TIME-ABSTRACTING BISIMULATION FOR MARKOVIAN TIMED AUTOMATA

MohammadSadegh Mohagheghi[1] and Hojjat Sharifi[2]

[1]Department of Computer Science, Vali-e-asr University of Rafsanjan, Rafsanjan, Iran

[2]Department of Computer Engineering, Vali-e-asr University of Rafsanjan, Rafsanjan, Iran

ABSTRACT

Markovian timed automata (MTA) has been proposed as an expressive formalism for specification of real-time properties in Markovian stochastic processes. In this paper, we define bisimulation relation for deterministic MTA. This definition provides a basis for developing effective algorithms for deciding bisimulation relation for such automata.

KEYWORDS

Formal methods, real-time systems, stochastic systems, bisimulation relation

1. INTRODUCTION

Timed automata have shown to be very useful formalism for modeling and analyzing of real-time systems [4]. Timed transition systems (usually with infinite state space) are used for defining the semantics of timed automata. While in timed automata for any state, there are some (deterministic or nondeterministic) transitions to other states according to some constraints, using the concept of probability leads to probabilistic timed automata [1]. In this extension of timed automata, probability distributions are associated with each transition. Markovian timed automata (MTA) [7,11] is an extension of probabilistic timed automata (PTA) in which each location has an exponentially distributed residence time. UPPAAL [8] and PRISM[9] are two well-known tools that implement verification algorithms for probabilistic timed automata.

State-space explosion is known as one of the major challenges in the verification of many models. One solution to combat this challenge is to apply the concept of bisimulation [5]. Informally, a bisimulation for an automaton is an equivalence relation over its states, such that states in the same equivalence class are essentially indistinguishable over the evolution of the system. PRISM supports both week and strong bisimulation for probabilistic systems. In addition to state-space reduction, bisimulation relations can also be used for other purposes such as stepwise system development (see [3] for example). According to the type of transition system, there are various definitions for bisimulation relations. For example, in [6] this relation was studied for probabilistic systems. That definition has been proposed for a system with finite state space while for many real-time models and semantics of timed automata, the state space is infinite. In general, the problem of computing bisimular states of infinite transition systems is not

DOI : 10.5121/ijsea.2016.7403

decidable and to cope with this problem, many modified versions of bisimulation relations have been proposed. For example, in [2] timed bisimulation and probabilistic timed bisimulation have been studied. In [3] the concept of week bisimulation for PTA has been introduced and a definition for sets of states, called classes, has been utilized in order to overcome the problem of infinite states. This method can be considered as a very promising one for deciding bisimulation relation for Markovian timed automata.

In this paper we specify the notion of zones and cylinder sets for MTA and accordingly, extend the definition of bisimulation relation to such type of formalism. The proposed definitions and concepts provide the required basics for developing deciding algorithms for MTA.

2. PRELIMINARIES

In this section we review some basic concepts and definitions which will be used in the subsequent discussions.

Distributions. A (discrete probability) distribution over a countable set S is defined as a function $\mu : S \to [0,1]$. The set of all distributions over set S is denoted by $Distr(S)$. For a given distribution $\mu : S \to [0,1]$ and a set $Q' \in S$, we define $\mu(Q') = \sum_{q \in Q'} \mu(q)$. For a set H we define $\Pr : F(H) \to [0,1]$ as a probability measure on the measurable space $(H, F(H))$ where $F(H)$ is a $\sigma-$algebra over H.

Definition 1. Markov Decision Process: A Markov decision process (MDP) [12] is a tuple $D = (Act, S, s_0, P)$ where Act is a finite set of actions; S is a set of states; $s_0 \in S$ is the initial state; and $P : S \times Act \times F(S) \to [0,1]$ is a transition probability function, where $P(.,.,A)$ is measurable for any $A \in F(S)$. In fact, $P(s, \alpha, A)$ shows the probability of one step transition from state $s \in S$ to the set of destination states $A \in F(S)$ using action $\alpha \in Act$.

Equivalences and Partitions. If R is an equivalence relation on a set S then S/R shows the set of equivalence classes and for every $s \in S$ we use the notion of $[s]_R$ for the equivalence class of s with respect to R. A partition of a set S is a set P consisting of pair wise disjoint nonempty subsets of S such that $\cup_{B \in p} B = S$. A partition P is finer than a partition P' (and P' is coarser than P) if and only if each $B \in P$ is contained in some $B' \in P'$.

Time and Clocks. Let $X = \{x_1, ..., x_n\}$ be a set of nonnegative real variables mentioned as clocks. The values of all clocks increase at the same rate as real-time.

A valuation over X (clock valuation) is a mapping $\eta : X \to R \geq 0$ assigning real values to clocks. We use $V(X)$ for all possible clock valuations over X. For a valuation η and a time value $t \in R \geq 0$ let $\eta + t$ denote the valuation such that $(\eta + t)(x) = \eta(x) + t$, for each clock $x \in X$. For a subset of clocks $\overline{X} \subseteq X$ the reset of $\overline{X}$, denoted by $\eta[\overline{X} := 0]$, is the valuation η' such that $\forall x \in \overline{X} : \eta'(x) = 0$ and $\forall x \notin \overline{X} : \eta'(x) = \eta(x)$. Through this paper we use $\overline{X}$ for subsets of X.

A clock constraint ϕ over X is defined as follows:

$$\phi ::= true \mid false \mid x \sim c \mid \phi_1 \wedge \phi_2$$

where $\sim \in \{<, \leq, \geq, >\}$, x is a clock variable and $c \in N$. We call ϕ a *clock-zone* if it has some relations of the form $x - y \sim c$ where $x, y \in X$. Notice that in this paper we use the word *zone* for a class of states with common location, i.e. a zone is a pair (l, ϕ) where $l \in Loc$ and ϕ is a clock-zone.

Definition 2. (Syntax of MTA) A Markovian timed automata (MTA) is a tuple $M = (Act, X, Loc, l_0, E, \rightarrow)$ where

- Act is a finite set of actions;
- X is a finite set of clocks;
- Loc is a finite set of locations;
- $l_0 \in Loc$ is the initial location;
- $E : Loc \rightarrow R > 0$ is the exit rate function;
- $\rightarrow \subseteq Loc \times Act \times B(X) \times Distr(2^X \times Loc)$.

For simplicity, we abbreviate $(l, \alpha, grd, \zeta) \in \rightarrow$ by $l \xrightarrow{\alpha, grd} \zeta$ where l is the source location, α is an action, grd is a clock constraint called a guard, and ζ is a probability distribution over $2^X \times Loc$ where X is the set of locations to be reset and Loc denotes destination location. Notice that this definition doesn't include location invariants.

Although in probabilistic timed automata we can have nondeterministic transitions [1] but in this paper we assume that the transitions are deterministic. In other words, for each location $l \in Loc$ and action $\alpha \in Act$ there exists at most one probability distribution and moreover, clock constraints does not contain neither $\vee$ nor $\neg$ operator.

Let $I(l,\eta) \in Act$ be the set of actions enabled in location $l \in Loc$ under clock valuation $\eta \in V(X)$. Although in the definition of MTA in [7] probabilistic distributions may not be total but in this paper we suppose they are total. It means that for each ζ in each $l \xrightarrow{\alpha, grd} \zeta$, we have $\sum_{Y \subset X, l \in LOC} \zeta(Y, l) = 1$.

Definition 3. For each $l \in Loc$ and $\alpha \in Act$, $guard(l, \alpha)$ is a clock constraint that is defined for the transition with l as source state and α as action. We define $P(l_1, \alpha_1, l_2) = p$ if $\exists grd \in guard(l, \alpha), \overline{X} \subset X : l \xrightarrow{\alpha, grd} \zeta, \zeta(\overline{X}, l_2) = p$. Like [7] we define the semantics of an MTA as a continues-state Markov decision process, where states are (like Timed automata) of the form (l, η) where $l \in Loc$ and η is a clock valuation over X. In fact, the execution of an MTA can be traced by a sequence of states (paths). One can use the concept of paths to calculate the probability of reaching some states.

Definition 4. (Path) A finite path in MTA M is of the form $l_0 \xrightarrow{\alpha_0,t_0} l_1 \xrightarrow{\alpha_1,t_1} ..l_n$ where for each transition $l_i \xrightarrow{\alpha_i,grd_i} \zeta_i$ of M with $\zeta_i(l_i,X_i)>0$, clock constraint η_i should be valid on entering location l_i that by definition $\eta_0=\vec{0}$ and $\eta_i + t_i \models g_i$ and $\eta_{i+1}=(\eta_i+t_i)[X_i=0]$. We denote with $Paths^M$ the set of finite paths in M. For $\rho \in Paths^M$ we use $\rho[n]=l_n$ for the *n*-th location of ρ.

Definition 5 (Semantics of MTA): let $M=(Act,X,Loc,l_0,E,\rightarrow)$ be an MTA. The (continuous state) MDP associated with M is $D(M)=(Act,S,s_0,P)$ where $S=\{(l,\eta)\mid l\in L,\eta\in V(X)\}$, $S_0=(l_0,\vec{0})$ and for each edge $l \xrightarrow{\alpha,grd} \zeta$ in M with $\zeta(\overline{X},l')=p>0$ and any $\eta \models grd$, we have :

$$P((l,\zeta),\alpha,A)=\int_{R\geq 0} E(l)e^{-E(l)\tau}.1_g(\eta+\tau).p\,d\tau \qquad (1)$$

where $A=\{(l',\eta')\mid \exists \tau\in R\geq 0.\eta'=(\eta+\tau)[\overline{X}=0] \text{ and } \eta+\tau\models grd\}$ and $1_g(.)$ is a boolean function where $1_g(\tau+\eta)=\begin{cases}1 & \text{if } \tau+\eta\models g,\\ 0 & \text{otherwise}\end{cases}$.

Note that the set A in Definition 5 is indeed a zone [8]. More precisely, for a state (l,η), time duration *t*, and set of clocks $\overline{X}\subseteq X$ a zone is defined as

$\text{Zone}((l,\eta),t,\overline{X})=\{(l,\eta')\mid \eta'(x)=0 \text{ if } x\in\overline{X},\ 0\leq\eta'(x)-\eta(x)\leq t \text{ if } x\notin\overline{X}\}$ which includes all states with location l and clock valuations that either set a clock variable x to 0 or add duration *t* to x. In fact, for a transition $l \xrightarrow{\alpha,grd} \zeta$ if the clock valuation is η (when transition occurs) and $\zeta(\overline{X},l')=p>0$, then for each clock variable $x\in\overline{X}$ we should have $\eta'(x)=0$ and for other clock variables $x'\notin\overline{X}$, $\eta'(x')=\eta(x')+t'$ where $0\leq t'\leq t$. We can also suppose that $\eta(t')\models grd$ for those t'. As a consequence, we have the following result:

For a transition $l \xrightarrow{\alpha,grd} \zeta$ in M with $\zeta(X,l')=p>0$ and $\eta\models grd$ we have

$$\text{P}((l,\eta),\alpha,\text{Zone}((l',\eta),t,X))=\int_{\tau=0}^{\tau=t}\text{E}(l)\text{e}^{-\text{E}(l)\tau}.1_g(\eta+\tau).\text{p}\,\text{d}\tau \qquad (2)$$

where *t* is the maximum value that $\eta+t\models grd$. We use MDP(M) for associated MDP of a given MTA M.

Definition 6 (Zone equivalency). Let R be an equivalence relation on the set of states of MDP(M). Suppose s_1 and s_2 are two states of MDP(M) such that s_1 R s_2. For each set of clock variables $\overline{X},\overline{Y}\subset X$ and each $t\in R_{\geq 0}$ we have $Zone(s_1,t,\overline{X})\,R\,Zone(s_2,t,\overline{Y})$ if and only if for each

state $s'_1 \in \text{Zone}(s_1, t, \overline{X})$ there exists a state $s'_2 \in \text{Zone}(s_2, t, \overline{Y})$ that $s_1' \, R \, s_2'$. Moreover, we can develop definition of partitions on zones. Like $[s]_R$ that shows the block of states equivalent to s we define:

$$[\text{Zone}(s_1, t, \overline{X})]_R = \{\text{Zone}(s_1', t', \overline{X'}) \mid \text{Zone}(s_1, t, \overline{X}) \, R \, \text{Zone}(s_1', t', \overline{X'})\} \quad (3)$$

In fact this definition shows the class of equivalent states related to some zones. Moreover, for an MTA M we show the set of all (accessible) zones by Zones(M). A zone is accessible if all of its states are accessible. On the other hand, the number of all accessible zones is in general infinite. In many cases, definition of bisimulation and simulation relations for Probabilistic Transition Systems are based on the probability of performing transition from a state to a class of states (related to equivalence relation) [3, 4, 7, 10]. We can generalize the above equation for classes of zones:

Definition 7. Let $C \in \text{Zones(M)}/_R$ be an equivalence class under relation R. Suppose that all $Z \in C$ are pair wise disjoint. For any $s = (l, \eta)$ we define $P((l, \eta), \alpha, C) = \sum_{z \in C} P((l, \eta), \alpha, z)$. In other words, probability of reaching from s to a class of zones C is equal to sum of probability of reaching from s to any members of C. The definition of semantic of MTA (like other probabilistic automata) is based on the probability of transition from one state to a class of states. But because transitions in MTA are continuous time, the probability of reaching to a destination state from some source states is defined based on cylinder sets. In the following we review the concept of cylinder set [8674].

Definition 8. (Cylinder set) Given an MTA M, we show a cylinder set by $C(l_0, \alpha_0, I_0, ..., \alpha_{n-1}, I_{n-1}, l_n)$ where $(l_0, ..., l_n) \in Loc^{n+1}$ and $I_i \subseteq R_{\geq 0}$. The cylinder set denotes a set of infinite paths ρ in M such that $\rho[i] = l_i$ and $\rho < i > \in I_i$. Let $\Pr_{\eta 0}^M(C)$ denote the probability of C (probability of paths that belong to C) such that the initial clock valuation in location l_0 is η_0. Formally $\Pr_{\eta 0}^M(C) := P_0^M(\eta_0)$, where $P_i^M(\eta_i)$ is defined as follows:

$$P_i^M(\eta_i) = \begin{cases} 1 \;\; if \;\; i = n \\ \int E(l_i).e^{-E(l_i)\tau}.1_{g_i}(\eta_i + \tau).p_i.P_{i+1}^M(\eta_i + 1)d\tau \;\; if \;\; 0 \leq i < n \end{cases}$$ where $\eta_{i+1} = (\eta_i + \tau)[X_i = 0]$. In fact $P_i^M(\eta_i)$ shows the probability of set of transitions, according to C that starts from l_i with clock valuation η_i to l_n.

With Cylinder set, we can extend definition (1) and replace a Cylinder set instead of the set A: $P^M(s, \alpha, I, C) = \Pr_{\eta 0}^M(C')$ where $s = (l, \eta_0)$ and $C' = (l, \alpha, I, C)$.

3. Bisimulation Relation for MTA

In many types of probabilistic automata a bisimulation is an equivalence relation R such that $s_1 R s_2$ if the probability of reaching any equivalence class for both s1 and s2 is equal. But it is not

the case for MTA (also not for PTA) because the number of classes is infinite and the probability of reaching a class of equivalence states is zero. (Remember the relation (1) in definition 3) To solve this problem we use the concept zones (or set of classes like [3]).

Because the definition of bisimulation is based on semantic of an automaton, we define it for MTA as follows:

Definition 8. (Bisimulation) Two states $s_1 = (l_1, \eta_1)$ and $s_2 = (l_2, \eta_2)$ are bisimular if

1- For each $C \in Zones(M)/_R$ and each $\alpha \in Act$: $P(s_1, \alpha, C) = P(s_2, \alpha, C)$,
2- For each d >= 0 the above condition holds for s1 + d and s2 + d i.e after any time step (and before a location transition) either any of two states reach bisimular states or a deadlock occurs for both.

If we have some locations as final locations we should add third condition to above:

3- Either $l_1 \in L_f$ and $l_2 \in L_f$ or $l_1 \notin L_f$ and $l_2 \notin L_f$.

We write $s_1 \approx_R s_2$ (or simply $s_1 \approx s_2$) if s_1 and s_2 are bisimular. Two locations l_1 and l_2 are bisimular (we denote $l_1 \approx l_2$) if two states $s_1 = (l_1, 0)$ and $s_2 = (l_2, 0)$ are bisimular. As a result we can define a partition on the set of locations of a MTA. In this case a partition for LOC is a set $\Pi = \{B_1, ..., B_k\}$ such that $B_i \neq \phi$ (for $0 < i \leq k$), $B_i \cap B_j = \phi$ ($for\ 0 < i, j \leq k\ and\ i \neq j$), $Loc = \cup_{0<i\leq k} B_i$ and for each l and l' such that $l \approx l'$ iff there exists a block $B_i \in \Pi$ that $l \in B_i$ and $l' \in B_i$.

We can also extend the definition of bisimulation to cylinder sets and show that if $s_1 \approx s_2$ then the probability of reaching any final location under particular sets of bisimilar cylinder sets will be equal. For two sequences as the form $\sigma = l_0, \alpha_0, I_0, l_1, \alpha_1, I_1, ..., l_{n-1}, \alpha_{n-1}, I_{n-1}, l_n$ and $\sigma' = l'_0, \alpha'_0, I'_0, l'_1, \alpha'_1, I'_1, ..., l'_{n-1}, \alpha'_{n-1}, I'_{n-1}, l'_n$ (where n > 0) two cylinder sets $C(\sigma)$ And $C'(\sigma')$ are bisimular if for each $0 \leq i \leq n$: $\alpha_i = \alpha'_i$ and $I_i = I'_i$ and $l_i \approx l'_i$ (that means l_i and l'_i belong to the same block.) . This means that for each path in $C(\sigma)$ there is a path in $C'(\sigma')$ that their actions and time intervals are stepwise the same and their locations are stepwise bisimular.

A set of all bisimular cylinder sets defines a block cylinder set. Formally a block cylinder set denoted by $Block\_C(B_0, \alpha_0, I_0, ..., \alpha_{n-1}, I_{n-1}, B_n)$ is defined as $\bigcup_{l_i \in B_i, 0 \leq i \leq n} C(l_0, \alpha_0, I_0, ..., \alpha_{n-1}, I_{n-1}, l_n)$ where each B_i is a Block of bisimulation relation. Notice that we can define the probability of a block cylinder set as the sum of probability of its cylinder sets: $Pr^M_{\eta 0}(Block\_C(B_0, \alpha_0, I_0, ..., \alpha_{n-1}, I_{n-1}, B_n)) = \sum_{l_i \in B_i} Pr^M_{\eta 0}(C(l_0, \alpha_0, I_0, ..., \alpha_{n-1}, I_{n-1}, l_n))$. So we can redefine bisimulation relation for states as the probabilities of block cylinder sets:

Theorem 1: two states $s_1 = (l_1, \eta_1)$ and $s_2 = (l_2, \eta_2)$ are bisimular if for any block cylinder set Block_C and any $\alpha \in Act$ and interval $I \subseteq R_{\geq 0}$ we have $P^M(s_1, \alpha, I, Block\_C) = P^M(s_2, \alpha, I, Block\_C)$

Proof: By induction on the length of Block_C.

Notice that according to Definition 8 one condition for bisimularity of two states $s_1 = (l_1, \eta_1)$ and $s_2 = (l_2, \eta_2)$ is that $E(l_1) = E(l_2)$. We use this condition in the first step of decision algorithm for bisimulation.

4. CONCLUSION AND FUTURE WORKS

In this paper we have defined the bisimulation relation for markovian timed automata and have shown that by this definition we can reduce the state space of a MTA. Also we review some definitions that are necessary for decision algorithm for bisimulation of MTA.

We have many future works to be done. First of all we want to propose an algorithm for deciding bisimulation relation for MTA. Also we can show some other applications of bisimulation for MTA. In addition to this one can define a logical characterization of timed-abstract bisimulation and study bisimulation from logical point of view.

REFERENCES

[1] Kwiatkowska, M., Norman, G., Segala, R., & Sproston, J. (2002). Automatic verification of real-time systems with discrete probability distributions.*Theoretical Computer Science*, *282*(1), 101-150.

[2] Sproston, J., & Troina, A. (2010). *Simulation and bisimulation for probabilistic timed automata* (pp. 213-227). Springer Berlin Heidelberg.

[3] Lanotte, R., Maggiolo-Schettini, A., & Troina, A. (2010). Weak bisimulation for probabilistic timed automata. *Theoretical Computer Science*, *411*(50), 4291-4322

[4] Alur, R., & Dill, D. L. (1994). A theory of timed automata. *Theoretical computer science*, *126*(2), 183-235.

[5] Baier, C., & Katoen, J. P. (2008). *Principles of model checking* (Vol. 26202649, pp. 19-82). Cambridge: MIT press

[6] Baier, C., Engelen, B., & Majster-Cederbaum, M. (2000). Deciding bisimilarity and similarity for probabilistic processes. *Journal of Computer and System Sciences*, *60*(1), 187-231.

[7] Chen, T., Han, T., Katoen, J. P., & Mereacre, A. (2010). *Computing maximum reachability probabilities in Markovian timed automata*. Technical report, RWTH Aachen.

[8] Tripakis, S., & Yovine, S. (2001). Analysis of timed systems using time-abstracting bisimulations. *Formal Methods in System Design*, *18*(1), 25-68.

[9] Larsen, K. G., Pettersson, P., & Yi, W. (1997). UPPAAL in a nutshell.*International Journal on Software Tools for Technology Transfer (STTT)*, *1*(1), 134-152

[10] Kwiatkowska, M., Norman, G., & Parker, D. (2002). PRISM: Probabilistic symbolic model checker. In *Computer performance evaluation: modelling techniques and tools* (pp. 200-204). Springer Berlin Heidelberg.

[11] Brázdil, T., Korenčiak, Ľ., Krčál, J., Novotný, P., & Řehák, V. (2015). Optimizing performance of continuous-time stochastic systems using timeout synthesis. In *Quantitative Evaluation of Systems* (pp. 141-159). Springer International Publishing

[12] Legay, A., Sedwards, S., & Traonouez, L. M. (2014). Scalable verification of Markov decision processes. In *Software Engineering and Formal Methods*(pp. 350-362). Springer International Publishing.