Timed pushdown automata revisited

Sławomir Lasota
University of Warsaw

***Abstract*—We prove two results on timed extensions of pushdown automata (PDA). As our first result we prove that the model of dense-timed PDA of Abdulla *et al.* collapses: it is expressively equivalent to dense-timed PDA with *timeless* stack. Motivated by this result, we advocate the framework of first-order definable PDA, a specialization of PDA in sets with atoms, as the right setting to define and investigate timed extensions of PDA. The general model obtained in this way is Turing complete. As our second result we prove NEXPTIME upper complexity bound for the non-emptiness problem for an expressive subclass. As a byproduct, we obtain a tight EXPTIME complexity bound for a more restrictive subclass of PDA with timeless stack, thus subsuming the complexity bound known for dense-timed PDA.**

This is a joint work with with Lorenzo Clemente from University of Warsaw.

I. Introduction

Background. Timed automata [1] are a popular model of time-dependent behavior. A timed automaton is a finite automaton extended with a finite number of variables, called clocks, that can be reset and tested for inequalities with integers; so equipped, a timed automaton can read timed words, whose letters are labeled with real (or rational) timestamps. The value of a clock implicitly increases with the elapse of time, which is modeled by monotonically increasing timestamps of input letters.

We investigate timed automata extended with a stack. An early model extending timed automata with an untimed stack, which we call *pushdown timed automata* (PDTA), has been considered by Bouajjani *et al.* [2]. Intuitively, PDTA recognize timed languages that can be obtained by extending an untimed context-free language with regular timing constraints. A more expressive model, called *recursive timed automata* (RTA), has been independently proposed (in an essentially equivalent form) by Trivedi and Wojtczak [3], and by Benerecetti *et al.* [4]. RTA use a timed stack to store the current clock valuation, which can be restored at the time of pop. This facility makes RTA able to recognize timed language with non-regular timing constraints (unlike PDTA).

More recently, *dense-timed pushdown automata* (dtPDA) have been proposed by Abdulla *et al.* [5] as yet another extension of PDTA. In dtPDA, a clock may be pushed on the stack, and its value increases with the elapse of time, exactly like the value of an ordinary clock. When popped from the stack, the value may be tested for inequalities with integers. The non-emptiness problem for dtPDA is solved in [5] by an ingenious reduction to non-emptiness of classical *untimed* PDA. As a byproduct, this shows that the untiming projection of dtPDA-language is context-free. Perhaps surprisingly, we prove the semantic collapse of dtPDA to PDTA, i.e., dtPDA with timeless stack but timed control locations: every dtPDA may be effectively transformed into a PDTA *that recognizes the same timed language*. Notice that this is much stronger than a mere reduction of the non-emptiness problem from the former to the latter model. Intuitively, the collapse is caused by the accidental interference of the LIFO stack discipline with the monotonicity of time, combined with the restrictions on stack operations assumed in dtPDA. Thus, dtPDA are equivalent to PDTA, and therefore included in RTA. The collapse motivates the quest for a more expressive framework for timed extensions of PDA.

Timed register pushdown automata. We advocate *sets with atoms* as the right setting for defining and investigating timed extensions of various classes of automata. This setting is parametrized by a logical structure $\mathbb{A}$, called *atoms*. Intuitively speaking, sets with atoms are very much like classical sets, but the notion of finiteness is relaxed to orbit-finiteness, i.e., finiteness up to an automorphism of atoms $\mathbb{A}$. The relaxation of finiteness allows to capture naturally various infinite-state models. For instance, ignoring some inessential details, register automata [6] (recognizing data languages) are expressively equivalent to the reinterpretation of the classical definition of 'finite NFA' as 'orbit-finite NFA' in sets with *equality atoms* $(\mathbb{N}, =)$ (see [7] for details), and analogously for register pushdown automata [8].

Along similar lines, timed automata (without stack) are essentially a subclass of NFA in sets with *timed atoms* $(\mathbb{Q}, \leq, +1)$, i.e., rationals with the natural order and the $+1$ function (see [9] for details). The automorphisms of timed atoms are thus monotonic bijections from $\mathbb{Q}$ to $\mathbb{Q}$ that preserve integer differences. In fact, to capture timed automata it is enough to work in a well-behaved subclass of sets with timed atoms, namely in *(first-order) definable sets*. Examples of definable sets are

$$A \ = \ \{(x, y, z) \in \mathbb{Q}^3 \ : \ x < y < z + 1 < x + 4\}$$
$$A' \ = \ \{(x, y) \in \mathbb{Q}^2 \ : \ x = y \ \vee \ y > x + 2\}.$$

The first one is orbit-finite, while the other is not.

By reinterpreting the classical definition of PDA in definable sets we obtain a powerful model, which we call *timed register PDA* (trPDA), where, roughly speaking, a clock (or even a tuple of clocks) may be pushed to, and popped from the stack, conditioned by arbitrary clock constraints referring possibly to other clocks. Notice that monotonicity is not part of the definition of timed atoms, and thus in general trPDA read non-

The last author acknowledges a partial support of the Polish National Science Centre grant 2012/07/B/ST6/01497.

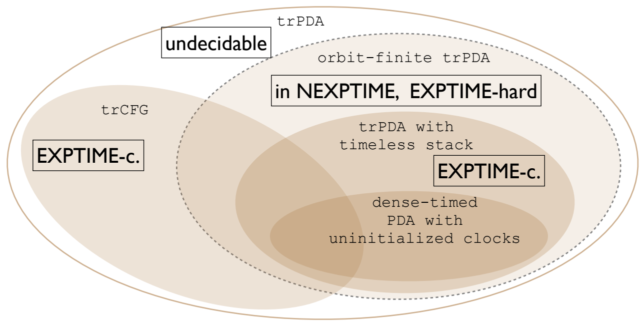

Fig. 1: Classes of timed pushdown languages.

monotonic timed words, unlike classical timed automata or dense-timed PDA. This is not a restriction, since monotonicity can be checked by the automaton itself, and thus we can model monotonic as well as non-monotonic timed languages. An example language recognized by a trPDA (or even by trCFG) is the language of palindromes over the alphabet A defined above. Another example is the language of bracket expressions over the alphabet $\{[,]\} \times \mathbb{Q}$, where the timestamps of every pair of matching brackets belong to A'. These languages intuitively require a timed stack in order to be recognized, and thus fall outside the class of dtPDA due to our collapse result.

Contributions. In view of possible applications to verification of time-dependent recursive programs, we focus on the computational complexity of the non-emptiness problem for trPDA. We isolate several interesting classes of trPDA, which are summarized in Fig. 1. All intersections are non-trivial. Our model subsume dtPDA, for the simple reason that the finite-state control is essentially a timed-register NFA, which subsumes timed automata, i.e., the finite-state control of dtPDA. For the general model we prove undecidability of non-emptiness. This motivates us to distinguish an expressive subclass, which we call *orbit-finite* trPDA, which is obtained from the general model by imposing a certain orbit-finiteness restriction on push and pop operations. We show that non-emptiness of orbit-finite trPDA is in NExpTime. This is shown by reduction to non-emptiness of the least solution of a system of equations over sets of integers (cf. [10] and references therein). This reduction is the technical core of the results. Moreover, it shows the essentially *quantitative* flavor of the dense time domain $(\mathbb{Q}, \leq, +1)$ as opposed to other kind of atoms, like equality $(\mathbb{N}, =)$ or total-order atoms $(\mathbb{Q}, \leq)$. Note that $(\mathbb{R}, \leq, +1)$ has the same first-order theory of the rationals, and thus considering the latter instead of the reals is with no loss of generality. Interestingly, our proofs work just as well over the discrete time domain $(\mathbb{Z}, \leq, +1)$.

In order to establish the claimed complexity upper bound, we establish, along the way, tight complexity results for solving systems of equations in special form. From this analysis, we derive ExpTime-completeness of the subclass of trPDA with timeless stack. Due to our collapse result, under a simple technical assumption that preserves non-emptiness, dtPDA can be effectively transformed into trPDA with timeless stack, and thus we subsume the ExpTime upper bound shown in [5].

Finally, we consider the reinterpretation of context-free grammars in sets with timed atoms. We prove that timed context-free languages are a strict subclass of trPDA languages, and that their non-emptiness is ExpTime-complete.

Except for the technical results, we offer a wider perspective on modeling timed systems. We claim that sets with atoms have a significant and still unexplored potential for capturing timed extensions of classical models of computation.

References

[1] R. Alur and D. L. Dill, “A theory of timed automata,” *Theor. Comput. Sci.*, vol. 126, pp. 183–235, April 1994.

[2] A. Bouajjani, R. Echahed, and R. Robbana, “On the automatic verification of systems with continuous variables and unbounded discrete data structures,” in *Hybrid Systems’94*, 1994, pp. 64–85.

[3] A. Trivedi and D. Wojtczak, “Recursive timed automata,” in *Proc. of ATVA’10*. Springer, 2010, pp. 306–324.

[4] M. Benerecetti, S. Minopoli, and A. Peron, “Analysis of timed recursive state machines,” in *Proc. of TIME’10*, sept. 2010, pp. 61–68.

[5] P. A. Abdulla, M. F. Atig, and J. Stenman, “Dense-timed pushdown automata,” in *Proc. of LICS’12*, june 2012, pp. 35–44.

[6] M. Kaminski and N. Francez, “Finite-memory automata,” *Theor. Comput. Sci.*, vol. 134, no. 2, pp. 329–363, 1994.

[7] M. Bojańczyk, B. Klin, and S. Lasota, “Automata theory in nominal sets,” *Logical Methods in Computer Science*, vol. 10, no. 3:4, 2013.

[8] E. Y. C. Cheng and M. Kaminski, “Context-free languages over infinite alphabets,” *Acta Inf.*, vol. 35, no. 3, pp. 245–267, 1998.

[9] M. Bojanczyk and S. Lasota, “A machine-independent characterization of timed languages,” in *In Proc. of ICALP’12*. Springer, 2012, pp. 92–103.

[10] A. Jeż and A. Okhotin, “Complexity of equations over sets of natural numbers,” *Theory Comput. Syst.*, vol. 48, no. 2, pp. 319–342, 2011.

[11] L. Clemente and S. Lasota, “Timed pushdown automata revisited,” University of Warsaw, Tech. Rep., 2015. [Online]. Available: http://arxiv.org/abs/1503.02422