Journal of Theoretical and Applied Information Technology
28th February 2013. Vol. 48 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THREE-VALUED ABSTRACTION IN MODEL CKECKING PROBABIBLISTIC REAL-TIME TEMPORAL LOGIC OF KNOWLEDGE

ZHIFENG LIU, BO SUN, CONGHUA ZHOU

School of Computer Science and Telecommunication Engineering, Jiangsu University,

Zhenjiang, 212013, China

E-mail: liuzf@ujs.edu.cn

ABSTRACT

Model checking in probabilistic real-time temporal logic of knowledge PTACTLK confronts the same challenge as in traditional model checking, that is the state space explosion problem. Abstraction is one of the most effective methods to alleviate the state space explosion problem, under the traditional framework of two-valued abstraction, the abstract model obtained using abstraction techniques is only the upper approximation of the original model. In this paper, we introduce three-valued abstraction into model checking probabilistic real-time temporal logic of knowledge, define the abstract model of a probabilistic real time interpreted system and present the three-valued semantics of PTACTLK on the abstract model. We prove that the abstract model obtained using the abstraction techniques is not only the upper approximation of the original model but also its lower approximation. At last, a simple communication protocol is adopted to illustrate the effectiveness of our abstraction techniques.

Keywords: *Model Checking (MC), Probabilistic Real-time Temporal Logic of Knowledge (PRTLK), PTACTLK, State Space Explosion(SSE), Three-valued Abstraction(TA)*

1. INTRODUCTION

Model checking[1] is a very important automated verification technique for finite state systems, which has been used in the verification of hardware checking, communication protocols and control systems and has attracted wide attention. In model checking, a multi-agent system S is modelled as a suitable model M_S , and the specification P to be verified is represented as a logical formula ϕ_P , thus the problem of checking whether a specification P is satisfied by a multi-agent system M is converted into whether the model checking problem $M_S \models \phi_P$ is valid. Reasoning about knowledge[2] has always been a core in artificial intelligence, thus many specification forms based on modal logic have been proposed and refined in the recent years, and the most popular one is the temporal logic of knowledge, which is a specification language used for modelling and reasoning for multi-agent systems. However, the verification of temporal logic of knowledge using model checking remains lack for several essential functionalities, and one of them is real-time. In paper[3], A. Lomuscio etc have considered the real-time aspect and proposed a logic to reason about real-time and knowledge in multi-agent systems, that is real-time temporal logic of knowledge TACTLK. By introducing the probabilistic factor into TACTLK we can obtain probabilistic real-time temporal logic of knowledge ---- PTACTLK.

Model checking in probabilistic real-time temporal logic of knowledge PTACTLK confronts the state space explosion problem as in traditional model checking, that is the state space grows exponentially with the increase of the number of concurrent components. As the states satisfying the verified property are searched exhaustively by the model checking algorithm, a too large or infinite state space will severely affect the efficiency of model checking. To alleviate the state space explosion problem, many techniques have been proposed by researchers, such as statute of partial order[4], symmetric statute[5], symbolic computation based on OBDD and abstraction [6, 7] etc. Abstraction is one of the most effective methods to alleviate the state space explosion problem, which uses an abstraction function to divide the state space of the original model equivalently and as a result a corresponding abstract model is obtained, in this way, the information of the original model that is irrelevant to the property to be verified is ignored and the model checking procedure is carried out in the obtained abstract model. As the state space of the abstract model is comparably smaller, the efficiency of verification by model checking is improved significantly.

Under the traditional framework of two-valued abstraction, the abstract model deduced from a original model is only the upper approximation of the original model. That is, if a property is satisfied by the abstract model, then it is also satisfied by the original model. However, if a property is not satisfied by the abstract model, it cannot be inferred that it is not satisfied in the original model either. Three-valued abstraction techniques[8] can alleviate this problem efficiently, which introduces the third value that represents uncertainty besides true and false, and in this way, the abstract model obtained using three-valued abstraction techniques is not only the upper approximation of the original model, but also its lower approximation. The study of three-valued abstraction techniques contains two aspects: the construction of abstract model and abstraction refinement.

In model checking temporal logic, abstraction and abstraction refinement techniques have been studied[7,9], and in real time systems, the construction of abstract model has also been studied[10]. However, as far as we know, none of these techniques has ever been studied in model checking probabilistic real-time temporal logic of knowledge. Therefore, the three-valued abstraction techniques in model checking probabilistic real-time temporal logic of knowledge have been systematically studied in this paper. Our work is as following: (1). For the real time part of probabilistic real-time temporal logic of knowledge PTACTLK, that is PTACTL, the abstract discrete clock valuations[10] is applied to implicitly construct the clock regions of the state space of a probabilistic real time interpreted system, and in this way we can obtain a finite form of the state space of the probabilistic real time interpreted system. For the epistemic operator *K* in PTACTLK, the definition of epistemic equivalent to an agent between two abstract states is given by us, therefore, the abstract states satisfying the constraints of this definition can be combined into one equivalent class and the state space of the probabilistic real time interpreted system can be further simplified. (2). Using the abstraction techniques presented above, the corresponding abstract model M^A can be deduced from the original model M of a probabilistic real time interpreted system, the three-valued semantics of PTACTLK on the abstract model is given, and we also prove that the abstract model obtained using the abstraction techniques is not only the upper approximation of the original model but also its lower approximation. (3). Finally, a simple communication protocol is adopted to illustrate the effectiveness of our abstraction techniques.

2. THE SYNTAX AND SEMANTICS OF PROBABILISTIC REAL-TIME TEMPORAL LOGIC OF KNOWLEDGE PTACTLK

Probabilistic real-time temporal logic of knowledge PTACTLK is obtained by introducing the probabilistic factor into real-time temporal logic of knowledge TACTLK, which is used for modelling and reasoning for multi-agent systems. TACTLK is the fusion of TCTL representing branching real time[11] and the modal logic[12] S5n representing knowledge operators.

Definition 1. (Syntax of probabilistic real-time temporal logic of knowledge PTACTLK)

Suppose *PV* is a set of propositional variables containing the symbol T representing the constant true, *Ag* is a set of *m* agents, and *I* is a time interval in $\Re$ with integer bounds. Let $p \in PV$, $i \in Ag$, and $\Gamma \subseteq Ag$,then the set of PTACTLK formulae is defined as following:

$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid A(\varphi U_I^{\rhd p} \varphi) \mid A(\varphi R_I^{\rhd p} \varphi) \mid$$
$$K_i^{\rhd p} \varphi \mid D_\Gamma^{\rhd p} \varphi \mid E_\Gamma^{\rhd p} \varphi \mid C_\Gamma^{\rhd p} \varphi, where\ \rhd \in \{<, \leq, >, \geq\}.$$

The formula $A(\varphi U_I^{\rhd p} \psi)$ represents that such a phenomenon holds with probability $\rhd p$: for each computation path we have φ holds until, in the interval I , ψ holds; $A(\varphi R_I^{\rhd p} \psi)$ represents that such a phenomenon holds with probability $\rhd p$: for each computation path we have either ψ holds until, in the interval I , both φ and ψ hold, or ψ always holds in the interval I ; $K_i^{\rhd p} \varphi$ denotes that with probability $\rhd p$, agent i considers φ as possible.

The other basic temporal modalities are introduced as usual: $AG_I^{\rhd p} \varphi = A(\perp R_I^{\rhd p} \varphi)$, $AF_I^{\rhd p} \varphi = A(\mathrm{T} U_I^{\rhd p} \varphi)$, $\perp = \neg \mathrm{T}$, $\alpha \to \beta = \neg \alpha \vee \beta$ and $\alpha \leftrightarrow \beta = (\alpha \to \beta) \wedge (\beta \to \alpha)$.

In order to illustrate the semantics of probabilistic real-time temporal logic of knowledge, let us first review the knowledge about clock constraint[13], discrete probability distribution and probabilistic timed automata[14].

Definition 2. (Clock constraint) A clock constraint g over the set of clock variables C is formed according to the following grammar: $g := x < c \mid x \le c \mid x > c \mid x \ge c \mid g \wedge g$. Where $c \in$ □ and $x \in C$ is a clock variable. We use $CC(C)$ to denote the set of all the clock constraints over the set of clock variables C.

Definition 3. (Discrete probability distribution) The discrete probability distribution over a countable set Q is a function: $\mu : Q \to [0,1]$, such that $\sum_{q \in Q} \mu(q) = 1$. For an uncountable set Q', we use $Dist(Q')$ to represent the set of probability distributions over its countable subsets.

Timed automata[15,16] are used for modelling the behaviours of time-critical systems, probabilistic timed automata are an extension of timed automata with the ability to express relative likelihoods of state transitions, and the definition of probabilistic timed automata is in the following.

Definition 4. (Probabilistic timed automata) A probabilistic timed automaton can be represented as a tuple: $PTA = (L, Act, C, l_0, prob, Inv)$ .where L is a finite set of locations, $l_0 \in L$ is the initial location, Act is a finite set of actions, C represents a finite set of clock variables, $prob$ is a probabilistic edge relation: $prob \subseteq L \times CC(C) \times Act \times Dist(2^C, L)$ and Inv is a location invariant function $Inv : L \to CC(C)$, it assigns to each location a clock constraint which defines the condition that must be satisfied for the probabilistic timed automaton to stay in this location.

A discrete transition can be made according to $(l, g, \alpha, p) \in prob$ which is enabled, that is the clock constraint g is satisfied by the current clock valuation v, then the probability of moving to location l' from l and resetting all the clocks in D to 0 is $p(D, l')$.

As a multi-agent system(MAS) is composed by n agents($n > 0$,and n is an integer), if each agent $i(1 \le i \le n)$ is modelled as a probabilistic timed automaton $PTA_i = (L_i, Act_i, C_i, l_0^i, prob_i, Inv_i)$, then the multi-agent system can be modelled as the parallel composition of these n probabilistic timed automata. The definition of the parallel composition of many probabilistic timed automata is as the following.

Definition 5. (Parallel composition of probabilistic timed automata) The parallel composition of n probabilistic timed automata $PTA_i (1 \le i \le n)$ is a global probabilistic timed automaton $PTA = (L, Act, C, l_0, prob, Inv)$, where $L = \prod_{i=1}^{n} L_i$, that is a global location l of probabilistic timed automaton PTA is a tuple $l = (l_1, l_2, ..., l_n)$, $l_i \in L_i (1 \le i \le n)$, $Act = \bigcup_{i=1}^{n} Act_i$, $C = \bigcup_{i=1}^{n} C_i$, $l_0 = (l_0^1, l_0^2, ..., l_0^n)$, $l_0^i (1 \le i \le n)$ is the initial location of the ith probabilistic timed automaton PTA_i ; $Inv(l_1, l_2, ..., l_n) = \bigwedge_{i=1}^{n} Inv(l_i)$; if the transition $((l_1, l_2, ..., l_n), g, \alpha, p) \in prob$ is enabled, then the probability of moving to global location $(l_1^{'}, l_2^{'}, ..., l_n^{'})$ from $(l_1, l_2, ..., l_n)$ and resetting all the clocks in $D = \sum_{i \in Act(\alpha)} D_i$ to 0 is given by p.

The symbol $Act(a)$ above represents the set of indices of the probabilistic timed automata whose sets of actions Act_i ($1 \le i \le n$) contain the action a, that is $Act(a) = \{1 \le i \le n \mid a \in Act_i\}$. And D ($D \subseteq C$) is the set of clock variables that are reset to 0 during the global transition.

As the semantic model of TACTLK is real time interpreted systems, the semantic model of PTACTLK is probabilistic real time interpreted systems , which is defined as following.

Definition 6. (The semantic model of PTACTLK) The probabilistic real time interpreted system corresponding to the probabilistic timed automaton $PTA = (L, Act, C, l_0, prob, Inv)$ is a tuple $M = (Q, q_0, P, \sim_1, ..., \sim_n, P_1, ..., P_n, V)$, where:

- Q is a finite set of states, which is the subset of $L \times R^{|C|}$: $Q \subseteq L \times R^{|C|}$, and $R^{|C|}$ is the set of clock valuations over the set of clock variables C ,then each state in Q is a tuple (l, v) composed by a location l and a clock valuation v. All states in Q are reachable.
- $q_0 = (\ell_0, v_0)$ is the initial state such that $\forall x \in C$, $v_0(x) = 0$.
- P is the probabilistic function of state transitions, $P : Q \times Q \to [0,1]$, which says the probability of moving to another state from one state is between 0 and 1, and we have $\forall q \in Q, \sum_{q' \in Q,} P(q, q') = 1$ The transition relation between two state sets can be denoted as: $E \subseteq (L \times R^{|C|}) \times (Act \bigcup R^{+}) \times (L \times R^{|C|})$, there exists two kinds of transitions:

(1) Time transition: for $\delta \in R^{+}, (\ell, v) \xrightarrow{\delta} (\ell, v + \delta)$, if and only if $v \models Inv(\ell), v + \delta \models Inv(\ell)$;

(2) Action transition: for

$a \in Act, (\ell, v) \xrightarrow{a} (\ell', v')$, if and only if $(\exists cc \in CC(C))(\exists D \subseteq C)$ such that $\ell \xrightarrow{cc,a,D} \ell' \in E$ and $v \models cc$, $v' = v[D := 0] \models Inv(\ell')$. where $v' = v[D := 0]$ represents that clock valuation v' is obtained in this way: $\forall x \in D, v'(x) = 0$; $\forall x \in C \setminus D, v'(x) = v(x)$, the symbol D represents the set of clock variables that are reset to 0 in this transition.

- $\sim_i \subseteq Q \times Q$ $(1 \le i \le n, n$ is the number of agents) is an epistemic equivalent relation : $(\ell, v) \sim_i (\ell', v')$ if and only if for agent i $(1 \le i \le n)$, we have $\ell_i(\ell) = \ell_i(\ell')$ and $v \cong v'$. $\ell_i(\ell)$ represents the location component of agent i in the global location ℓ , and $v \cong v'$ denotes that clock valuations v and v' are equivalent. In fact, $\sim_i$ is an epistemic accessibility relation.
- $P_i : Q \times Q \to [0,1]$ $(1 \le i \le n)$ is the probability function over epistemic relations, such that $\forall q \in Q, \sum_{q' \in Q,} P_i(q, q') = 1$.
- $V : Q \to 2^{PV}$ is a valuation function, and we have $V((\ell, v)) = V_{PTA}(\ell)$, $V_{PTA}(\ell)$ represents the set of propositional variables that are valid in the location ℓ of probabilistic timed automaton PTA .

Definition 7. (Semantics of PTACTLK: the satisfaction relations)

Let $M = (Q, q_0, P, \sim_1, ..., \sim_n, P_1, ..., P_n, V)$ be a probabilistic real time interpreted system, $M, q \models \alpha$ represents that the PTACTLK formula α is true at state q in M , and in the following satisfaction relations, the symbol M is omitted. Suppose p , ϕ and φ in the following are all PTACTLK formulae, the satisfaction relation "|=" is defined inductively in the following:

- $q \models p$ iff $p \in V(q)$, where p is an atomic proposition;
- $q \models \neg p$ iff $p \notin V(q)$, where p is an atomic proposition;
- $q \models \phi \vee \varphi$ iff $q \models \phi$ or $q \models \varphi$;
- $q \models \phi \wedge \varphi$ iff $q \models \phi$ and $q \models \varphi$;
- $q \models A(\phi U_I^{\rhd p} \varphi)$ iff $\forall \rho \in f_{PTA}(q), \rho \models \phi \mathrm{U}_I \varphi$ and $\sum_{\rho \in f_{PTA}(q)} \Pr ob(\rho) \rhd p$, where $\rhd \in \{<, \le, >, \ge\}$.
- $q \models A(\phi R_I^{\rhd p} \varphi)$ iff $\forall \rho \in f_{PTA}(q), \rho \models \phi R_I \varphi$ and $\sum_{\rho \in f_{PTA}(q)} \Pr ob(\rho) \rhd p$;
- $q \models K_i^{\rhd p} \varphi$ iff $\sum_{q' \models \varphi} P_i(q, q') \rhd p$;
- $q \models E_\Gamma^{\rhd p} \varphi$ iff $\sum_{q' \models \varphi} \sum_{i \in \Gamma} P_i(q, q') \rhd p$;
- $q \models D_\Gamma^{\rhd p} \varphi$ iff $\sum_{q' \models \varphi} \prod_{i \in \Gamma} P_i(q, q') \rhd p$.

In the above, $\forall \rho \in f_{PTA}(q)$ denotes any path starting from the state q ; $\rho \models \phi U_I \varphi$ if and only if $(\exists r \in I)$ $(\pi_\rho(r) \models \varphi$ $and (\forall r' < r)(\pi_\rho(r') \models \phi))$, that is, for a path ρ starting from the state q , there exists a time point r in the time interval I such that on path ρ the state corresponding to time r satisfies the formula φ , and all the previous states on ρ satisfy the formula ϕ . $\rho \models \phi R_I \varphi$ if and only if $(\forall r \in I)$ $(\pi_\rho(r) \models \varphi$ or $(\exists r' < r)(\pi_\rho(r') \models \phi))$. $C_\Gamma^{\rhd p} \varphi$ is the transitive closure of $E_\Gamma^{\rhd p} \varphi$, and its satisfaction relation is ignored.

3. ABSTRACTION

3.1 The Abstraction Techniques

Every state in the set of states Q of a probabilistic real time interpreted system can be represented in the form: (ℓ, v) , where ℓ is a global location, and v is a clock valuation for all the clock variables. Due to the continuous nature of time, we have $v \in R^+$, and therefore the state space of probabilistic real time interpreted systems is infinite. As a result, the existing model checking algorithms for finite state systems can not be applied to model checking probabilistic real time interpreted systems. The state space of probabilistic real time interpreted systems needs to be divided equivalently such that the problem of model checking for probabilistic real time interpreted systems can be converted into model checking for finite state systems.

Suppose $M = (Q, q_0, P, \sim_1, ..., \sim_n, P_1, ..., P_n, V)$ is the original model of a probabilistic real time interpreted system, our aim is to present an abstraction technique, and using it a corresponding abstract model $M^A = (Q', q_0', P^{l'}, P^{u'}, \sim_1', ..., \sim_n', P_1^{l'}, P_1^{u'}, ..., P_n^{l'}, P_n^{u'}, V', ?)$ can be deduced from the original model M . Therefore , the model checking procedure can be carried out in the obtained abstract model while preserving the properties of the original model.

For every clock variable $x \in C$ in the probabilistic timed automaton that models a multi-agent system, we use Ix to represent its integer part variable and Fx its fractional order variable. *Ix* represents the integer part of clock variable x , that is, if $x \le c_x$,then $I_x = \lfloor x \rfloor$;otherwise $I_x = c_x$ (c_x is the maximum constant value compared to clock variable x). Therefore, Ix is an integer ranging between 0 and c_x. For a clock valuation v , order the clock variables that are smaller than or equal to the corresponding maximal constant value c_x according to the values of their fractional parts $fract(x)$, and we use Fx to represent the position of clock variable x in this order. For a clock variable x satisfying $x \le c_x$, the fractional order variable Fx = 0 if and only if the fractional part of x is zero, that is $fract(x)$=0. Therefore, Fx is an integer ranging between 0 to n (n is the number of clock variables).

Definition 8. (Discrete clock valuations) For the set of clock variables C of the probabilistic timed automaton modelling a multi-agent system, the corresponding discrete clock valuation v^d is a function: for every clock variable $x \in C$, assigns to Ix a value from $\{0,\ldots,c_x\}$ and to Fx a value from $\{0,\ldots,n\}$, and we use $v^d(x)$ to denote the pair $(v^d(Ix), v^d(Fx))$.

Definition 9. (Abstract discrete clock valuations) Given a set of clock variables C, one abstract discrete clock valuation over C is a function v^a : for each clock variable $x \in C$, it assigns a value from $\{0,...,c_x\}$ to Ix, and a value from F^a to F_x^a .

Definition 10. (Epistemic equivalent to agent i between two concrete states: $\sim_i$) For two concrete states (l,v) , (l',v') in a probabilistic real time interpreted system, the epistemic equivalent to agent $i (1 \le i \le n)$ between them can be represented as following: $(l,v) \sim_i (l',v')$ if and only if $l_i(l) = l_i(l')$ and $v \cong v'$. That is, the location component of agent i in the global location l is the same as that in the global location l', and the two clock valuations v and v' are equivalent.

Definition 11. (Epistemic equivalent to agent i between two abstract states: $\sim_i^{'}$)

Suppose (l,v) , (l',v') are two abstract states, the epistemic equivalent to agent i between them is defined as following: $(l,v) \sim_i^{'} (l',v')$ if and only if for each concrete state s_1 in (l,v) , and for each concrete state s_2 in (l',v') , it always holds that $s_1 \sim_i s_2$. That is, the two concrete states are epistemic equivalent to agent i .

Having the definition of epistemic equivalent to agent i $(1 \le i \le n)$ between two abstract states, the abstract states satisfying the constraints in the definition can be combined into one abstract state, that is a equivalent class. And therefore the state space of probabilistic real time interpreted systems can be further simplified.

3.2 Constructing the Abstract Model

The abstract model M^A corresponding to the original model M can be deduced according to the abstraction techniques presented above, that is the abstract discrete clock valuations and the epistemic equivalent to an agent between two abstract states.

Definition 12. (The abstract model of a probabilistic real time interpreted system)

The abstract model M^A corresponding to the original model M of a probabilistic real time interpreted system is a tuple $M^A = (Q', q_0^{'}, P^{l'}, P^{u'}, \sim_1^{'},..,\sim_n^{'}, P_1^{l'}, P_1^{u'},...., P_n^{l'}, P_n^{u'}, V', ?)$, where

- $Q^{'} = L^{'} \times R_C^A$ is the set of states of the abstract model, $L^{'}$ represents the set of abstract global locations and R_C^A represents the set of abstract discrete clock valuations over the set of clock variables C .
- $q_0^{'} = (l_0^{'}, v_0^{'})$ is the initial abstract state, and to one of its concrete initial state $q_0 = (l_0, v_0)$ it holds that $l_0 \in L_0$, and $\forall x \in C$, $v_0(x) = 0$.
- $P^{l'}, P^{u'} : Q' \times Q' \to [0,1]$ are matrices describing the lower and upper bounds for transition probabilities between abstract states such that for any two abstract states $\forall q_1, q_2 \in Q'$, we have $P^{l'}(q_1, Q') \le 1$, $P^{u'}(q_1, Q') \le 1$, $P^{l'}(q_1, Q') \le P^{u'}(q_1, Q')$ $P^{l'}(q_1, q_2) \le P^{u'}(q_1, q_2)$, where $P^{l'}(q_1, Q') = \sum_{q \in Q'} P^{l'}(q_1, q)$, $P^{u'}(q_1, Q') = \sum_{q \in Q'} P^{u'}(q_1, q)$. The state transition relation in the abstract model can be represented as: $E^{'} \subseteq (L^{'} \times R_C^A) \times (Act \bigcup R^+) \times (L^{'} \times R_C^A)$, and two kinds of transitions are available:

(1).Time transition: $(\ell, v^a) \xrightarrow{d} (\ell, v^a + d)$, $\forall d > 0$, if and only if $v^a \models Inv(\ell)$ and $v^a + d \models Inv(\ell)$ $(\forall d > 0)$;

(2).Action transition: $(\ell, v^a) \xrightarrow{a} (\ell^{'}, v_a^{'})$ $(a \in Act)$, if and only if $(\exists cc \in CC(C))(\exists D \subseteq C)$, $\ell \xrightarrow{cc,a,D} \ell^{'}$, $v^a \models cc$ and $v_a^{'} = v^a(D=0) \models Inv(\ell^{'})$. $v_a^{'} = v^a(D=0)$ represents that the abstract clock valuation $v_a^{'}$ is obtained in this way: for each clock variable x in the set $D \subseteq C$,its integer part variable I_x and abstract fractional order variable F_x^a are all reset to

0, and the values of all the other clock variables in $v_a^{'}$ are the same as in v^a.

- $\sim_i^{'} \subseteq Q^{'} \times Q^{'} (1 \le i \le n)$ is an epistemic equivalent relation, the epistemic equivalent to agent i between two abstract states (ℓ, v^a), $(\ell^{'}, v_a^{'})$ can be represented in the following: $(\ell, v^a) \sim_i^{'} (\ell^{'}, v_a^{'})$, if and only if the two abstract states satisfy the constraints in definition 11.
- $P_i^{l'}, P_i^{u'} : Q' \times Q' \to [0,1]$ $(1 \le i \le n)$ are matrices describing the lower and upper bounds for probabilities over epistemic relations such that $\forall q_1, q_2 \in Q'$, $P_i^{l'}(q_1, q_2) \le 1$, $P_i^{u'}(q_1, q_2) \le 1$, $P_i^{l'}(q_1, q_2) \le P_i^{u'}(q_1, q_2)$, $P_i^{l'}(q_1, Q') \le P_i^{u'}(q_1, Q')$.
- $V^{'} : Q^{'} \to 2^{PV}$ is a valuation function, for an abstract state (ℓ, v) we have $V((\ell, v)) = \bigcap_{i=1}^{n} V_{PTA}(\ell_i)$ (ℓ_i denotes the global location of the ith concrete state of this abstract state). That is to say, the set of propositional variables that are valid in an abstract state is the intersection of the sets of the propositional variables that are valid in each of its concrete state.
- $? : Q^{'} \to 2^{PV}$ is a valuation function, which assigns to each abstract state the set of propositional variables whose truth value is uncertain in this state.

Definition 13. (The three-valued semantics of PTACTLK on the abstract model)
$M^A = (Q', q_0^{'}, P^{l'}, P^{u'}, \dot{\sim}_1, .., \sim_n^{'}, P_1^{l'}, P_1^{u'}, ..., P_n^{l'}, P_n^{u'}, V', ?)$ is an abstract model of a probabilistic real time interpreted system M, $M^A, q \models \alpha$ denotes that the PTACTLK formula α holds on the abstract state q of the abstract model M^A. And the symbol M^A is omitted in the following satisfaction relations. Suppose p, ϕ and φ in the following are all PTACTLK formulae, the satisfaction relation "$\models$" is defined inductively as in the following:

- $q \models \begin{cases} p, \ \ iff \ \ p \in V^{'}(q), that\ is\ \forall s \in q, p \in V(s), \ \ p \in Ap; \\ \neg p, iff \ \ p \notin V^{'}(q), that\ is\ \forall s \in q, p \notin V(s), \ \ p \in Ap; \\ ?\,p, other\ cases. \end{cases}$
- $q \models \begin{cases} \phi \vee \varphi, & iff \ \ q \models \phi \ or \ q \models \varphi; \\ \neg(\phi \vee \varphi), & iff \ \ q \models \neg\phi \ and \ \ q \models \neg\varphi; \\ ?(\phi \vee \varphi), & other\ cases. \end{cases}$
- $q \models \begin{cases} \phi \wedge \varphi, & iff \ \ q \models \phi \ and \ q \models \varphi; \\ \neg(\phi \wedge \varphi), & iff \ \ q \models \neg\phi \ or \ q \models \neg\varphi; \\ ?(\phi \vee \varphi), & other\ cases. \end{cases}$
- $q \models \begin{cases} A(\phi U_I^{\ge p} \varphi), \ \ iff \ \forall \rho' \in f_{PTA}(q), \rho' \models \phi U_I \varphi, and \\ \qquad \sum_{\rho' \in f_{PTA}(q)} \Pr ob(\rho') \ \ge p; \\ \neg A(\phi U_I^{\ge p} \varphi), iff \ \forall \rho' \in f_{PTA}(q), \rho' \models \neg(\phi U_I \varphi), and \\ \qquad \sum_{\rho' \in f_{PTA}(q)} \Pr ob(\rho') \ \ge 1 - p; \\ ?\,A(\phi U_I^{\ge p} \varphi), other\ cases. \end{cases}$
- $q \models \begin{cases} A(\phi R_I^{\ge p} \varphi), \ \ iff \ (\forall \rho' \in f_{PTA}(q)), \rho' \models \phi R_I \varphi, and \\ \qquad \sum_{\rho' \in f_{PTA}(q)} \Pr ob(\rho') \ \ge p; \\ \neg A(\phi R_I^{\ge p} \varphi), iff \ (\forall \rho' \in f_{PTA}(q)), \rho' \models \neg(\phi R_I \varphi), and \\ \qquad \sum_{\rho' \in f_{PTA}(q)} \Pr ob(\rho') \ \ge 1 - p; \\ ?\,A(\phi R_I \varphi), other\ cases. \end{cases}$
- $q \models \begin{cases} K_i^{\ge p} \varphi, \ \ \ iff \ \ \sum_{q' \models \varphi \wedge q \sim_i q'} P_i^{l'}(q, q') \ge p; \\ \neg K_i^{\ge p} \varphi, \ iff \ \ \sum_{q' \models \neg\varphi \wedge q \sim_i^{'} q'} P_i^{l'}(q, q') \ge 1 - p; \\ ?\,K_i^{\ge p} \varphi, other\ cases. \end{cases}$
- $q \models \begin{cases} E_\Gamma^{\ge p} \varphi, \ \ iff \ \sum_{q' \models \varphi} \sum_{i \in \Gamma} P_i^{l'}(q, q') \ge p; \\ \neg E_\Gamma^{\ge p} \varphi, iff \ \sum_{q' \models \neg\varphi} \sum_{i \in \Gamma} P_i^{l'}(q, q') \ge 1 - p; \\ ?\,E_\Gamma^{\ge p} \varphi, other\ cases. \end{cases}$
- $q \models \begin{cases} D_\Gamma^{\ge p} \varphi, \ \ iff \ \sum_{q' \models \varphi} \prod_{i \in \Gamma} P_i^{l'}(q, q') \ge p; \\ \neg D_\Gamma^{\ge p} \varphi, iff \ \sum_{q' \models \neg\varphi} \prod_{i \in \Gamma} P_i^{l'}(q, q') \ge 1 - p; \\ ?\,D_\Gamma^{\ge p} \varphi, other\ cases. \end{cases}$

As $C_\Gamma^{\ge p} \varphi$ is the transitive closure of $E_\Gamma^{\ge p} \varphi$, the satisfaction relation of $C_\Gamma^{\ge p} \varphi$ on the abstract model is ignored. In the definitions above, $q \models ?\phi$ represents that the truth value of PTACTLK formula ϕ on the abstract state q is uncertain.

3.3 Property Preservation Theorem

The aim of abstraction is to simplify the original model of the system while preserving its properties, and in the following we prove that the satisfaction relations of PTACTLK formulae are preserved in the abstract model. That is, if a PTACTLK formula ϕ is satisfied by an abstract model M^A, it can be inferred that the formula ϕ is also satisfied in the original model M; if a PTACTLK formula ϕ is not satisfied by an abstract model M^A, it can be concluded that this formula is not satisfied by the original model M either. In other words, the abstract model is not only the upper approximation of the original model, but also its lower approximation.

Theorem 1. Suppose $M=(Q,q_0,P,\sim_1,...,\sim_n,P_1,...,P_n,V)$ is a probabilistic real time interpreted system, $M^A=(Q',q_0^{'},P^{l'},P^{u'},\sim_1^{'},..,\sim_n^{'},P_1^{l'},P_1^{u'},...,P_n^{l'},P_n^{u'},V',?)$ is the corresponding abstract model obtained according to the abstraction techniques presented above, and ϕ is a PTACTLK formula, we have the following conclusions: (1). If $M^A,s^{'}\models\phi$, then $M,s\models\phi$; (2). If $M^A,s^{'}\models\neg\phi$, then $M,s\models\neg\phi$.

4. A CASE STUDY

In this section, we present a simple communication protocol , which is adopted to illustrate the effectiveness of our abstraction techniques.

4.1 A Simple Communication Protocol

In our communication protocol, there are two agents: *Sender* and *Receiver*, and they are connected through a communication channel. *Sender* collects data from the environment and sends the collected data to *Receiver* via the channel. The responsibility of *Receiver* is to receive the data from *Sender*. The probabilistic timed automata of *Sender* and *Receiver* are presented in Fig.1 and Fig.2, respectively. Suppose that the communication channel is reliable, that is, the data can't be lost during the transmission. The protocol works as follows.

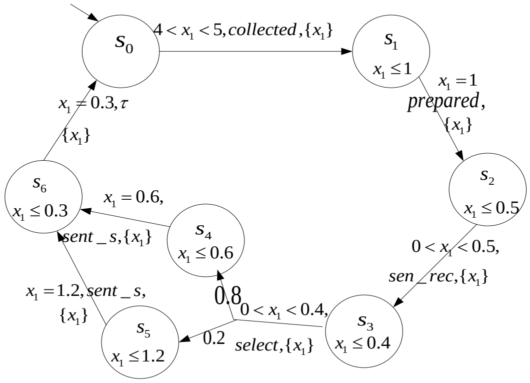

Figure 1. Probabilistic Timed Automaton Sender

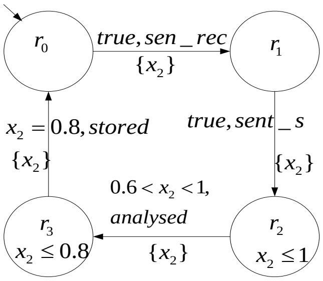

Figure 2. Probabilistic Timed Automaton Receiver

Agent Sender has one clock variable x_1, which is used to record the time that has elapsed in each location. In the initial location s_0, the value of x_1 is 0. Sender first collects data for 4 to 5 seconds, then it moves to location s_1, action *collected* represents that Sender has finished the work of collecting data. In order to send the collected data to Receiver, Sender prepares communication for 1 second and moves to location s_2, action *prepared* denotes that Sender has finished the work of preparing communication. In location s_2, *Sender* sends a synchronization signal *sen_rec* to *Receiver* within 0.5 second and moves to location s_3. Suppose that when sending data to Receiver , Sender uses two special ports: port1 and port2 , each port is connected with the channel. However, when Sender is to send data it can only selects one port randomly, and the probability of selecting port1 is 0.8, and port2 is 0.2. If port1 is selected, Sender has to wait for 0.6 second and then begins to send data via port1; if Sender selects port2, it needs to wait for 1.2 seconds. In location s_3, Sender selects a port randomly within 0.4 second. If the selected port is port1, then it moves to location s_4, else it moves to s_5. No matter which port is selected, Sender will reach to location s_6 after sending data to Receiver, and it will return to the initial location after staying in location s_6 for 0.3 second.

Agent Receiver has a clock variable x_2, and its initial location is r_0. Receiver will move to location r_1 from the initial location when it receives the synchronization signal *sen_rec* from Sender. After Receiver has received data from Sender it will move to location r_2, in which it takes 0.6 to 1 second for Receiver to analysis the received data, and after that Receiver will move to location r_3. Action *analysed* denotes that Receiver has completed the work of analysing the received data. In r_3, *Receiver* spends 0.8 second to store the processed data and then returns to the initial location r_0. Action *stored* represents that the work of storing the processed data has been completed.

In the two probabilistic timed automata, besides the edges that are labelled with probability explicitly, the transition probability for all the other edges is 1, which is ignored for simplicity.

In our communication protocol, a concrete state should be in this form: $((i,j),(v_{x_1},v_{x_2}))$, $(0\le i\le 6, 0\le j\le 3)$, where (i,j) is the global location of this state, the two components i, j represent that agents Sender, Receiver are in locations s_i, r_j , respectively; (v_{x_1}, v_{x_2}) is a clock valuation over the set of clock variables C={x_1, x_2}, which denotes that under this state the values of clock variables x_1, x_2 are v_{x_1}, v_{x_2} , respectively. As the ranges of the values of the clock variables x_1, x_2 in this protocol are $0 \le v_{x_1} \le 5, 0 \le x_2 \le 5$, respectively, and v_{x_1}, v_{x_2} are all real numbers, there are infinite states in this protocol, that is, the state space of our protocol is infinite.

4.2 Constructing the Abstract Model of the Protocol

The abstraction techniques presented in section 3 are applied to construct the abstract model of our communication protocol such that the infinite state space of the protocol can be simplified into a finite form.

For any concrete state $((i, j), (v_{x_1}, v_{x_2}))$ of this protocol, its clock valuation (v_{x_1}, v_{x_2}) is represented using the corresponding abstract discrete clock valuation. In this way, the concrete states whose clock valuations belong to the same abstract discrete clock valuation and global locations are the same can be combined into one abstract state, therefore the state space of the protocol can be greatly reduced and we can obtain the finite form of the infinite state space of our protocol. In the obtained finite form of the protocol, using the definition of epistemic equivalent to agent *Sender* (*Receiver*) between two abstract states, any abstract states satisfying the constraints of this definition can be combined into one equivalent class and the state space will be further simplified. At last, the corresponding abstract model of the communication protocol can be obtained. In Fig. 3, all the reachable states of the abstract model obtained using our abstraction techniques are given.

From the abstract model, it can be seen that using our abstraction techniques the infinite state space of the communication protocol has been simplified into a finite form which only has 9 abstract states.

Each state and its location invariant of the abstract model presented above is in the following:

$S_0 : ((0,0), ((0,0), (0,0))), Inv(0,0) = true;$

$S_1 : ((1,0), ((0,0), (4,\alpha))), Inv(1,0) = x_1 \le 1;$

$S_2 : ((2,0), ((0,0), (5,0))), Inv(2,0) = x_1 \le 0.5;$

$S_3 : ((3,1), ((0,0), (0,0))), Inv(3,1) = x_1 \le 0.4;$

$S_4 : ((4,1), ((0,0), (0,\alpha))), Inv(4,1) = x_1 \le 0.6;$

$S_5 : ((5,1), ((0,0), (0,\alpha))), Inv(5,1) = x_1 \le 1.2;$

$S_6 : ((6,2), ((0,0), (0,0))), Inv(6,2) = x_1 \le 0.3 \wedge x_2 \le 1;$

$S_7 : ((0,2), ((0,0), (0,\alpha))), Inv(0,2) = x_2 \le 1;$

$S_8 : ((0,3), ((0,\alpha), (0,0))), Inv(0,3) = x_2 \le 0.8.$

That is, in the obtained abstract model $M^A = (Q', q_0^{'}, P^{l'}, P^{u'}, \tilde{\sim}_1, .., \sim_n^{'}, P_1^{l'}, P_1^{u'}, ..., P_n^{l'}, P_n^{u'}, V', ?)$ of the communication protocol, we have:the abstract state space $Q^{'} = \{S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8\}$; the initial abstract state $q_0^{'} = S_0$; the state transition probabilities: $P^{l'}(S_3, S_4) = P^{u'}(S_3, S_4) = 0.8,$ $P^{l'}(S_3, S_5) = P^{u'}(S_3, S_5) = 0.2$, that is, the transition probability from state S_3 to S_4 is 0.8 , and from S_3 to S_5 is 0.2, the transition probability of all the other neighboring states is 1.

each abstract state and itself is epistemic equivalent to agent *Sender*, that is, for each state $Si \in Q^{'}(0 \le i \le 8)$, $Si \sim_{Sender}^{'} Si$ holds. Similarly, it is also hold to agent *Receiver*.

4.3 Model Checking the Communication Protocol

The first property to be verified is: in the communication protocol, agent *Sender* considers that such a behaviour holds with probability greater than or equal to 0.6, in which after *Sender* sends a synchronization signal *sen_rec* to *Receiver*, *Receiver* will receive the data between 0.5 and 1 second. The property can be expressed by the PTACTLK formula $\phi_1 = K_{Sender}^{\ge 0.6}(sen\_rec \wedge AF_{(0.5,1)} receive)$, where *receive* denotes that *Receiver* has received the data.

The second property to be verified is: agent *Sender* considers that such a behaviour holds with probability greater than or equal to 0.15, in which after *Sender* sends a synchronization signal *sen_rec* to *Receiver*, *Receiver* will receive the data between 1.6 and 2 seconds. The property can be expressed by the PTACTLK formula $\phi_2 = K_{Sender}^{\ge 0.15}(sen\_rec \wedge AF_{(1.6,2)} receive)$.

In the obtained abstract model M^A, as *Sender* moves to the global state S_3 after its sending the synchronization signal, and *Receiver* moves to state S_6 after its receiving the data, then we only need to check the time interval and the transition probability between the two states . In the abstract model, it can be easily seen that there are two paths from S_3 to S_6, in the first path, S_3 moves to S_6 via S_4, that is, $S3 \to S4 \to S6$; and for the second one, the intermediate state is S_5, that is, $S3 \to S5 \to S6$.

We first consider path: $S3 \to S4 \to S6$. As the time interval from S_3 to S_4 is (0,0.4), and the one from S_4 to S_6 is 0.6, then it can be inferred that on this path the time interval between S_3 and S_6 is (0.6,1). What's more, the transition probability from S_3 to S_4 is 0.8, and the one from S_3 to S_6 is 1, then it can be learned that the transition probability from S_3 to S_6 is $0.8\times1=0.8$ on this path. From the above analysis, we can conclude that after *Sender* sends a synchronization signal *sen_rec* to *Receiver*, *Receiver* will receive the data between 0.6 and 1 second with probability 0.8. In other words, it satisfies the first property: $M^A \models \phi_1$.

Let us learn about the second path: $S3 \to S5 \to S6$. As the time interval from S_3 to S_5 is (0,0.4), and the one from S_5 to S_6 is 1.2, then it can be deduced that on this path the time interval between S_3 and S_6 is (1.2,1.6). In addition, the transition probability from S_3 to S_5 is 0.2, and the one from S_5 to S_6 is 1, then it can be computed that the transition probability from S_3 to S_6 is $0.2\times1=0.2$ on this path. From the above analysis, we can conclude that after *Sender* sends a synchronization signal *sen_rec* to *Receiver*, *Receiver* will receive the data between 1.2 and 1.6 seconds with probability 0.2. Thus, it can be seen that time interval (1.2,1.6) is not a subset of (1.6,2), in other words, it doesn't satisfy the second property: $M^A \not\models \phi_2$.

As in Section 3.3 we have proved that the abstract model obtained according to our abstraction techniques is the upper approximation of the original model , it can be concluded that the communication protocol satisfies the first property: $M \models \phi_1$; what's more, we have also proved that the abstract model is the lower approximation of the original model, then it can be deduced that our communication protocol does not satisfy the second property , that is, $M \not\models \phi_2$.

Using our abstraction techniques, the infinite state space of the communication protocol has been reduced to 9 states of the abstract model, which illustrates the effectiveness of our abstraction techniques.

5. CONCLUSIONS AND FUTURE WORK

In order to alleviate the state space explosion problem in model checking probabilistic real-time temporal logic of knowledge, we present the abstraction techniques. The infinite state space of a probabilistic real time interpreted system can be simplified into a finite form using abstract discrete clock valuations; and using the definition of epistemic equivalent to an agent between two abstract states, the corresponding equivalent relations can be deduced and which can be used to combine the abstract states, therefore, the state space of a probabilistic real time interpreted system can be further simplified. We define the abstract model of a probabilistic real time interpreted system, present the three-valued semantics of probabilistic real-time temporal logic of knowledge on the abstract model, and prove that the abstract model obtained using the abstraction techniques is not only the upper approximation of the original model but also its lower approximation. At last, a simple communication protocol is adopted to illustrate the effectiveness of our abstraction techniques.

There are many interesting avenues for future research. When it can not be known whether a PTACTLK property is satisfied by the abstract model, that is $M^A \models ?\phi$, we have no way of deciding whether the original model satisfies the property, in this case the abstract model needs to be refined. Therefore, refining the abstract model according to the reason that causes the failure of abstraction is a valuable direction for future research.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant Nos.61003288, 61111130184), the People with Ability Foundation of Jiangsu University(12JD061).

REFRENCES:

[1] [1] Edmund M. Clarke, Orna Grumberg and Doron A. Peled, “Model Checking”, *MIT Press*, Cambridge, MA, 2000.

[2] R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, “Reasoning about Knowledge”, *MIT Press*, Cambridge, MA, 1995.

[3] Lomuscio, W. Penczek, and B. Wozna, “Bounded model checking for knowledge and real time”, *Artif. Intell.*, 2007, pp.1011-1038.

[4] Cormac Flanagan, Patrice Godefroid, “Dynamic partial-order reduction for model checking software”, *ACM SIGPLAN Notices*, Vol. 40, No. 1, 2005, pp110-121.

[5] Prasad Sistla, Patrice Godefroid, “Symmetry and reduced symmetry in model checking”. *ACM Transactions on Programming Languages*

and Systems, Volume 26 Issue 4, 2004, pp702-734.

[6] Conghua Zhou, Bo Sun, Zhifeng Liu, “Abstraction for model checking multi-agent systems”. *Frontiers Of Computer Science in China*, Volume 5, Number 1, June 2010, pp14-25.

[7] Edmund M. Clarke, Orna Grumberg, David E. Long., “Model Checking and Abstraction”. *ACM Transactions on Programming Languages and Systems*, 1992, 16 (5): Pages 1512-1542.

[8] Grumberg, “3-Valued Abstraction for (Bounded) Model Checking”, *Lecture Notes In Computer Science*, 2009, Vol.5799: pp21-21.

[9] Orna Grumberg, “Abstraction and Refinement in Model Checking”. *Lecture Notes In Computer Science*, 2006, Vol4111, 219-242.

[10] Edmund M. Clarke, Flavio Lerda, and Muralidhar Talupur, “An Abstraction Technique for Real-Time Verification”, *Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems*, 2007, pp1-17.

[11] Sascha Konrad, Betty H. C. Cheng, “Real-time specification patterns”. *Proceedings of the 27th international conference on Software engineering*, St.Louis, Missouri(USA), May15–21, 2005, Pages 372-381.

[12] Carlos Areces, Guillaume Hoffmann and Alexandre Denis, “Modal Logics with Counting”, *Lecture Notes in Computer Science*, 2010, Volume 6188, pp98-109.

[13] Christel Baier, Joost- Pieter Katoen, “Principles of Model Checking”. *MIT Press*, 2008 , Page 678.

[14] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. “Automatic verification of real-time systems with discrete probability distributions”. *Theoretical Computer Science*, 282:2002 pp101-150.

[15] Johan Bengtsson and Wang Yi, Timed Automata: Semantics, Algorithms and Tools. *Lecture Notes in Computer Science*, 2004, Volume 3098/2004, pp87-124.

[16] R. Alur, D. Dill, “A theory of timed automata”, *Theoret. Comput. Sci*. 126 (1994) 183–235.

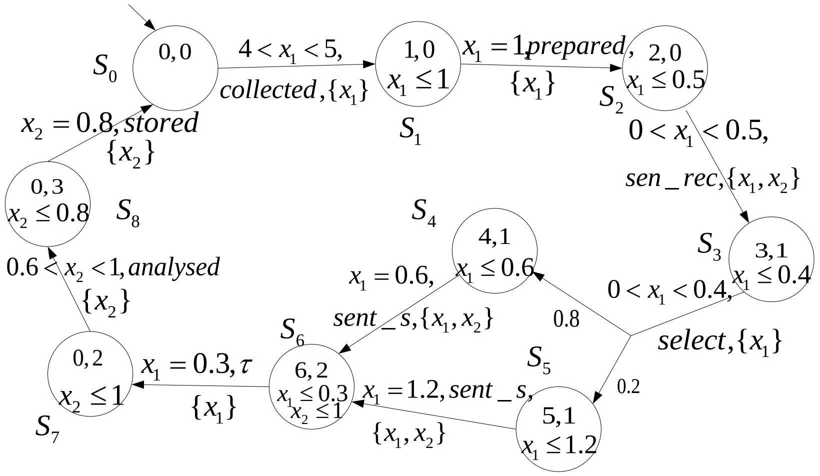

Figure 3. The abstract model of the communication protocol