary state machines with special nodes, called *boxes*, which correspond to invocations of other components in a potentially recursive manner. As shown in [5], RSMs are closely related to

☆ Partially supported by Italian MIUR Project "Integrating automated reasoning in model checking: towards push-button formal verification of large-scale and infinite-state systems", PRIN-20079E5KM8.

* Corresponding author.
E-mail addresses: massimo.benerecetti@unina.it (M. Benerecetti), adrperon@unina.it (A. Peron).

http://dx.doi.org/10.1016/j.tcs.2016.02.021
0304-3975/

Pushdown Systems (PDSs). While PDSs have been widely studied in the literature within the field of program verification, RSMs seem to be more appropriate for visual modeling. In the context of state-transition formalisms, Timed Automata [2], TAs for short, have received a lot of attention in the past twenty years and have become the reference framework for modeling real-time systems. Their good computational properties, deriving from the decidability of emptiness and the reachability problems [2], have also led to the development of automated tools for the analysis of real-time systems [23,18]. In this paper we consider a real-time extension of RSMs, called *Timed RSMs* (TRSMs for short), which allows to model real-time recursive systems. Roughly speaking, a TRSM is an indexed collection of Timed Automata, referred to as *timed components*, with the additional feature of providing boxes corresponding to invocations of other timed components. All the timed components can refer to a common set of clocks used to constrain their behavior. Within a timed component, the only explicit update of clocks is the standard operation of *clock reset*. In addition, clock updates occur when invocations of other components or returns from components are performed. In particular, at invocation time one can choose to reset clocks, and at return time some clocks may be restored to the values they had at invocation time.

The idea of extending formalisms, which implicitly allow to model recursive systems (e.g., PDSs), with real-time features has already been proposed in the literature (e.g., [10–12,16,15]). In particular, [10,15,16] introduce and study Pushdown Timed Automata (PTAs), namely Timed Automata augmented with a pushdown control stack. The work in [10,11] considers the reachability problem, namely reachability of a given control location, in PTAs (or, equivalently, Context-Free Timed Systems), reducing it to the reachability problem in standard PDSs. The work in [15,16], on the other hand, considers the more general problem of binary reachability in PTAs, namely reachability between configurations, showing that the problem is decidable as well.

Notice, however, that besides that fact that TRSMs, similarly to RSMs, are more appropriate than PTAs for visual modeling, PTAs are not expressive enough to account for storing and restoring clock values. Indeed, the control stack of PTAs can trace the history of component invocation, but cannot be exploited to record the history of clock values stored at invocation time and needed at time of the matching return for restoration. In particular, the clock store/restore mechanism allows to model in a very natural way a notion of time spent by the local computation of a timed component. In other words, in TRSMs one can predicate on the evolution of time within a single component, abstracting away the time elapsed within the invoked components. Since the number of recursive invocation can be unbounded, this notion of local time does not seem amenable to modeling by means of PTAs without employing an unbounded number of clocks.

The correspondence between RSMs and PDSs, as introduced in [5], can be lifted to the timed setting. Since, however, PTAs are not expressive enough to account for TRSMs, we propose Extended PTAs (EPTAs), which augment PTAs with an additional stack used to store clock valuations. The two stacks are independent, in the sense that the control stack is used to save control symbols and the valuation stack is used only for storing clock valuations. As a consequence, EPTAs, besides the standard clock reset operations, also allows for Store and Restore operations on clock valuations. We shall prove that TRSMs are equivalent (via timed bisimulation) to a syntactic restriction of EPTAs, where the operations on the two stacks are synchronized. EPTAs offer a suitable framework to study timed languages exhibiting context-free properties both in the untimed "control" part and in the associated "timestamps". For instance, we provide an example of a context-free timed language with mirror distribution of symbols and of temporal delays between consecutive symbols.

In the paper we identify and study meaningful subclasses of TRSMs and corresponding (via bisimulation) subclasses of EPTAs. The first subclass, called Local TRSM (L-TRSM), restricts TRSMs so as to allow for a local use of clocks. This is obtained by requiring that all the clocks get restored when returning from a component invocation. Therefore, the temporal information may flow from the calling environment to the invoked component but cannot flow in the other direction from the callee back to the caller. The smaller class, called Initialized TRSM (I-TRSM), completely isolates the temporal context of an invoked component, by further requiring that all the clocks are reset (initialized) at each invocation. In a sense, the temporal dimension of each called component of a I-TRSM proceeds independently from the temporal evolution of the calling environment, since the temporal information in I-TRSM neither flows from the calling environment to the invoked component nor from the callee back to the caller.

The expressiveness, decidability and complexity results are the main technical contributions of the paper. A complete comparison of the expressive power (as acceptors of timed languages) of these classes is presented. Decidability and complexity results for the reachability problem are provided for all the classes as well. In particular, we show that, from the expressiveness viewpoint TAs, I-TRSM, L-TRSM and TRSMs form a strictly increasing hierarchy. Unfortunately, we show that I-TRSM and L-TRSM are not closed under intersection, even when intersected with TAs. From the reachability viewpoint, we show that the problem is undecidable for the general classes of TRSMs and EPTAs. However, decidability can be recovered for their subclasses I-TRSM, L-TRSM (and, correspondingly, for I-EPTA and L-EPTA), for which the complexity of the problem is established. In particular, we prove that, despite the fact that the class I-TRSM (and I-EPTA) is more expressive than TAs, the complexity of the reachability problem remains unchanged, namely **PSPACE**-complete. The reachability problem for L-TRSM (and L-EPTA) turns out to be **EXPTIME**-complete.

Related work The relation with the basic traditional formalisms of Pushdown Systems, Recursive State Machines [5], Pushdown Timed Automata [15] has been already outlined in the initial part of the introduction. Even if a precise comparison goes beyond the goals of the paper, a qualitative relation with the formalism of Stopwatch Automata (SWAs for short) [14] is also outlined. The distinguishing feature of SWAs is the ability to freeze/unfreeze the temporal progress of clocks. The invocation/return mechanism introduced in TRSMs, together with the ability of freely choosing the set of clocks to restore upon