International Journal of Control and Automation
Vol.7, No.2 (2014), pp.251-262
http://dx.doi.org/10.14257/ijca.2014.7.2.24

A Study of Complex Property Counterexample Generation Method for Markov Model

Mingyu Ji[1], Zhiyuan Chen[2] and Yanmei Li[2]

[1]*College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China*
[2]*College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*

jimingyu@hrbeu.edu.cn, chenzhiyuan@hrbeu.edu.cn, liyanmei@hrbeu.edu.cn

Abstract

With the wide application of probabilistic systems, the research of counterexample generation for probabilistic system with model checking has attracted wide attention. For counterexample of complex parametric system, proposes a counterexample generation algorithm for multiple until constraint formulae of probabilistic computation tree logic act on continuous time probabilistic model and gives another counterexample computation method based on automaton theory. At last, the example analysis is given. The theoretical analysis and example result show that the feasibility and validity of the method.

Keywords: *probabilistic system, model checking, until formula, automaton, counter example generation*

1. Introduction

With the increasing complexity of computer hardware and software systems, how to ensure system accuracy and reliability becomes an increasingly pressing issue. For concurrent systems, because of its inherent nondeterministic, the problem is more difficult. In the past few decades, many researchers have made great efforts to solve this problem and have made important progresses.

In many of the theories and methods proposed for this purpose, the model checking has become an important mean for its simplicity and high automaton [1] and has got a great success in verification of computer hardware, communication protocol, software system [2-5].

As a branch of model checking, the probabilistic model checking technique [6] has been widely used in many fields, such as communication protocol [7], biological modeling [8] and so on because of it can describe the random characteristics, time characteristics and spatial characteristics that exist in system mode [9].

Because model checking can provides counterexample information of properties violation, so that it becomes an efficient mean for automatic detecting faults. Now, the counterexample solution method of the probabilistic model property become important research topics that international experts and scholars interest.

The literature [10] and [11] study the counterexample generation method for continuous time Markov model and give the comparative analysis of the various counterexample

Supported by the Fundamental Research Funds for the Central Universities under Grant No.DL11BB08 and the National Natural Science Foundation of China under Grant No. 71001023

ISSN: 2005-4297 IJCA

algorithms. The literature [12] shows the counterexample algorithm of reachability probability of until formula under transition step constraint for discrete time Markov chain. The literature [13] uses probabilistic timed automata as system model, and gives the counterexample algorithm of until formula property based on weighted directed graph. Literature [14] expresses the probabilistic timed automata semantic as Markov decision process, and then studies the counterexample generation algorithm of until formula.

Based on the above literatures, the counterexample solution of probability model is mainly based on basic probability model and makes the time constraint until formulae as the properties to be verified. For systems biology, multiple until formulae properties verification problem of probability model that supporting complex constraint description is widely researched. The literature [15] aims to stratified continuous time Markov chain model, validates continuous stochastic logic multiple until formula with transition step time constraints, puts forward the verification algorithm and gives the description of algorithm complexity and example analysis, but the counterexample representation and generation method not be involved.

In this paper, we focus on counterexample solution problem solving, and propose a counterexample generating method of continuous probability model for multiple until formulae properties with transition step and transition resource consumption constraints.

The rest of paper is organized as follows: Section 2 presents the necessary background on probability model with complex parameters. In Section3 introduces the syntax of temporal logic and semantics of multiple until formulae. Section 4 explains how to solve counterexample and describes corresponding algorithm. Section 5 gives another counterexample solution method based on automaton product model. In section 6, the calculated results of formulae counterexample are given with two methods based on the specific instance. Finally, in Section 7 we give the conclusions and directions of future research.

2. Probability model

Probability model checking is based on the construction and analysis of a probabilistic model and has been widely applied in the analysis and design process of software and hardware system. The common probability models include Markov chains and Markov decision processes. These models can be extended in several directions, including the addition of rewards and continuous time.

Definition 1. A labeled discrete time Markov chain(DTMC) is a tuple M = (S,AP,L,P) where S is a finite set of states, AP is a set of atomic propositions, $L: S \to 2^{AP}$ is a labeling function that assigns to each $s \in S$ a set $L(s)$ of atomic propositions, and $P: S \times S \to [0,1]$ is a probability matrix satisfying $\sum_{s' \in S} P(s,s') \in \{0,1\}$ for all $s \in S$.

Definition 2. A labeled continuous time Markov chain (CTMC) is a tuple M =(S,AP,L,R) where $R: S \times S \to R_{\geq 0}$ is transition rate matrix, and other notations are defined as for DTMC.

Because DTMC and CTMC have no ability to describe the characteristics of resource consumption in the transition process, so we introduce the definition of Markov reward model (MRM) based on Markov chain model.

Definition 3. Corresponding to the DTMC and CTMC, the MRM are called DTMRM and CTMRM respectively, and expressed as M =(S,AP,L,P,N) and M =(S,AP,L,R,N) where $N: S \times S \to R_{\geq 0}$ is transition reward matrix, and other notations are defined as for DTMC and CTMC.

This paper mainly aims at the verification of CTMRM properties. In CTMRM, the transition probability from state s_1 to a particular state s_2 within t time units meets the exponential distribution and is given by $\frac{R(s_1,s_2)}{E(s_1)}(1-e^{-E(s_1)\cdot t})$, where $E(s_1)=\sum_{s\in S'} R(s_1,s)$ denotes the total rate at which any transition outgoing from state s_1 is taken.

3. Temporal logic

In probabilistic model checking process, the property to be verified is generally described by some sort of temporal logic formula. At present, the temporal logic applied to discrete model and continuous model include probabilistic computation tree logic (PCTL) [16] and continuous stochastic logic(CSL) [17] and so on. Among them, PCTL, as a branching time temporal logic, introduces probability operator on the base of computation tree logic, which can achieve quantitative validation analysis of the model property.

Definition 4. The syntax of PCTL is defined as the state formulae set ϕ and the path formulae set φ, as follows:

$$\phi ::= true \mid f \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid P_{\Delta p}(\varphi)$$

$$\varphi ::= \bigcirc \phi \mid \phi_1 \cup \phi_2 \mid \phi_1 \cup^n \phi_2$$

Where $p \in [0,1]$ be a real number, $f \in AP$ be a atomic proposition, $\Delta \in \{=, <>, \geq, \leq\}$ be a comparison operator, and $n \subseteq R_{\geq 0}$ a nonempty transition step interval. We use $\phi_1, \phi_2, \ldots$ for state formulae and $\varphi_1, \varphi_2, \ldots$ for path formulae.

Because of PCTL path formulae have the limitation that any temporal operator must follows a state formula and do not have the ability to describe the transition resource consumption in transition process of probability model. Next we introduce improved probabilistic reward computation tree logic (PRCTL*) which state formulae are defined same as state formulae of PCTL, which path formulae as follows:

$$\varphi ::= \phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc_r^n \varphi \mid \varphi_1 \cup_r^n \varphi_2$$

Where $n \subseteq R_{\geq 0}$ and $r \subseteq R_{\geq 0}$ respectively represent transition step interval and transition resource consumption interval, $\bigcirc$ is next operator, $\cup$ is until operator. Path formula $\phi_1 \cup_{\leq r_1}^{\leq n_1} \phi_2$ asserts that formula ϕ_2 in the current path is satisfied in some state ultimately, and all states before that state in the path satisfy formula ϕ_1, and transition step number of the path is less than or equal to n_1, meanwhile the total resource consumption in transition process is less than or equal to r_1.

This paper research counterexample problem of state formula $P_{\Delta p}(\phi_1 \bigcup_{\le r_1}^{\le n_1} \phi_2 \bigcup_{\le r_2}^{\le n_2} \cdots \bigcup_{\le r_{k-1}}^{\le n_{k-1}} \phi_k)$ based on single until constraint path formula $\phi_1 \bigcup_{\le r_1}^{\le n_1} \phi_2$, where Δ takes the comparison operator $\le$.

4. Counterexample of until Constraint Path Formula

In order to avoid the complex integral computation in the solving probabilistic counterexample of single until formulae $\phi_1 \bigcup^{\le h} \varphi$ and $\phi_1 \bigcup^{\le t} \varphi$ (h is transition step constraint and t is time constraint), the general approach is as follows: do discretization and uniformization pretreatment first, and then transform the original model into discrete probability model, so the problem solution was transformed into the computation of counterexample path sets in the corresponding weighted directed graph using the shortest path algorithm. The complexity analysis and the detailed algorithm can be found in literature [12]. In this paper, the model pretreatment process method of CTMRM is similar to the above.

Definition 5. Pretreatment of CTMRM

Let M =(S,AP,L,R,N) be CTMRM, then the model after discretization and uniformization named uniformized continuous time Markov reward model (UDTMRM). The structure of UDTMRM is similar to DTMRM and is defined as tuple (S,AP,L,P_u,N), where probability distribution $P_u(s_1,s_2) = \frac{R(s_1,s_2)}{E(s_1)} \bullet \frac{E(s_1)}{\max\{E(s_i)\}}$ and $\sum_{s' \in S} P_u(s,s') = 1.$

Definition 6. Counterexample semantic description of $P_{\le p}(\phi_1 \bigcup_{\le r_1}^{\le n_1} \phi_2 \bigcup_{\le r_2}^{\le n_2} \cdots \bigcup_{\le r_{k-1}}^{\le n_{k-1}} \phi_k)$

For state s in UDTMRM, counterexample semantic of $P_{\le p}(\phi_1 \bigcup_{\le r_1}^{\le n_1} \phi_2 \bigcup_{\le r_2}^{\le n_2} \cdots \bigcup_{\le r_{k-1}}^{\le n_{k-1}} \phi_k)$ as follows:

$s \left| \neq P_{\le p}(\phi_1 \bigcup_{\le r_1}^{\le n_1} \phi_2 \bigcup_{\le r_2}^{\le n_2} \cdots \bigcup_{\le r_{k-1}}^{\le n_{k-1}} \phi_k) \right.$

$\Leftrightarrow prob(s, \phi_1 \bigcup_{\le r_1}^{\le n_1} \phi_2 \bigcup_{\le r_2}^{\le n_2} \cdots \bigcup_{\le r_{k-1}}^{\le n_{k-1}} \phi_k) > p$

$\Leftrightarrow$ Exist the finite path set C asserts that $\sum_{\sigma \in C} Prob(\sigma) > p$ and $\forall \sigma \in C, \sigma[0] = s \wedge \sigma \left| = \varphi \wedge pre(\sigma) \right| \neq \varphi.$

Among them, the path in finite path set C must satisfy transition step and transition consumption constraints of multiple until formulae.

We now present our algorithm based on the counterexample semantic above. The algorithm proceeds in four steps: First, The UDTMRM is transformed into a weighted directed graph (WDG). Afterwards, based on WDG, the counterexample evidence paths of single until formula with transition step constraints are found. Then, we verify the validity of resource consumption constraint for each evidence path. At last, we calculate the counterexample path probability and give the final counterexample path set.

Algorithm 1. Counterexample solution algorithm


```
1:for(i=1 to k-1)
2:  for( each s∈S )  { pn[s][i]=0; }
3:m=1; pr=0; s=s0; s'=null;
4:while (pr≤p)
5:{for(i=1 to k-1)
6:  { ρ'[m][i]=null;}
7: for(i=1 to k-1)
8:   {Next: do
9:    { pns[i]++;
10:     ρ=SP(s, φ[i],n[i], φ[i+1], pn[s][i]);
11:    if ( ρ==null)
12:     {if (i<>1)
13:      {i--;  ρ'[m][i]= null; s=s'; pn[s][i]++;
14:      goto next;}
15:    else  break next; }
16:    x=first( ρ ); totalr=0; flag=true;
17:    while((y=next( ρ ,x))!=null)
18 :     { totalr= totalr +R(x,y); x= y;
19 :       if (totalr>r[i])
20:         { flag=false; break;}}
21:   } While(flag <>true)
22:  ρ'[m][i]= ρ ;
23:  s'=s; s=last( ρ ); i++;}
24: pr=pr+ prob( ρ'[m] );
25: m=m+1;}
26: if(pr≥p) return ( ρ'[1]…ρ'[m−1] );
27: else  return null;
```

The basic algorithm for counterexample solution of $P_{\leq p}(\phi_1 \bigcup_{\leq r_1}^{\leq n_1} \phi_2 \bigcup_{\leq r_2}^{\leq n_2} \cdots \bigcup_{\leq r_{k-1}}^{\leq n_{k-1}} \phi_k)$ is presented as above. The input include state set S of UDTMRM, state s needed to verified, transition step constraint array n which length is k-1 and transition resource consumption constraint array r which length is k-1, and state formulae set array ϕ that contained in multiple until formula. The output is multiple until constraint formulae counterexample path set or null set.

The core algorithm of algorithm 1 is function named SP on line 10, which can return the next transition step constraint satisfaction path of the i-th single until formula in multiple until formulae. For DTMC and PCTL state formulae $\phi \bigcup^{h} \varphi$, the core algorithm can complete the calculation of k shortest counterexample path. The time complexity of core algorithm is $O(hm + hk\log(m/n))$, where n and m are the number of states and transitions, h is the step bound, and k is the number of shortest paths.

5. Automaton Computation Method

In the following, we give the definition of Multi Until Formula Nodeterministic Finite Automaton (MUFNFA) by the semantic analysis of multi until formula of PRCTL* and give the example description.

Definition 7. NFA

NFA is a quintuple $B = (\Sigma, Q, Q_0, \delta, F)$, wherein Σ is a non-empty finite alphabet, Q be a finite state set, $Q_0 \subseteq Q$ be initial state set, $\delta : Q \times \Sigma \to 2^Q$ be transition function, $F \subseteq Q$ be terminated state set.

Definition 8. MUFNFA

For path Formula $\varphi = f_1 \bigcup f_2 \bigcup \cdots \bigcup f_k$, the corresponding automaton named MUFNFA is expressed as $B_\varphi = (\Sigma, Q, Q_0, \delta, F)$. Where alphabet $\Sigma = 2^{\{f_1, f_2, \cdots, f_k\}}$, $Q = \{q_0, q_1, \ldots q_{k-1}, q_k, \perp\}$, $\{q_k, \perp\}$ is absorbing states set, the initial state set $Q_0 = \{q_0\}$, and terminate state set $F = \{q_k\}$. For the input letter a, if $0 \le i \le k-1$ and $i \le j \le i+1$, then when $f_i \in a$, transition function $\delta(q_i, a) = \{q_i, q_j\}$, when $f_i \notin a$, $\delta(q_i, a) = \perp$.

Example 1 MUFNFA example

Figure 1 is the corresponding MUFNFA of multi until formula $f_1 \bigcup f_2$, where q_0 is the initial state and q_2 is terminate state.

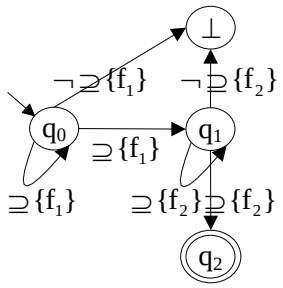


Figure 1. NFA of formula $f_1 \bigcup f_2$

Next, we give the formal definition of product model $M \times B_\varphi$ of nondeterministic finite automaton B_φ corresponding to the multi until formula and DTMRM M.

Definition 9. Product model

Let $B_\varphi = (\Sigma, Q, Q_0, \delta, F)$ be the corresponding nondeterministic finite automaton of multi until formula φ, $M = (S, P, AP, L, N, v)$ be a DTMRM, then their synchronization evolution product model denoted by $M \times B_\varphi = (S^*, P^*, AP^*, L^*, N^*, v^*)$.

The symbols of $M \times B_\varphi$ are described below.

State set is expressed as:

$S^* = \{s^* = (s,q) \mid s = \in S \wedge q \in Q\}$,

Transition probability is expressed as:

$$p^*((s,q_i),n,(s',q_j)) = \begin{cases} p(s,n,s') & \text{if} \quad q_j \in \delta(q_i, L(s')) \\ 0 & \text{otherwise} \end{cases}$$

Labeling function is expressed as:

$$L^*(s,q_i) = \begin{cases} L^*(s) \cup \{accept\} & if \ q_i \cap F \neq \varnothing \\ \varnothing & if \ \perp \in q_i \\ L^*(s) & \text{otherwise} \end{cases}$$

Atomic proposition set is expressed as:

$AP^* = AP \bigcup \{accept\}$

Transition consumption is expressed as:

$N^*(S^* \times S^*) = N(S \times S)$

The initial distribution set is expressed as:

$$v^*(s,q) = \begin{cases} v(s) & if \ q \in \delta(q_0, L(s)) \\ 0 & \text{otherwise} \end{cases}$$

For DTMRM M and given PRCTL* multi until formula $\varphi = f_1 \bigcup_{r_1}^{n_1} f_2 \bigcup_{r_2}^{n_2} \cdots f_k$, where $n_i = [n_{il}, n_{ih}]$ and $r_i = [r_{il}, r_{ih}]$, the computation method of formula satisfiability probability for specified state s in DTMRM as follows:

First formula φ is represented as nondeterministic finite automaton B_φ, making it as receivers of path set of original DTMRM, and then constructs product model of original DTMRM and nondeterministic finite automaton, and then calculates the accepting states reachable probability starting at some product model initial state s^* in specified constraints of product model, viz. $P_s^M(\varphi) = P_{s^*}^{M \times B}(\Diamond_r^n accept)$ [18]. Then, we can use the corresponding method to solve the counterexample path sets.

6. Example Analysis

In this section, we give the detail descriptions of counterexample set generation process according to two methods mentioned in the above paragraphs, that is the counterexample solution algorithm and reachability probability calculation method based on automaton product model.

Now, consider the initial state of UDTMRM and the formula $P_{\leq 0.05}(a \bigcup_{\leq 40}^{\leq 2} b \bigcup_{\leq 15}^{\leq 2} c \bigcup_{\leq 20}^{\leq 2} d)$ to be verified in Figure 2, we give the detail description of the formula counterexample generation process by counterexample solution algorithm. In this paper, we do not specify the weighted directed graph of model.

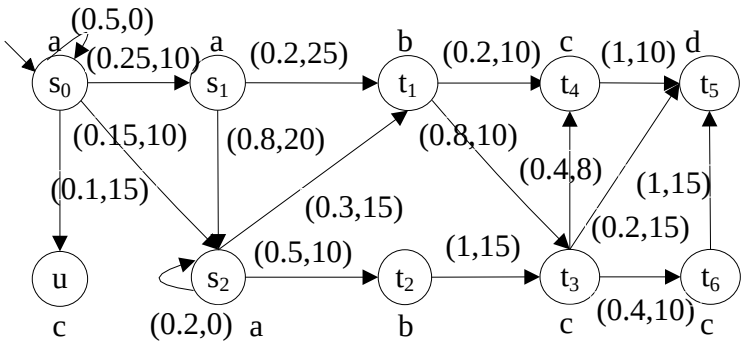


Figure 2. An UDTMRM example

The path sets ρ_i , probability value, transition step and transition resource consumption that satisfy until formula of $P_{\leq 0.05}(a \bigcup_{\leq 40}^{\leq 2} b \bigcup_{\leq 15}^{\leq 2} c \bigcup_{\leq 20}^{\leq 2} d)$ as follows:

For formula $a \bigcup b$:

$\rho_0 = s_0 s_2 t_1 = 0.045$, transition step is 2 and transition resource consumption is 25;

$\rho_1 = s_0 s_1 t_1 = 0.05$, transition step is 2 and transition resource consumption is 35;

$\rho_2 = s_0 s_1 s_2 t_1 = 0.06$, transition step is 3 and transition resource consumption is 45;

$\rho_3 = s_0 s_1 s_2 t_2 = 0.1$, transition step is 3 and transition resource consumption is 40;

$\rho_4 = s_0 s_2 t_2 = 0.075$, transition step is 2 and transition resource consumption is 20;

For formula $b \bigcup c$:

$\rho_5 = t_1 t_4 = 0.2$, transition step is 1 and transition resource consumption is 10;

$\rho_6 = t_1 t_3 = 0.8$, transition step is 1 and transition resource consumption is 10;

$\rho_7 = t_2 t_3 = 1$, transition step is 1 and transition resource consumption is 15;

For formula $c \bigcup d$:

$\rho_8 = t_4 t_5 = 1$, transition step is 1 and transition resource consumption is 10;

$\rho_9 = t_3 t_5 = 0.2$, transition step is 1 and transition resource consumption is 15;

$\rho_{10} = t_3 t_6 t_5 = 0.4$, transition step is 2 and transition resource consumption is 25;

$\rho_{11} = t_3 t_4 t_5 = 0.4$, transition step is 2 and transition resource consumption is 18;

Combining with the constraint conditions of the multiple until formula, the ultimate evidence paths in counterexample sets include:

$\rho_0\rho_5\rho_8 = s_0s_2t_1t_4t_5 = 0.009$;

$\rho_0\rho_6\rho_9 = s_0s_2t_1t_3t_5 = 0.0072$;

$\rho_0\rho_6\rho_{11} = s_0s_2t_1t_3t_4t_5 = 0.0144$;

$\rho_4\rho_7\rho_{11} = s_0s_2t_2t_3t_4t_5 = 0.03$;

$\rho_1\rho_6\rho_{11} = s_0s_1t_1t_3t_4t_5 = 0.016$;

$\rho_4\rho_7\rho_9 = s_0s_2t_2t_3t_5 = 0.015$;

$\rho_1\rho_5\rho_8 = s_0s_1t_1t_4t_5 = 0.01$;

$\rho_1\rho_6\rho_9 = s_0s_1t_1t_3t_5 = 0.008$;

Final, we can get the counterexample path sets that are all greater than 0.05 and the counterexample path set whose path number is least and whose probability sum is biggest is $(s_0s_2t_2t_3t_4t_5, s_0s_1t_1t_3t_4t_5, s_0s_2t_2t_3t_5)$. The path set probability sum is 0.061.

Now, based on computation method provided by this paper we give the detailed description of calculation process of multi until probability formula $P(s_0, a\bigcup_{\le 40}^{\le 2} b\bigcup_{\le 15}^{\le 2} c\bigcup_{\le 20}^{\le 2} d)$ for the system model in Figure 2.

First, for multi until formula $a\bigcup b\bigcup c\bigcup d$, we construct nondeterministic finite automaton which can be shown in Figure 3, then construct the synchronous evolution product model with original DTMRM model in Figure 2. The product model are shown in Figure 4, where initial state set is $\{s_0q_0\}$, accepting state sets are respectively $\{t_5q_4\}$, accepting state set are distinguished by dashed line ellipse and the states including $\perp$ and the transitions related with $\perp$ are all omitted.

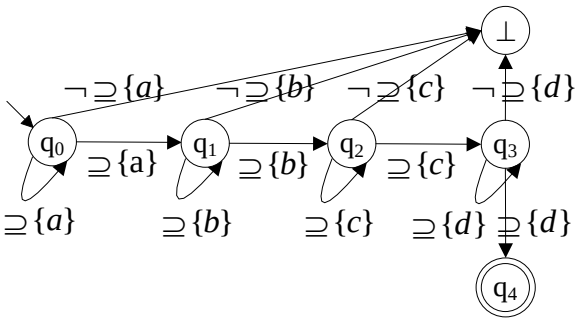


Figure 3. NFA of formula $a\bigcup b\bigcup c\bigcup d$

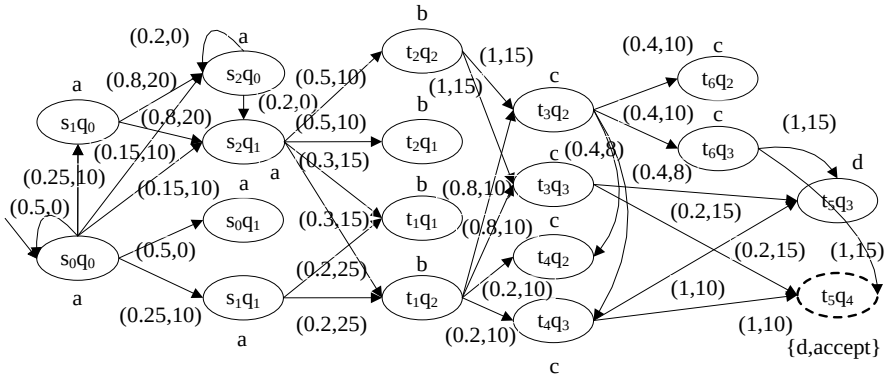


Figure 4. Product model of DTMRM and $a \bigcup b \bigcup c \bigcup d$

Second step, we find the formula satisfying path sets of product model.

For product model of Figure 4, the satisfying path set of path formula $a \bigcup b \bigcup c \bigcup d$ starting at the initial state be expressed as:

$\{(s_0q_0)^+(s_1q_0)(s_2q_0)^*(s_2q_1)(t_1q_2)(t_3q_2)(t_6q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_0)(s_2q_0)^*(s_2q_1)(t_1q_2)(t_3q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_0)(s_2q_0)^*(s_2q_1)(t_1q_2)(t_3q_2)(t_4q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_0)(s_2q_0)^*(s_2q_1)(t_1q_2)(t_4q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_0)(s_2q_0)^*(s_2q_1)(t_2q_2)(t_3q_2)(t_6q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_0)(s_2q_0)^*(s_2q_1)(t_2q_2)(t_3q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_0)(s_2q_0)^*(s_2q_1)(t_2q_2)(t_3q_2)(t_4q_3)(t_5q_4);$

$(s_0q_0)^+(s_2q_0)^*(s_2q_1)(t_1q_2)(t_3q_2)(t_6q_3)(t_5q_4);$

$(s_0q_0)^+(s_2q_0)^*(s_2q_1)(t_1q_2)(t_3q_3)(t_5q_4);$

$(s_0q_0)^+(s_2q_0)^*(s_2q_1)(t_1q_2)(t_3q_2)(t_4q_3)(t_5q_4);$

$(s_0q_0)^+(s_2q_0)^*(s_2q_1)(t_1q_2)(t_4q_3)(t_5q_4);$

$(s_0q_0)^+(s_2q_0)^*(s_2q_1)(t_2q_2)(t_3q_2)(t_6q_3)(t_5q_4);$

$(s_0q_0)^+(s_2q_0)^*(s_2q_1)(t_2q_2)(t_3q_3)(t_5q_4);$

$(s_0q_0)^+(s_2q_0)^*(s_2q_1)(t_2q_2)(t_3q_2)(t_4q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_1)(t_1q_2)(t_3q_2)(t_6q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_1)(t_1q_2)(t_3q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_1)(t_1q_2)(t_3q_2)(t_4q_3)(t_5q_4);$

$(s_0q_0)^+(s_1q_1)(t_1q_2)(t_4q_3)(t_5q_4)\}$

Wherein symbol * indicates finite repetition more than or equal to zero times and symbol + indicates finite repetition more than zero times [19].

Third step, we filter reachable path sets under the transition step constraints and transition resource consumption constraints in multi until formula $a \bigcup_{\le 40}^{\le 2} b \bigcup_{\le 15}^{\le 2} c \bigcup_{\le 20}^{\le 2} d$, and then obtain the final satisfying path sets.

The final constraints satisfying path sets of $a \bigcup_{\leq 30}^{\leq 2} b \wedge c \bigcup_{\leq 40}^{\leq 3} d$ are:

$\{(s_0q_0)(s_1q_1)(t_1q_2)(t_3q_3)(t_5q_4);$
$(s_0q_0)(s_1q_1)(t_1q_2)(t_3q_2)(t_4q_3)(t_5q_4);$
$(s_0q_0)(s_1q_1)(t_1q_2)(t_4q_3)(t_5q_4);$
$(s_0q_0)(s_2q_1)(t_1q_2)(t_3q_3)(t_5q_4);$
$(s_0q_0)(s_2q_1)(t_1q_2)(t_3q_2)(t_4q_3)(t_5q_4);$
$(s_0q_0)(s_2q_1)(t_1q_2)(t_4q_3)(t_5q_4);$
$(s_0q_0)(s_2q_1)(t_2q_2)(t_3q_3)(t_5q_4);$
$(s_0q_0)(s_2q_1)(t_2q_2)(t_3q_2)(t_4q_3)(t_5q_4)\}$

Last step, calculate the transition probability of final path sets and obtain the final calculation results: $P(s_0, a \bigcup_{\leq 40}^{\leq 2} b \bigcup_{\leq 15}^{\leq 2} c \bigcup_{\leq 20}^{\leq 2} d) = 0.1096$. The results is greater than 0.05 and the corresponding counterexamples path sets are exist and be the same with the results computed by first method.

7. Conclusion and Future Work

In this paper, for counterexample solution problem of multiple until constraint formulae, we put forward a counterexample generation algorithm and one kind of automaton counterexample solution method, and show that the algorithm and the method are effective by example analysis. In the further, we will further consider the solution problem of smallest counterexample finite path sets, and study the optimization problem of algorithms and the probabilistic model checking method based on automaton theory.

Acknowledgements

The authors would like to thank the anonymous reviewers of this paper for their carefully reading of the manuscript as well as their many constructive comments. This paper is supported by the Fundamental Research Funds for the Central Universities (Grant No.DL11BB08), National College Students' Innovative projects (Grant No.201310225064); and the National Natural Science Foundation of China (Grant No. 71001023).

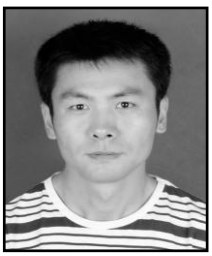
References

[1] F. Ansgar and H. Ralf, "Model checking driven static analysis for the real world: designing and tuning large scale bug detection", Innovations in Systems and Software Engineering, vol. 9, no. 1, **(2013)**, pp. 45-56.
[2] W. Liu, W. Ma and Y. Yang, "Analysis and improvement of the ban modified andrew secure RPC protocol", Journal of Networks, vol. 6, no. 4, **(2011)**, pp. 662-669.
[3] H. Shi, W. Ma, M. Yang and X. Zhang, "A case study of model checking retail banking system with spin", Journal of Computers, vol. 7, no.10, **(2012)**, pp. 2503-2510.
[4] W. Zhang, W. Ma, H. Shi and F. Zhu, "Model checking and verification of the internet payment system with spin", Journal of Software, vol. 7, no. 9, **(2012)**, pp. 1941-1949.
[5] M. Gueffaz, S. Rampacek and C. Nicolle, "Temporal logic to query semantic graphs using the model checking method", Journal of Software, vol. 7, no. 7, **(2012)**, pp. 1462-1472.
[6] F. Antonio, C. Ghezzi and G. Tamburrelli, "Runtime efficient probabilistic model checking", Proceedings of th 33rd international conference on software engineering, **(2011)**, pp. 341-350.
[7] Q. Li, S. Péter, J. Pang and M. Sjouke, "Comparative analysis of clustering protocols with probabilistic model checking", IEEE 6th International Symposium on Theoretical Aspects of Software Engineering, **(2012)**, pp. 249-252.

[8] H. John, K. Marta, N. Gethin, P. David and T. Oksana, "Probabilistic model checking of complex biological pathways", Theoretical Computer Science, vol. 391, no. 3, **(2008)**, pp. 239-257.
[9] J. Niu, G. Zeng and W. Wang, "An approach of model checking time or space performance", Chinese Journal of Computers, vol. 33, no. 9, **(2010)**, pp. 1621-1633.
[10] Junhua Zhang, Zhiqiu Huang, Zining Cao and Fangxiong Xiao, "Counterexample for timed probabilistic reachability in uniform CTMDP," Future Information Tech. and Management Eng'g, **(2008)**, pp. 612-615.
[11] H. Aljazzar, S. Leue, "Generation of counterexamples for model checking of Markov decision processes," International Conference on the Quantitative Evaluation of Systems, pp. 197-206, 2009.
[12] T. Han, J.-P. Katoen and B. Damman, "Counterexample generation in probabilistic model checking," IEEE Transactions on Software Engineering, vol. 35, no. 2, pp. 241-257, 2009.
[13] J. Zhang, Z. Huang, Z. Cao and F. Xiao, "Counterexample generation for probabilistic timed automata model checking", Journal of Computer Research and Development, vol. 45, no. 10, **(2008)**, pp. 638-1645.
[14] J. Wang and G. Zhang, "Counterexample representation for probabilistic timed automata model checking", Journal of Soochow University (Natural Science Edition), vol. 27, no. 02, **(2011)**, pp. 36-43.
[15] L. Zhang, D. N. Jansen, F. Nielson and H. Hermanns, "Automata-based CSL model checking," Proceedings of 38th International Colloquium on Automaton, Languages and Programming, **(2011)**, pp. 271- 282.
[16] H. Fecher, M. Huth, N. Piterman and D. Wagner, "PCTL Model Checking of Markov Chains: Truth and Falsity as Winning Strategies in Games", Performance Evaluation, vol. 67, no. 9, **(2010)**, pp. 858-872.
[17] L. Zhang and M. R. Neuhäußer, "Model Checking Interactive Markov Chains", Tools and Algorithms for the Construction and Analysis of Systems, Springer Berlin Heidelberg, **(2010)**, pp. 53-68.
[18] M. Ji, D. Wu and Z. Chen, "Verification method of conditional probability based on automaton", Journal of Networks, vol. 8, no. 6, **(2013)**, pp. 1329-1335.
[19] B. Damman, T. Han and J. P. Katoen, "Regular expressions for PCTL counterexamples", International Conference on the Quantitative Evaluation of Systems, **(2008)**, pp. 179-188.

Authors

Mingyu Ji. He was born in 1980, Harbin, Heilongjiang province, China. He received the computer application technique master degree in 2004 from Harbin Engineering University, china. Now he is currently a Ph.D candidate in computer science and technology in Harbin Engineering University and works as a lecturer in Northeast Forestry University, china. His research interests include specification and verification of probabilistic and stochastic systems and model checking.



Zhiyuan Chen. He is a Ph.D candidate in computer science and technology in Harbin Engineering University. He received his bachelor degrees and master degrees from Jilin University in 2002 and 2005 respectively, both in computational mathematics. He is currently a lecturer in the College of Computer Science and Technology, Harbin Engineering University. His research interests include model checking and modal logic.

Yanmei Li. She was born in Harbin, Heilongjiang, P.R. China, 1980. She received her master's degree in computer science from Harbin Engineering University in 2008, and she is now pursuing her doctor's degree in College of computer science and technology, Harbin Engineering University. Her research interests include model checking, temporal logic.

