MDP + TA = PTA: Probabilistic Timed Automata, Formalized (Short Paper)

Simon Wimmer[1] and Johannes Hölzl[2]

[1] TU München wimmers@in.tum.de
[2] VU Amsterdam jhl890@vu.nl

Abstract. We present a formalization of probabilistic timed automata (PTA) in which we try to follow the formula "MDP + TA = PTA" as far as possible: our work starts from existing formalizations of Markov decision processes (MDP) and timed automata (TA) and combines them modularly. We prove the fundamental result for probabilistic timed automata: the region construction that is known from timed automata carries over to the probabilistic setting. In particular, this allows us to prove that minimum and maximum reachability probabilities can be computed via a reduction to MDP model checking, including the case where one wants to disregard unrealizable behavior.

1 Introduction

Timed automata (TA) [1] are a widely used formalism for modeling nondeterministic real-time systems. Markov decision processes (MDPs) with discrete time are popular for modeling probabilistic systems with nondeterminism. Probabilistic timed automata (PTA) fuse the concepts of TA and MDPs and allow probabilistic modeling of nondeterministic real-time systems. PRISM [3] implements model checking functionality for MDPs and PTA and has successfully been applied to a number of case studies [6].

We have previously formalized MDPs [2] and TA [8] in Isabelle/HOL. This paper presents an Isabelle/HOL formalization of PTA, which follows the formula "MDP + TA = PTA" as far as possible by combining our existing formalizations modularly. We prove the fundamental result for PTA: the region construction that is known from TA carries over to the probabilistic setting. In particular, we prove that minimum and maximum reachability probabilities (with respect to possbile resolutions of nondeterminism) can be computed via a reduction to MDP model checking, including the case where one wants to disregard unrealizable behavior. This work is a necessary first step towards our long-term goal of certifying the computation results of PRISM's backward reachability algorithm [4] for reducing PTA to MDP model checking. The formalization can be found in the Archive of Formal Proofs [9].

2 Preliminaries

Markov chains A probability mass function (PMF, or discrete distribution) $\mu :: \sigma\, pmf$ is a function $\sigma \Rightarrow \mathbb{R}_{\geq 0}$ with countable support $\{x \mid \mu\, x \neq 0\}$ whose range sums to 1. Any PMF forms a monad, thus we have $(map_{pmf}\, f\, \mu)\, y = \mu\, \{x \mid f\ x = y\}$ and $(ret_{pmf}\, x)\, x = 1$. A *Markov chain (MC)* is represented by the transition system $K :: \sigma \Rightarrow \sigma\, pmf$ (its kernel, which is commonly represented by a transition matrix $\mathbb{R}^{|\sigma| \times |\sigma|}$), mapping each state to a distribution of next states. The trace space $T_K\, s$ is the probability measure with the property $T_K\, s\, (x_0 \cdots x_n) = K\, s\, x_0 * \cdots * K\, x_{n-1}\, x_n$ (where $(x_0 \cdots x_n)$ is the set of state traces starting with $x_0, \cdots, x_n$). A *probabilistic coupling* with respect to a relation R exists between two PMFs μ and μ' (written $rel_{pmf}\, R\, \mu\, \mu'$) if there exists a distribution ν on the product type, such that the support of ν is a subset of R and the marginal distributions of ν are $\mu = map_{pmf}\, \pi_1\, \nu$ and $\mu' = map_{pmf}\, \pi_2\, \nu$. Probabilistic couplings allow us to relate two Markov chains.

Markov decision processes MDPs are automata allowing probabilistic and nondeterministic choice. An MDP is represented by the transition system $K :: \sigma \Rightarrow \sigma\, pmf\, set$, where σ is the type of states, and the probability distributions over the next states of type $\sigma\, pmf$ are called *actions*. Each MDP gives rise to a set of MCs, each showing one possible behaviour. We introduce, coinductively, *configurations* $\sigma\, cfg$, where each $c :: \sigma\, cfg$ consists of a state σ, an action $\sigma\, pmf$, and a continuation $\sigma \Rightarrow \sigma\, cfg$. The configurations give rise to a Markov chain $K_c :: \sigma\, cfg \Rightarrow \sigma\, cfg\, pmf$, by mapping the continuations over the actions. Each $c :: \sigma\, cfg$ whose actions are closed under K and which is in state s induces a MC showing a possible behavior of the MDP K starting in s. To simplify the theory, we assume that $K\, x \neq \emptyset$. See [2] for details.

Timed Automata Compared to standard finite automata, TA introduce a notion of clocks. Clocks are indexed by natural numbers and do not have any structure. A *clock valuation* u is a function of type $\mathbb{N} \Rightarrow \mathbb{R}$. Locations and transitions are guarded by *clock constraints*, which have to be fulfilled to stay in a location or to take a transition. Clock constraints are conjunctions of constraints of the form $c \sim d$ for a clock c, an integer d, and $\sim \in \{<, \leq, =, \geq, >\}$. We write $u \vdash cc$ if the clock constraint cc holds for the clock valuation u. We define a timed automaton A as a pair $(\mathcal{T}, \mathcal{I})$ where $\mathcal{I}$ is an assignment of clock constraints to locations (also named invariants) and $\mathcal{T}$ is a set of transitions written as $A \vdash l \longrightarrow^{g,r} l'$ where l and l' are the start and successor location, g is the guard of the transition, and r is a set of clocks that will be reset to zero when the transition is taken. States of TA are pairs of a location and a clock valuation. The operational semantics define two kinds of steps:

Delay: $(l, u) \to (l, u \oplus d)$ if $d \geq 0$ and $u \oplus d \vdash \mathcal{I}\ l$;
Action: $(l, u) \to (l', [r \to 0]u)$ if $A \vdash l \longrightarrow^{g,r} l'$, $u \vdash g$, and $[r \to 0]u \vdash \mathcal{I}\ l'$;

where $(u \oplus d)\, c = u\ c + d$ and $([r \to 0]u)\, c = \mathit{if}\ c \in r\ \mathit{then}\ 0\ \mathit{else}\ u\ c$.

Regions The initial decidability result [1] partitioned the set of clock valuations into a quotient of sets of clock valuations, the so-called regions, and showed that these yield a sound and complete abstraction[3]. Our formalization [8] proves this fundamental result and decidability of reachability properties for ordinary TA.

3 Probabilistic Timed Automata

PTA fuse the concepts of TA and MDPs: discrete transitions are replaced by probability distributions over pairs of a set of clocks to be reset and a successor location. An example of a PTA is depicted in the left part of Fig. 1.

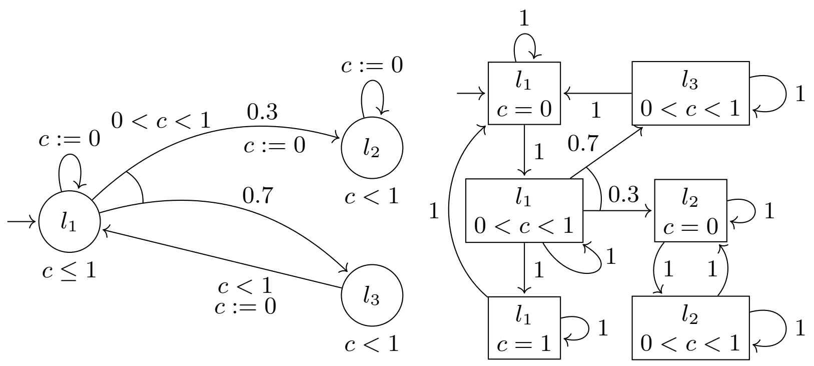

Fig. 1: Example of a PTA with one clock and its region graph

Definition Consequently, the syntactic definition of PTA is very similar to TA. The only difference is that now transitions are of the form $A \vdash l \longrightarrow^{g} \mu$ for μ of type $(\mathbb{N}\,\mathit{set} \times \sigma)\,\mathit{pmf}$ (the clocks to reset and σ for the type of locations).

Typical presentations define the semantics of PTA based on the notion of so-called *probabilistic timed structures*, which are just a special type of MDP. We omit this detour and directly formalize PTA in terms of MDPs. Consequently, to formalize the semantics of a PTA, we define its kernel K as the smallest set that is compatible with

$$\frac{(l,u) \in S \qquad t \geq 0 \qquad u \oplus t \vdash \mathcal{I}\ l}{\mathit{ret}_{\mathit{pmf}}\ (l, u \oplus t) \in K\ (l,u)}\ \textsc{Delay}$$

$$\frac{(l,u) \in S \qquad A \vdash l \longrightarrow^{g} \mu \qquad u \vdash g}{\mathit{map}_{\mathit{pmf}}\ (\lambda(r,l).\,(l, [r \to 0]u))\ \mu \in K\ (l,u)}\ \textsc{Action}$$

where S is the set of valid states. A state (l,u) is valid if l belongs to A and if $u \vdash \mathcal{I}\ l$. For technical reasons there is a third rule to add self loops for non-valid states. These do not change the semantics as they are not reachable from valid states. The MDP K is uncountably infinite as S is generally infinite.

[3] We use the same notions as in [8]. Soundness: for every abstract run, there is a concrete instantiation. Completeness: every concrete run can be abstracted.

Region Graph We want to reduce the computation of reachability probabilities for A to a computation on a finite MDP. Analogous to TA, this reduction can be obtained through the region quotient. More precisely, we will partition S into a finite set of states $\mathcal{S}$ of the form (l, R), for l a location of A, and R a region of A such that $\forall u \in R.\, u \vdash \mathcal{I}\ l$. With this notion, the finite MDP, coined *region graph* in [5], is defined through its kernel $\mathcal{K}$:

$$\frac{(l,R) \in \mathcal{S} \qquad R' \in \mathit{Succ}\ R \qquad \forall u \in R'.\, u \vdash \mathcal{I}\ l}{\mathit{ret}_{\mathit{pmf}}\ (l, R') \in \mathcal{K}\ (l,R)}\ \textsc{Delay}_{\mathrm{R}}$$

$$\frac{(l,R) \in \mathcal{S} \qquad A \vdash l \longrightarrow^{g} \mu \qquad \forall u \in R.\, u \vdash g}{\mathit{map}_{\mathit{pmf}}\ (\lambda(r,l).\, (l, \{[r \to 0]u \mid u \in R\})\ \mu \in \mathcal{K}\ (l,R)}\ \textsc{Action}_{\mathrm{R}}$$

Here $\mathit{Succ}\ R$ denotes the set of regions that can be reached from R by delaying for an arbitrary amount of time. Again, for technical reasons there is a third rule to add self loops for non-valid states. The *maximum* probability (under all valid initial configurations) to reach the state in the upper right of the region graph depicted in Fig. 1 is 0.7, while the *minimum* probability is 0.

4 Bisimulation

We relate the infinite MDP that defines the PTA with the finite region graph in a way that directly allows us to prove correctness of the reduction for maximum and minimum reachability probabilities in one go. Concretely, our agenda is to first define abstraction and representation functions that map between configurations of the infinite MDPs and the finite region graph, and vice versa. We then prove a more general bisimulation theorem on MDPs which states that the path measure assigned to related paths is the same for related configurations.

Representation and Abstraction We will use the overloaded notations α and rep to denote the abstraction and representation functions for states, actions, and configurations. The main difficulty of our formalization effort was to define these such that one obtains the desired properties. What are these properties? Chiefly, for a valid configuration c, the probability distributions of the successors of c and $\alpha\ c$ should expose a *probabilistic coupling* w.r.t. the relation $\lambda c\, a.\, \alpha\ c = a$. Moreover, the abstraction of a representative should yield the original object: $\alpha\,(\mathit{rep}\ x) = x$. Finally, validity should be preserved, i.e. $\alpha\,(l, u) \in \mathcal{S} \leftrightarrow (l, u) \in S$.

The elementary abstraction functions are easy to define: $\alpha\ (l, u) = (l, [u]_{\mathcal{R}})$ for $[u]_{\mathcal{R}}$ the unique region with $u \in [u]_{\mathcal{R}}$, and $\alpha\ t = \mathit{map}_{\mathit{pmf}}\ \alpha\, t$ for an action t. For a configuration c, $\alpha\ c$ is defined co-recursively in terms of c: the concrete configuration c is maintained as the internal state of $\alpha\ c$ and states and actions are simply mapped with α; the internal successor configuration however is determined by the continuation of c for the *unique* successor state s of c such that $\alpha\ s$ is the successor state of $\alpha\ c$. The definition of rep is more involved and omitted for brevity.

Bisimulation Theorem At the core of our argument lies the following bisimulation theorem on Markov chains:

$$T_K\ x\ A = T_L\ y\ B \ \text{ if } R\,x\,y \text{ and } \forall \omega\,\omega'.\ \mathit{rel}_{\mathit{stream}}\,R\,\omega\,\omega' \longrightarrow (\omega \in A \leftrightarrow \omega' \in B)$$
$$\text{and } \forall x\,y.\ R\,x\,y \longrightarrow \mathit{rel}_{\mathit{pmf}}\,R\,(K\ x)\,(L\ y)$$

where $T_{\{K,L\}}$ denotes the trace space induced by Markov chains K and L, respectively; x and y are states of K and L; $\mathit{rel}_{\mathit{stream}}\ R$ compares two traces pointwise by R; and A and B are sets of infinite traces of K and L. Finally, R has to be of the form $R\,s\,t = (s \in S \wedge f\,s = t)$ for some S and f.

For a configuration c with state s, we can instantiate this theorem by taking $K = K_c$, $L = \mathcal{K}_c$, $x = s$, $y = \alpha\,s$, $f = \alpha$, and S as the set of valid configurations of the PTA. The coupling property of K and L follows because

$$\mathcal{K}_c\,(\alpha\,c) = \mathit{map}_{\mathit{pmf}}\,\alpha\ (K_c\,c)\ .$$

For this instantiation, $\mathit{rel}_{\mathit{stream}}\,R\,x\,y$ essentially means that y is the pointwise abstraction of x. We consider reachability properties on state traces of the form $\varphi\,U\,\psi$ (where φ and ψ can be a mixture of predicates on location and clock), so

$$A = \{\omega \mid \varphi\,U\,\psi\,(\mathit{smap}\ \mathit{st}\,\omega)\} \text{ and } B = \{\omega \mid (\varphi \circ \mathit{rep})\,U\,(\psi \circ \mathit{rep})\,(\mathit{smap}\ \mathit{st}\,\omega)\}$$

where $\mathit{smap}\ \mathit{st}\,\omega$ maps traces of configurations to traces of MDP states. Consequently, the premise on A and B is easily satisfied if

$$\forall s\,t.\,\alpha\ s = \alpha\ t \longrightarrow \varphi\ s \leftrightarrow \varphi\ t \wedge \psi\ s \leftrightarrow \psi\ t\ ,$$

which matches exactly the property that is delivered by the region construction.

5 Taking Zenoness Into Account

So far, we have considered bisimulation properties between trace spaces of pairs of related configurations. Minimum and maximum reachability probabilities, however, are considered in relation to a set of configurations C. To compute these probabilities, one can consider the set of configurations C_α on the finite MDP such that $\forall c \in C.\,\alpha\,c \in C_\alpha$ and $\forall c \in C_\alpha.\,\mathit{rep}\,c \in C$. If C is the set of valid configurations, for instance, then C_α is easily proved to be the set of valid configurations of the region graph.

Often one wants to restrict C such that unrealizable behaviors are excluded: a configuration should not be able to keep time from passing beyond a fixed deadline. A configuration is *zeno* if it admits such behaviours. In the example, a zeno configuration could continuously take the loop transition on l_1 without letting any time pass. In [5] a *computable* description of C_α is given for the case that C is restricted to configurations that only yield non-zeno behaviors with probability 1.

The critical component of our proof for the correctness of C_α (in the sense outlined above) is that *rep* chooses the successor states always such that at least half of the amount of time that could possibly elapse does elapse.

Interestingly, the proof of $\forall c \in C_\alpha.\, rep\, c \in C$ was much harder to formalize than the other direction, although roles seem to flipped in the argument of [5]. For the harder direction, we illustrate the structure of our proof on the part that is concerned with the single region R_∞ in which all clocks have elapsed beyond the *maximal clock constant of the automaton* (the region $c > 1$ in the example): any run on the region graph that stays in R_∞ forever is classified as non-zeno.

Our argument establishes that for a transition $(l, R_\infty) \to (l', R_\infty)$ of the region graph, the representing transition $(l, u) \to (l', u')$ will incur a time delay of 0.5 if $u \neq u'$. An informal argument can get away by claming that the transition can always be chosen such that the latter condition is satisfied. Unfortunately, this is not immediately true for the semantics given above as the abstract transition could always be a reset transition and thus time would never be allowed to elapse. A possible remedy is to fuse delay and action transitions into a single step. We rather want to keep them separate and instead employ a probabilistic argument: assuming that a transition with $l = l'$ occurs infinitely often, with probability 1 a step with $u \neq u'$ has to occur infinitely often.

6 Conclusion

Discussion Our bisimulation argument neatly separates discrete, TA-related reasoning from probabilistic, MDP-related reasoning. In fact, most of the proofs take place on the discrete side, because none of the arguments to satisfy the bisimulation theorem are predominantly of probabilistic nature. As seen above, only the reasoning on zenoness needs to break with this style.

We found it crucial to carry out each argument on the right level of abstraction. There are three main levels to consider here (from low to high): Markov chains, MDPs and configuration traces, and states and state traces of the PTA. The theorem is usually stated on the highest level possible, and often we can move easily from a higher to a lower level by applying a number of rewrite rules. For the divergence argument, we even introduce another level of abstraction: since we are only concerned with time, the location part can be dropped, and thus we consider traces of clock valuations. The probabilistic argument described in the last section manifests a rare case where one needs to put in some upfront work on a lower level to hold the argument on the higher level together.

It is not yet clear to us whether it is necessary or advantageous to work with *rep*. In the current formalization it still plays an important rule by providing the diverging concrete witness for a diverging configuration of the region graph. The bisimulation argument in section 4, however, can be made relying only on α.

Lastly, our simple definition of the PTA semantics and of the region graph shows that derived concepts can be surprisingly easy to define—even compared to an informal definition—if the necessary foundations have already been laid.

Related Work We are not aware of any previous proof-assistant formalizations of PTA. There is, however, another formalization of TA and the region construction using PVS [11]. A formalization in Coq [7] is aimed at modeling a subclass of TA and proving properties of concrete automata.

Future Work We conjecture that many further results for PTA can be formalized by following the formula "MDP + TA = PTA" in the style that we outlined above. In particular, most of the more practical *zone* based (as opposed to region based) exploration methods for the reduction to a finite MDP should lie within the scope of this technique. The backward reachability algorithm of PRISM [4] is an instance. This also means that verified or certified model checkers for PTA can be devised from a modular combination of verified tools for MDPs and TA. Work in this direction already exists for the latter [10] but not the former formalism.

Acknowledgments We want to thank David Parker and Gethin Norman for clarifying our understanding of PTA model checking w.r.t. divergence. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 713999 – Matryoshka).

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Th. Comp. Sci. **126**
2. Hölzl, J.: Markov chains and Markov decision processes in Isabelle/HOL. JAR **59**(3)
3. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: CAV 11. LNCS, vol. 6806, pp. 585–591. Springer (2011)
4. Kwiatkowska, M., Norman, G., Sproston, J., Wang, F.: Symbolic model checking for probabilistic timed automata. Information and Computation **205**(7), 1027–1077 (2007)
5. Kwiatkowska, M.Z., Norman, G., Segala, R., Sproston, J.: Automatic verification of real-time systems with discrete probability distributions. Th. Comp. Sci. **282**(1)
6. Norman, G., Parker, D., Sproston, J.: Model checking for probabilistic timed automata. Formal Methods in System Design **43**(2), 164–190 (2013)
7. Paulin-Mohring, C.: Modelisation of timed automata in Coq. In: Proc. of STACS'01. pp. 298–315. LNCS 2215 (2001)
8. Wimmer, S.: Formalized timed automata. In: Blanchette, J.C., Merz, S. (eds.) ITP 2016, Proceedings. LNCS, vol. 9807, pp. 425–440. Springer (2016)
9. Wimmer, S., Hölzl, J.: Probabilistic timed automata. Archive of Formal Proofs (2018), `http://isa-afp.org/entries/Probabilistic_Timed_Automata.html`, Formal proof development
10. Wimmer, S., Lammich, P.: Verified model checking of timed automata. In: Beyer, D., Huisman, M. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 61–78. Springer International Publishing, Cham (2018)
11. Xu, Q., Miao, H.: Formal verification framework for safety of real-time system based on timed automata model in PVS. In: Proc. of IASTED'06. pp. 107–112 (2006)